Auton Agent Multi-Agent Syst (2006) 13:391–428
DOI 10.1007/s10458-006-0009-8

Adaptive decision-making frameworks for dynamic multi-agent organizational change

Cheryl Martin · K. Suzanne Barber

Published online: 9 June 2006

Abstract This article presents a capability called Adaptive Decision-Making Frameworks (ADMF) and shows that it can result in significantly improved system performance across run-time situation changes in a multi-agent system. Specifically, ADMF can result in improved and more robust performance compared to the use of a single static decision-making framework (DMF). The ADMF capability allows agents to dynamically adapt the DMF in which they participate to fit their run-time situation as it changes. A DMF identifies a set of agents and specifies the distribution of decision-making control and the authority to assign subtasks among these agents as they determine how a goal or set of goals should be achieved. The ADMF capability is a form of organizational adaptation and differs from previous approaches to organizational adaptation and dynamic coordination in that it is the first to allow dynamic and explicit manipulation of these DMF characteristics at run-time as variables controlling agent behavior. The approach proposed for selecting DMFs at run-time parameterizes all domain-specific knowledge as characteristics of the agents' situation, so the approach is application-independent. The presented evaluation empirically shows that, for at least one multi-agent system, there is no one best DMF for multiple agents across run-time situational changes. Next, it motivates the further exploration of ADMF by showing that adapting DMFs to run-time variations in situation can result in improved overall system performance compared to static or random DMFs.

Keywords Agent autonomy · Adjustable autonomy · Dynamic self-organization · Organizational adaptation · Multiagent systems

C. Martin (✉)
Applied Research Laboratories,
The University of Texas at Austin,
Austin, TX 78713, USA
e-mail: cmartin@arlut.utexas.edu

K. S. Barber
Department of Electrical and Computer Engineering,
Laboratory for Intelligent Processes and Systems,
The University of Texas at Austin, Austin, TX 78713, USA
e-mail: barber@mail.utexas.edu

1. Introduction

Many application domains for multi-agent systems are characterized by complex and dynamic environments. In such domains, an agent's situation, which is defined by characteristics of its own state, its goal(s), and its environment (including other agents), can change significantly and often as the system operates. This research focuses on giving agents the capability to adapt to changing situations by changing the decision-making frameworks (DMFs) in which they participate through a capability called Adaptive Decision-Making Frameworks (ADMF).

The implementation and use of ADMF is motivated by experiments showing performance improvements compared to the use of static or random DMFs in multi-agent systems. A DMF identifies a set of agents and specifies the set of interactions exercised by these agents as they determine how a goal or set of goals should be achieved. The interactions specified by a DMF are (1) decision-making control relationships and (2) authority-over relationships. A specification of decision-making control dictates which agents make decisions about how to achieve a goal. Making decisions for a goal refers to creating, selecting, and allocating sub-goals in order to achieve that goal. A specification of authority-over dictates to which agents the decision-makers can allocate subgoals (i.e. which agents the decision-makers have authority over). The analytical representation for DMFs is given in Section 3.1.

Agents capable of ADMF have the ability to make run-time changes to the DMFs in which they participate, and thus to change their individual decision-making interaction styles at run-time with respect to each goal they are considering. The ADMF capability is analytically defined in Section 3.5. Agents that are not capable of ADMF must use static DMFs throughout system operation. These static DMFs must be established prior to system start-up and are often implicitly defined by the overall system's design. This article focuses on defining and motivating the ADMF capability for agents operating in multi-agent systems. The experiments reported in Section 6 show that implementing the ADMF capability in a multi-agent system can result in significantly improved system performance across run-time situation changes. Specifically, ADMF can support improved and more robust performance compared to the use of a single static DMF.

1.1. Research motivation

As the demand for flexible, adaptive multi-agent behavior has increased, the challenges inherent in the design of multi-agent systems [10] have been amplified by the attempt to automate the processes governing how agents should adapt. The payoff for meeting these amplified challenges is the creation of more capable, more robust multi-agent systems. Different types of run-time adaptation in multi-agent systems have been identified by previous research, as described fully in Section 2. The research presented in this article concerns adaptation at the *organizational level* in multi-agent systems. In general, adapting the organization of a multi-agent system allows agents to overcome problems or improve performance by changing the pattern of information, control, and communication relationships among agents as well as the distribution of tasks, resources, and capabilities. For example, using ADMF, agents may be able to overcome agent failure (by restructuring collaborative decision-making to exclude failed agents), communication failure (by allowing agents awaiting orders to

eventually take initiative), and under-performance (by allowing agents to establish new collaborations that may work better).

The idea that different organizations work better under different situations was first formalized with respect to human organizations as the basis of *contingency theory* [32]. Contingency theory makes the following claims [47]: (1) there is no one best way to organize, (2) all ways of organizing are not equally effective, and (3) the best way to organize is contingent on environmental conditions. Although contingency theory originally considered organizational *design* with respect to the organization's environment, the application of this concept has been extended to organizational *change* with respect to task-environment characteristics [51]. That is, to remain effective over changing situations, an organization may need to change as well.

1.2. Experimental hypotheses

The question addressed by this research asks, "Can agents using ADMF perform significantly better than they could using static DMFs?" This question can be further divided into two parts. First, can the DMFs used by agents in a multi-agent system be represented and manipulated as a controlling variable for the performance of that system? Second, if so, can this controlling variable be manipulated dynamically by the agents in the system (themselves, using no external control) to improve the performance of the system over time? The experimental hypotheses explored by this research address these two questions explicitly, as follows:

Hypothesis 1: *The global decision-making framework (glb*DMF*) under which agents perform best differs as the situation that the agents encounter differs.* Investigating this hypothesis asks for verification of the foundational assumption that the DMF can be a controlling variable in the performance of a multi-agent system in a given situation.

Hypothesis 2: *Given that the glb*DMF *under which agents perform best differs as the situation that the agents encounter differs, agents operating under ADMF can perform significantly better than agents operating under static decision-making frameworks, given run-time situation changes.* Experiments exploring this hypothesis address whether the performance differences across DMFs in a given situation can be exploited to improve run-time performance by allowing agents to dynamically modify DMFs as situations change.

The remainder of this article first discusses related work to provide a context for the implementation and evaluation of the ADMF capability. Sections 3 and 4 then present, respectively, the analytical representations supporting ADMF and the overall characteristics required for implementing the ADMF capability in a multi-agent system. Section 5 follows with a description of the setup used for experiments reported in this article, and Section 6 provides the results from these experiments along with interpretation of the results. Section 7 concludes the article with a summary of the research.

2. Related work

The distribution of decision-making control in a multi-agent system is a characteristic of that system's organization. Therefore, ADMF can be classified as a type

of organizational adaptation for multi-agent systems. Other forms of organizational adaptation have been investigated by related work. The organization of a multi-agent system is defined by an organizational structure plus the identity of the agents who participate in that structure. An organizational structure defines the pattern of information, control, and communication relationships among agents as well as the distribution of tasks, resources, and capabilities [24, 51, 53]. Historically, the organization of multi-agent systems has been determined at design time by the system designer. Research has turned to organizational adaptation to provide improved system performance.

Organizational adaptation allows agents to change aspects of their organization at run-time. In general, agents attempt to realize performance gains by adapting their organization to their current situation. An agent's situation is defined by the current characteristics of its own state, its goals or tasks, and its environment (including other agents). There are two primary classifications for organizational adaptation: (1) organizational reconfiguration in which the structure of the organization remains the same but the identity of the participants in this structure may vary over time and (2) organizational restructuring in which the structure of the organization itself changes over time. The following Sections, 2.1 and 2.2, discuss various implementations that fall into each category. In addition to these types of organizational adaptation, the implementation of "dynamic coordination" mechanisms is closely related to behavior desired under ADMF. Section 2.3 explores dynamic coordination in detail.

2.1. Reconfiguration

Agent interaction within a problem-solving environment can be modeled as the fulfillment of certain application-specific roles [24, 34, 50, 51]. A role specifies application-specific tasks an agent takes on as well as other aspects of organizational structure (e.g. communication and control relationships) associated with those tasks. Organizations can be dynamically "reconfigured" by allowing agents to dynamically assume one or more different pre-defined roles during system operation [51]. Under reconfiguration, the organizational structure remains fixed [51], but the identity of agents participating in different pieces of that structure may vary as the system operates. Established DMFs (often implicit) among agents playing specific organizational roles remain static throughout system operation. Therefore, reconfiguration approaches differ from ADMF, which allows DMFs to be redefined dynamically as the system operates.

One good example of a system implementing organizational reconfiguration comes from research on flexible teamwork using the STEAM approach [55]. The teamwork model in STEAM allows agents to monitor their progress toward achieving team-oriented goals. Agents take on roles with associated application-specific responsibilities (e.g. "scout" or "company commander" in a synthetic battlefield attack helicopter domain). The STEAM model allows agents participating in a simulation to monitor their teammates for the failure to perform tasks associated with their respective roles. If a critical role failure is detected (e.g. the company's scout helicopter crashes into a hill before it can survey its assigned area [55]), the team can be reconfigured by substituting another agent into the failed role. The general teamwork model implemented by STEAM additionally supports a rich set of dynamic coordination capabilities (see Section 2.3).

Another example of organizational reconfiguration was employed by the RETSINA approach [18]. Under RETSINA, in information-agent applications, many

different agent instances may fulfill the role of "information provider" for a given problem. The RETSINA agent approach uses "middle agents" to help route information requests to deal with the failure and recovery of agents or communication links. If, for instance, an agent who was fulfilling a particular type of information provider role fails, a middle agent facilitates the placement of another agent in this role. The set of agents participating in a particular instance of problem solving may therefore vary as the system operates. In this manner, a RETSINA agent organization was able to adapt to unexpected events such as the appearance and disappearance of agents.

Although these reconfiguration approaches successfully address many problems encountered by multi-agent systems, this approach is limited because it assumes that the pre-defined organizational structure is appropriate and is only concerned with maintaining it [51].

2.2. Restructuring

Beyond reconfiguration, organizational restructuring allows agents to actually change their organization structure during system operation [51]. Recall that an organization's structure defines the pattern of information, control, and communication relationships among agents as well as the distribution of tasks, resources, and capabilities.

Organizational restructuring can be implemented at a low level of abstraction by, for example, combining and replicating agents in a network according to task load. Ishida, Gasser, and Yokoo describe a system implementing organizational self-design (OSD) based on strategic work-allocation and load-balancing [28]. The reorganization primitives provided by OSD dynamically vary the system's structure, while the structure of agents themselves remains the same. The OSD uses two primitives to support organizational adaptation: decomposition and composition. As agents become overloaded, they decompose themselves into two agents and share the work. If two agents are idle too long, they compose themselves into one agent, which frees up both computation and communication system resources. Restructuring of this sort affects the system at a fundamental level, allowing it to function more efficiently overall. Unfortunately, low-level reorganization primitives do not generalize well to other applications. For example, composition and decomposition would not work well for a team of agents designed to simulate a military attack team [55] or an American football team [50].

In order to create reorganization primitives that generalize well, an explicit representation of organizational knowledge should be employed. Agents must represent and understand their social and organizational roles and structure as well as the context in which these should be interpreted. Implementing organizational restructuring at a high level of abstraction first requires developing high-level computational descriptions of the organizational structure. Once these descriptions have been developed and incorporated into agent-based systems to guide agent behavior, the descriptions can be manipulated dynamically to implement organizational restructuring. Toward this objective, several researchers have taken steps to computationally define components of organizational structure, including roles and high-level coordination strategies [24, 51]. A formal representation of organizational knowledge has not yet been standardized for agent-based systems. In contrast to using explicit models of organizational structure, an agent's behavior can be guided with respect to its organizational context by incorporating organizational relationships and influence into the agent's model of motivation and utility evaluations [56].

Another direction of multi-agent research focuses on modeling teamwork directly [53, 54]. Teamwork models help agents monitor their collective performance in order to initiate reorganization. However, these models do not directly address a system's organizational structure.

2.3. Dynamic coordination

Both organizational reconfiguration and organizational restructuring allow agents to adapt their interactions to their current situation. Previous work on dynamic coordination also reflects the effort to adapt agent interactions to particular situations. Dynamic coordination allows agents to dynamically change the way interleaving agent actions are scheduled, change which agent is responsible for what goal, or change the mechanism through which agents achieve coordinated behavior (e.g., contract-net). The effects of dynamic coordination can be very similar to the effects of organizational adaptation, and these concepts are therefore difficult to distinguish from one another based solely on observation of system behavior. In general, dynamic coordination mechanisms allow interaction changes that are specific to the application or closely linked to how a specific task will be accomplished. Conversely, reorganization mechanisms focus primarily on manipulation of application-independent concepts (e.g. roles, connectivity, DMFs, etc.), which can, in turn, affect agent interaction and coordination behavior. Reorganization mechanisms generally operate at a higher level of abstraction than do dynamic coordination mechanisms.

Osawa presents an experiment showing that for at least one type of situational change in the pursuit game that dynamic coordination can be beneficial. The coordination strategies considered by Osawa's work are defined by previous experiments for the predator-prey application [52] and dictate how predator agents work together to capture the prey. These strategies include the specification of interactions and problem-solving algorithms specific to the pursuit problem. Osawa proposes a meta-level coordination strategy that dictates for each predator agent which single pursuit coordination strategy should be used for a given situation and point in the pursuit. This meta-level coordination strategy is based on successive use of individual coordination strategies in order of decreasing "freedom" for the agents. Freedom is defined as the number of possible local plans for the agents. The meta-level coordination strategy requires no communication overhead because all agents in the sample problem are assumed to have commonly held knowledge and reasoning algorithms that guarantee they will synchronously (in the same time step) choose the same coordination strategy. This work provides evidence that multi-agent systems can realize performance improvements by changing how they work together (coordination strategies) to adapt to current situations.

Excelente-Toledo and Jennings present a dynamic coordination approach for a grid world that goes beyond Osawa's static specification of contingency coordination mechanisms and allows agents to decide how they will coordinate at run-time, based on the evaluation of a set of available coordination mechanisms (e.g. contract net) [22]. Under this approach, when an agent discovers a goal that needs coordination for achievement (in this case, more than one agent must reach a goal grid square), it proposes the goal and possible coordination mechanisms to other agents. These other agents bid on whether they are willing to coordinate on this new goal based on a cost/benefit analysis taking into account the possible reward, the probability of success, and the cost of participating in the suggested coordination mechanism(s). If

an agent's bid to coordinate on the new goal is accepted, the agent adopts the goal and begins working within the proposed coordination mechanism. Using this approach, agents agree on a coordination mechanism before adopting a goal that may require coordination to achieve. If an agent does not believe that coordination is viable, it may reject the goal. (This differs significantly from the ADMF approach where goal adoption is assumed to be driven by factors that may be independent of selecting the coordination mechanism eventually adopted to achieve the goal.)

Also within the category of dynamic coordination, the overall application-specific role of an agent as well as its decision-making interactions with other agents may remain constant, but some of its lower-level tasks or actions may change to fit the situation. Representations such as production lattices [25] and partial global plans [20, 21] support the implementation of dynamic coordination mechanisms. Another example of dynamic coordination can be seen in the anti-air defense domain where agents must coordinate to destroy incoming warheads [42]. Although the agents' organizational structure is fixed, where each agent locally determines and implements its own behavior, the coordination among the agents (i.e. which incoming warhead each agent shoots down) is dynamic. Each agent uses payoff matrices to determine the most useful action for it to take, given its model of the environment and other agents. Overall, research using this example shows that allowing agents to determine their own coordination behavior (including action and message selection) on-the-fly (rather than relying on fixed protocols) performs well for dynamic, unpredictable domains [41, 42]. In addition, Decker and Lesser have shown that dynamic coordination is more effective than static coordination for distributed sensor networks [17]. Because organizational adaptation affects coordination by changing agent interaction behavior, it may hold similar promise for producing improved performance. However, for a fully flexible system, both organizational adaptation and dynamic coordination may prove useful. For example, both voting and negotiation can be selected as coordination mechanisms given the same chosen distribution of decision-making control.

2.4. Adjustable autonomy

As the sophistication of agent-based systems has increased over the past decade, the demand for agent-based systems capable of adjustable autonomy has increased tremendously [5, 19, 40, 46]. Unfortunately, because no explicit definition of agent autonomy has been agreed upon in the agent community, there is also little agreement about what it means to adjust an agent's autonomy. Therefore, various researchers investigating adjustable autonomy are considering the issue from various different viewpoints. Recently, several researchers investigating adjustable autonomy collaborated to describe the various dimensions of adjustable autonomy being explored and relate them to one another [7].

One realization of adjustable autonomy focuses on allowing a given agent to dynamically change the high level goals pursued by some agent as well as the priorities of these goals [43]. Other investigations of adjustable autonomy, including the research reflected in this article, focus on adjusting an agent's control in making decisions (e.g. degree of decision-making control [3], level of supervision for decisions made [40], or control over delegation and adoption of goals [23].

Under another common interpretation, the concept of adjustable autonomy focuses on human-to-agent interaction in which a human can give an agent (or an agent can assume) greater or fewer available options for carrying out a given goal. This view of

adjustable autonomy for agent-based systems has arisen to support varying degrees independence from human control for automated systems (e.g. robotic systems such as a Mars rover [19], long-duration space-flight control systems for advanced life support [31], industrial control systems such as oil refinery monitoring and emergency recovery [39], and intelligent personal assistants for activities such as scheduling meetings [46]). For systems such as these, autonomous action on the part of software-based systems is desirable under most circumstances, but system designers still want humans to be able to intervene when necessary (e.g. for re-tasking, safety or preference overrides, and system repair or calibration). The motivation for adjustable autonomy in these systems is clear: greater autonomy for agents frees humans from long-enduring, tedious, dangerous, costly, or error-prone tasks, but human guidance or intervention is still required under some circumstances to achieve desired results. In these systems, capability differences between humans and software-agents as well as issues of safety and trust motivate the implementation of adjustable autonomy.

This research reflected in this article defines agent autonomy with respect to decision-making control [3], which can be relative to any other agent, including a human. The distribution of decision-making control in the system dictates the level of autonomy each agent possesses. An agent's degree of autonomy, with respect to some goal that it actively uses its capabilities to pursue, is the degree to which the decision-making process used to determine how that goal should be pursued is free from intervention by any other agent. For a complete argument supporting the development of this definition of autonomy (see [4]). The ADMF supports adjustable autonomy under this definition by allowing agents to dynamically change their current degree of decision-making control for a given goal. In the experimental system considered by this article, all agents are capable of fully autonomous operation (complete decision-making control) but may choose to give up some or all of their decision-making control through ADMF in to achieve better performance in their current situation.

2.5. Coalition formation

The research area of coalition formation in multi-agent systems considers the problem of which agents should work together [45, 49]. A "coalition" is defined by this previous work as a group of agents working together [27], or more specifically, a group of agents working together to either achieve a group task or allocate globally assigned tasks to individual agents [12]. Coalition formation research has been very active for e-commerce application domains (e.g. a group of buyers work together to place a large order at a discount price). For coalition formation research, agents are often self-interested and motivated to join a coalition to improve their local payoffs. In general, agents have different capabilities, resources, or expertise that make forming coalitions mutually beneficial. Much work has focused on searching for optimal divisions of agents into coalitions such that the sum of the payoffs to all coalitions is maximized [48]. Agents often negotiate to form coalition groups. Recent work has addressed improving both the search for beneficial coalitions [12, 27] and the negotiation process to form coalitions [33] with respect to scalability to large numbers of agents (thousands).

In the future, coalition formation work can be leveraged to extend the ADMF capability to better determine which agents should work together, in a decision-making group, in addition to determining the best set of decision-making interaction

constraints defining *how* those agents should work together. The ADMF research presented in this article considers agents with homogeneous capabilities, resources, and expertise. This allows the experiments to focus primarily on determining the decision-making interactions of a group of agents rather than determining which agents should be in these groups. However, this second point will become increasingly important for the future development of the ADMF capability.

2.6. ADMF in context

The ADMF can be classified as an implementation of organizational restructuring, which is a type of organizational adaptation. Through ADMF, agents can model and change the locus of decision-making control and authority to assign tasks within their system at run-time. These changes are modifications to the organizational structure of the agents' system. ADMF does not model or attempt to change the complete organizational structure of a multi-agent system. It does not define or modify the distribution of information, tasks, resources, communication, or agent capabilities. The ADMF focuses on modeling and adapting who is in control of making decisions about how to achieve goals, how much control each decision-making agent has, and to whom tasks can be readily allocated by the decision-making agents.

Which DMF is most appropriate for a given situation depends on how well the agents perform in that situation under that DMF. Previous research has described both advantages and disadvantages for *statically* defined centralized, distributed, and local-control decision-making structures [9, 11, 15, 36, 38, 51]. Decentralized distributions of decision-making control (exhibiting distributed or local control) tend to result in faster problem solving if parallelism can be exploited and in decreased or normalized communication because only high-level partial solutions are transmitted to other agents, rather than raw data transmitted to a central site [38]. Decentralized decision-making can be more responsive under uncertainty and changes in the external environment, but centralized decision making tends to perform faster in some situations due to the absence of the requirement to negotiate [36]. A decentralized paradigm performs well when few resources are shared, but the centralized model performs well when many resources are shared [9]. For simple tasks, centralized systems are faster and more accurate, but for complex tasks distributed systems are faster and more accurate [15]. Related results in the field of distributed constraint optimization have shown that the best choice between partial problem centralization (OptAPO) and decentralization (Adopt) depends on communication latency [16]. ADMF attempts to leverage these well-known tradeoffs to improve performance by dynamically changing distribution of decision-making control in a system as the system characteristics change.

3. Analytical representations supporting ADMF

A computational representation is needed to realize the capability of ADMF. Such a representation gives the agent something to set, a "knob to turn" so to speak, allowing decision-making control to be assigned and adjusted. The concept of a DMF, and therefore its computational representation, is founded on the following definition of agent autonomy, developed in [4]:

> An agent's degree of autonomy, with respect to some goal that it actively uses its capabilities to pursue, is the degree to which the decision-making process, used to determine how that goal should be pursued, is free from intervention by any other agent.

The argument presented in [4] shows that the most appropriate type of intervention to be considered is of the type goal/task determination. Of the other possible types of intervention identified, environmental modification is shown to be non-autonomy altering, and belief influence is shown to have an indirect affect on autonomy. Since any given level of interface to an agent's belief system still requires that agent to internally interpret and process belief-altering inputs, any given agent is already free, at some basic (interpretive) level, from belief-influence intervention. On the other hand, if an agent is able to accept new goals at run-time through a task assignment action, then that agent is not free from interventions of the type goal/task determination.

3.1. DMF representation

The definition of agent autonomy given above ties autonomy to decision-making control and identifies the required elements of a DMF representation [4]. First, how, and to what extent, other agents intervene must be modeled. The essential elements describing this intervention include (1) which agents are in control of the decision-making process for a particular goal and (2) how much control each of these agents has within the process. However, representing implementation details of the decision-making process (e.g. a specific negotiation protocol) should be avoided because the representation must generalize across decision-making algorithms (e.g. it should allow voting as well as negotiation).

Second, the definition given above also indicates that an agent's decision-making control must be represented with respect to a goal, and that the agent must actively use its capabilities to pursue this goal. Therefore, the DMF representation must identify the goal or set of goals to which the corresponding DMF applies (i.e. agents make decisions about these goals within the DMF). Recall that making decisions for a goal refers to creating, selecting, and allocating sub-goals in order to achieve that goal. In addition, in order for an agent to actively use its capabilities to pursue a goal, it must "intend" the goal [14] or in some way form a commitment to the goal [13, 30]. Therefore, each goal to which a DMF applies must be intended by some agent in the system.

Third, any model related to decision-making control must also explicitly represent a constraint that enforces the authority of the decision-making agents to assign subgoals required to carry out the intended goal. This task assignment is a form of "goal/task determination" intervention. This authority-over constraint ensures that at least one agent will commit to execute the decisions made by the decision-making agents.

The representation for DMFs developed in this research is referred to as the *DMF representation*. It represents the entire framework in which each participating agent exercises a specified degree of decision-making control over a specified set of goals. The DMF representation consists of three components (D, G, C) identifying the set of decision-makers (D), the set of goals to make decisions about (G), and the set of agents who must carry out the decisions for a given DMF (C). Figure 1 provides a high-level conceptual view of the relationships among the sets in the DMF representation.

Table 1 presents the specification for the DMF representation describing the DMF's decision-makers (D), goals in focus (G), and agents under the authority-over

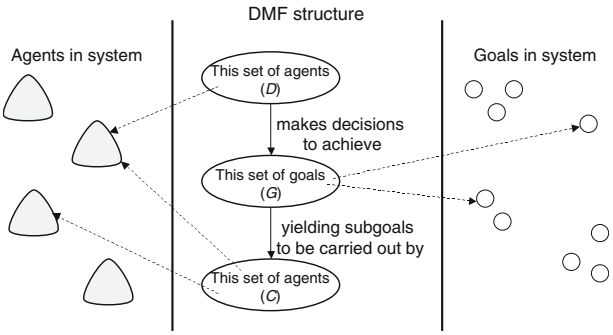


Fig. 1 DMF structure and DMF representation variables

Table 1 Specification for DMF representation

DMF representation (D, G, C)		
D	Decision-makers	$\{a_0[, a_1, a_{n-1}]\}$, or $\{a_1[, \ldots, a_{n-1}]\}$
G	Focus	$\{g_i^{a_0}[, g_j^{a_1}, \ldots, g_k^{a_{n-1}}]\}$ or $\{g_j^{a_1}[, \ldots, g_k^{a_{n-1}}]\}$
C	Authority-over constraint	$\{a_0[, a_1, \ldots, a_{n-1}]\}$ or $\{a_1[, \ldots, a_{n-1}]\}$

constraint (C). There are n agents in any given system. Agents must distinguish between "self" and "others," where "self" refers to the agent who is using the DMF representation to determine how much decision-making control it should exert for a given goal or to determine whether or not it is required to accept subgoal assignments for a given goal. For the purposes of this discussion, the following variables are defined:

Let a_0 represent the self-agent, with the unique identifier 0,
let a_i represent any other agent, with the identifier $i \neq 0$, and
let a_x and a_y represent any agent, with the identifiers x and y.

In addition, let $g_i^{a_x}$ represent the ith goal intended by agent $a_{x,}$.

The DMF representation is a tuple of three sets (D, G, C). The bracketed notation in Table 1 indicates that the enclosed elements are optional for each instance. Two versions of each specification are given, depending on whether the self-agent, a_0, is included in a given instance or not. Each set (D, G, C) in the DMF representation is explained in detail in the following sections.

3.1.1. Decision-makers, D

The set D identifies which agents make decisions about how the goals specified by G should be pursued. Making decisions for a goal refers to creating, selecting, and allocating sub-goals in order to achieve that goal. For the purposes of the experiments described in this paper, each agent in D plays an equal part in determining how to pursue G (e.g., true consensus). The full representation, previously developed [2, 4], supports varying strength among the decision makers to allow for supervisory and supervised decision-making roles.

A DMF constrains, but does not completely specify, the coordination strategy used for decision-making. This coordination strategy is selected after a DMF is established

[1]. For example, a DMF instance that specifies three decision-makers with equal decision-making power can be realized by a simple suggest-and-vote coordination strategy or by a three-way negotiation coordination strategy. For a given multi-agent system with given decision-making algorithms, a mapping must be drawn between possible assignments to D and which of the available decision-making algorithms can be used given the constraints specified by D [2]. Section 5.2 shows how this mapping is constructed for experiments considered by this article, and there are no experimental cases in which multiple coordination strategies are available given the DMF selected. Combining the ADMF capability with dynamic selection of coordination strategy (e.g., [22]) is an open area for future work. Future work may also explore more expressive representations for DMFs including the ability to represent veto power, information exchanging requirements among DMF members in the process of decision-making, and specific supervision or check-off requirements.

3.1.2. Focus, G

The focus of a DMF, G, identifies the goal(s) about which agents are making decisions. Any agent may make decisions for goals it intends to achieve as well as for goals that other agents intend to achieve. Cohen and Levesque provide a formal model of intention, specifying that once an agent chooses and commits to a goal, it intends that goal and will attempt to achieve it [14]. Additionally, agents may combine their goals in G for simultaneous solution. If the number of elements in G is greater than one ($|G| > 1$), then the decisions made by the members of D must satisfy all goals in G simultaneously.

The DMF representation imposes a constraint that no two agents may intend the same instance of a goal, as shown by the following example. Agent_1 (a robot) and Agent_2 (with the ability to manipulate a maze) may both intend the goal "Agent_1 travel through maze no 42." Agent_1 may intend to achieve the goal $g_1^{\text{Agent\_1}}$ = "Agent_1 travel through maze no 42," and Agent_2 may intend to achieve the goal $g_4^{\text{Agent\_2}}$ = "Agent_1 travel through maze no 42." In this case, $g_1^{\text{Agent\_1}}$ is equivalent to $g_4^{\text{Agent\_2}}$, but $g_1^{\text{Agent\_1}} \neq g_4^{\text{Agent\_2}}$. In other words, these intended goals look the same but are not the same instance and can be maintained independently. These two agents may or may not engage in collaborative decision-making. In fact, Agent_1 may not even know about Agent_2 or its goal. If the two agents do engage in collaborative decision-making to get the Agent_1 through the maze, they should both represent a focus of the form $G = \{g_1^{\text{Agent\_1}}, g_4^{\text{Agent\_2}}\}$. If, for some reason, the two agents become unable to work together (e.g. communication fails), their DMFs might change, but their goals would remain unchanged. The intentions themselves remain intact, and the two agents may each continue independently, in their own way, to try to get Agent_1 through the maze.

This representational approach differs significantly from previous work on representing joint intentions [14, 26] and joint commitments [13, 29], in which some collective entity (e.g., a group of agents) jointly holds a distributed commitment to a goal or set of goals. Under this previous work, an agreement to work together within the collective is formed simultaneously with establishing the commitment to the joint goal itself. An intention formed jointly using this previously developed representation cannot remain intact for independent pursuit once the agents become unable to work together. Conversely, the representational approach presented by this research

better supports the implementation of ADMF by clearly separating the representation of intention to achieve a goal from the representation of how that intention may be "joint" and thus executed in some coordinated fashion within an established organization, the latter of which is subject to change under ADMF.

3.1.3. Authority-over constraint, C

The set *C* simply lists the agents who are bound to carry out the decisions made by the decision-makers identified in the set *D*. The decision-makers are said to have authority over the agents in *C* because these agents in *C* have previously committed to the DMF, thereby committing to accept task assignments from the decision-makers, which are required to carry out the goal(s) identified by *G*. The authority-over constraint, *C*, ensures that some agent(s) will carry out the decisions of the decision-making group. If an agent listed in *C* fails to accept its required task assignment, it must pay a penalty for breaking its commitment to the DMF (see Section 4.3).

3.1.4. Cardinality and conflicting DMFs

Figure 2 shows allowable cardinality relationships among variables within DMFs. For any goal, only one DMF can be established at a given time. (Otherwise, one could not determine how decisions should actually be made to pursue that goal). One established DMF may apply to any number of goals ($i \geq 1$). Any number of agents ($k \geq 1$) may participate in an established DMF. A given agent can participate in any number of established DMFs ($j \geq 0$) either as a member of *D*, a member of *C*, or a member of both *D* and *C* for each of these DMFs. In general, an agent may participate in multiple, simultaneous established DMFs up to and including one for every goal in the system.

Because only one DMF can be established for a given goal at any given time, any two DMFs are defined to *conflict* if they apply to the same goal, as shown in Fig. 3. The DMFs that do not conflict may be established simultaneously. Only one of a set of conflicting DMFs can be established at any given time. Further, no DMF in the set of established DMFs should conflict with any other DMF in that set. Each individual agent is responsible for making sure that it does not have more than one DMF active for any given goal in its intentional model.

3.1.5. DMF terminology

A particular set of values applied to the (*D*, *G*, *C*) DMF representation is referred to as an instance of a *DMF instance*. Each agent in either the set *D* or the set *C* for a given

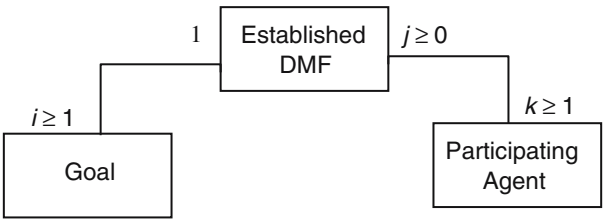


Fig. 2 Cardinalities for relationships of DMF concepts

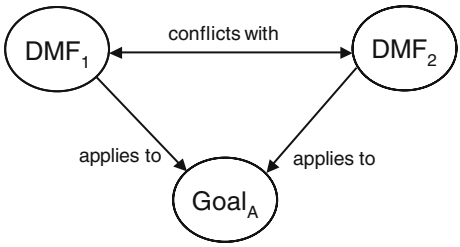


Fig. 3 Depiction of concept of conflicting DMFs

DMF instance is said to *participate* in the associated DMF. A DMF is said to *apply to* each of the goals specified by G for the associated DMF instance. A DMF that is currently in effect in the multi-agent system is called an *established DMF*. Each agent participating in an established DMF has a *view* of the DMF. That is, each participating agent models the established DMF, using an independent DMF instance, as part of its own internal belief set as pictured in Fig. 4. This figure shows that both Agent 1 and Agent 2 participate in a given DMF, and both of these agents represent this DMF independently (shown in thought bubbles for each agent).

The representation leaves open the possibility that some participating agent may have an incomplete model of a given established DMF (e.g., an agent in the set C may not model other agents in the set C).

3.1.6. Decision-making styles

An agent's individual decision-making interaction style describes how that agent participates in a given DMF. Agents capable of adapting DMFs adopt a decision-making interaction style for each goal they pursue according to the DMF applied to those goals. Interpreting DMFs with respect to the decision-making interaction styles of participating agents allows an agent to classify its decision-making interaction style for a given DMF instance and match that prescribed style to the set of algorithms it can use to act and interact within that framework.

Applying named labels to these decision-making styles allows agent designers to discuss agent behavior within a DMF using qualitative labels, which are easier to discuss verbally and in text than are the associated assignments to DMF variables. Three

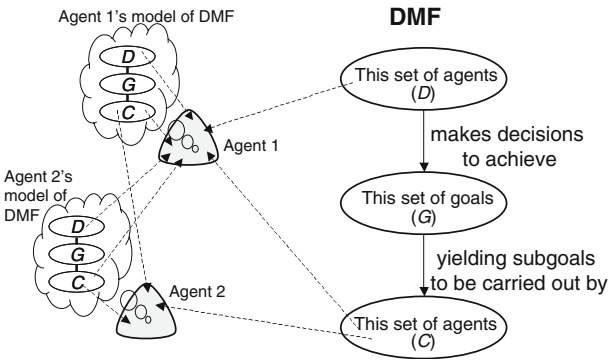


Fig. 4 Participating agents' view of a given DMF

discrete categories of decision-making interaction styles are classified and associated with named labels for this purpose in this article:

- Command-driven—The agent does not make any decisions about how to pursue its goal and must obey orders given by some other agent(s).
- True consensus—The agent shares decision-making control equally with all other decision-making agents.
- Locally autonomous/master—The agent makes decisions alone and may or may not give orders to other agents.

Table 2 provides the mapping from the DMF representation to these named labels.

An entire DMF can also be labeled informally according to the decision-making interaction styles of participating agents (e.g. a "master/command-driven" DMF would involve one agent with a "master" decision-making interaction style and at least one agent with a "command-driven" decision-making style). These qualitative names are used throughout the remainder of this article to convey these characteristics for the corresponding DMF assignments.

3.2. Global DMFs

The DMF concept can be extended to define a *glb*DMF, as the set of all DMFs that are established in a system at a given time. Figure 5 shows a pictorial example of a *glb*DMF. A *glb*DMF must contain one or more DMFs that apply to each goal in the system, and a *glb*DMF can contain no conflicting DMFs. If no DMF is established for a given goal in the system at a given time, then the *glb*DMF corresponding to

Table 2 Classification of decision-making styles for qualitative names

An agent, a_0, is considered	DMF representation constraints
Locally autonomous if	
(1) it is the only decision-making agent,	($\|D\| = 1$ and $a_0 \in D$)
(2) the focus of its decision making is one of its own intended goals, and	($\|G\| = 1$ and $g_i^{a_0} \in G$)
(3) it is the only agent on the authority-over list	($\|C\| = 1$ and $a_0 \in C$)
Master if	
(1) it is the only decision-making agent and	($\|D\| = 1$ and $a_0 \in D$)
(2) at least one other agent is on the authority-over list	($\exists a_i : (a_i \in C)$)
Command-driven if	
(1) it is not making decisions and	($a_0 \notin D$)
(2) it is on the authority-over list	($a_0 \in C$)
True consensus if	
(1) it is making decisions along with at least one other agent,	($\|D\| > 1$ and $a_0 \in D$)
(2) the decision-making power of each agent making decisions is equal, and	(Always true for all members of D in the default case where no representation of variable decision-making power is included in D)
(3) all agents who are making decisions (and only these agents) are on the authority-over list.	($\|D\| = \|C\|$ and $\forall a_x$: if $a_x \in D$ then $a_x \in C$)

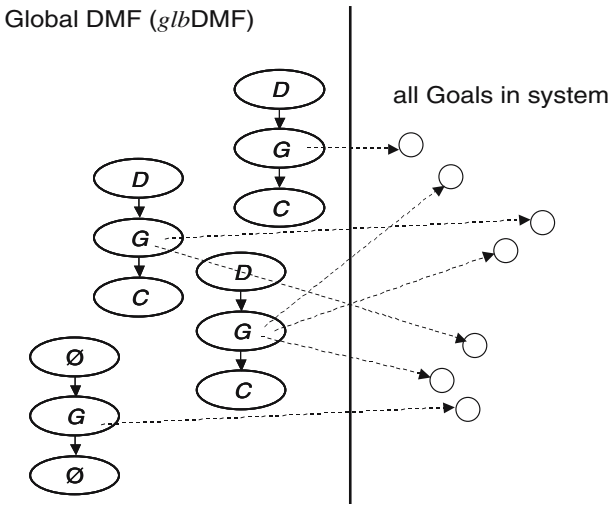


Fig. 5 Example of global DMF (*glb*DMF)

that time must include a NULL DMF (no decision-makers (D), no authority-over members (C)) for that goal (as shown in the lower-left corner of Fig. 5).

Let F represent the space of all possible *glb*DMFs in a system. $f \in \mathrm{F}$ is a *glb*DMF. Let $\hat{\mathrm{F}}(x) \subseteq \mathrm{F}$ be the achievable *glb*DMFs in situation x. That is, $\hat{\mathrm{F}}(x)$ is the subset of all possible *glb*DMFs that can be established, if not already existing, in a given situation x. For all *glb*DMFs considered in this article, $\hat{\mathrm{F}}(x)$ depends only on variables in x that relate to communication availability: all agents in D and C must be able to communicate with one another in order to establish the DMF if it is not already established.

Three types of *glb*DMFs can be named and labeled using the labels defined in the previous section for any system containing $n \geq 2$ agents with one goal each. These frameworks correspond to maximum participation by all system agents in one of the three salient decision-making interaction styles identified above in Section 3.1.6.

(1) All LA *glb*DMF: Each agent in the system is locally autonomous for its own goal.
(2) All CN *glb*DMF: Every agent in the system participates in one consensus decision-making framework in which every agents' goal is considered concurrently.
(3) All M/CD *glb*DMF: One agent in the system acts as a master and makes decisions about its own goal as well as the goals of every other agent in the system. All other agents in the system are command-driven to that master.

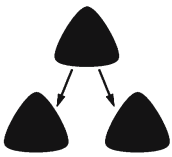
Additional *glb*DMFs can be defined for any number ($n > 2$) of system agents. An agent designer may identify a subset of all valid *glb*DMFs as relevant for a given system or set of goals. This subset composes the set of all possible *glb*DMFs for that system.

Table 3 specifies the details of each *glb*DMF in considered as the available set for the purposes of experiments in this article. The formal specification for each agent's subjective view of the *glb*DMF is provided along with a simple icon summarizing the relationships among the agents. The possible global decision-making frameworks considered for the experimental three-agent system include (1) "All LA," all the

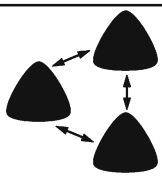
Table 3 Global DMFs for a 3-agent system with icons

All LA		Styles and DMF models (D, G, C) held by each agent
agent a_1	LA	($\{a_1\}$, $\{g_1^{a_1}\}$, $\{a_1\}$)
agent a_2	LA	($\{a_2\}$, $\{g_1^{a_2}\}$, $\{a_2\}$)
agent a_3	LA	($\{a_3\}$, $\{g_1^{a_3}\}$, $\{a_3\}$)

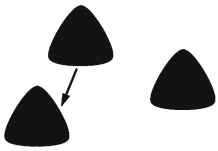
All M/CD		Styles and DMF models (D, G, C) held by each agent
agent a_1	LA	($\{a_1\}$, $\{g_1^{a_1}\}$, $\{a_1\}$)
	M	($\{a_1\}$, $\{g_1^{a_2}\}$, $\{a_2, a_1\}$)
	M	($\{a_1\}$, $\{g_1^{a_3}\}$, $\{a_3, a_1\}$)
agent a_2	CD	($\{a_1\}$, $\{g_1^{a_2}\}$, $\{a_2, a_1\}$)
agent a_3	CD	($\{a_1\}$, $\{g_1^{a_3}\}$, $\{a_3, a_1\}$)



All CN		Styles and DMF models (D, G, C) held by each agent
agent a_1	CN	($\{a_1, a_2, a_3\}$, $\{g_1^{a_1}, g_1^{a_2}, g_1^{a_3}\}$, $\{a_1, a_2, a_3\}$)
agent a_2	CN	($\{a_1, a_2, a_3\}$, $\{g_1^{a_1}, g_1^{a_2}, g_1^{a_3}\}$, $\{a_1, a_2, a_3\}$)
agent a_3	CN	($\{a_1, a_2, a_3\}$, $\{g_1^{a_1}, g_1^{a_2}, g_1^{a_3}\}$, $\{a_1, a_2, a_3\}$)



2M/CD-1LA		Styles and DMF models (D, G, C) held by each agent
agent a_1	CD	($\{a_3\}$, $\{g_1^{a_1}\}$, $\{a_1, a_3\}$)
agent a_2	LA	($\{a_2\}$, $\{g_1^{a_2}\}$, $\{a_2\}$)
agent a_3	LA	($\{a_3\}$, $\{g_1^{a_3}\}$, $\{a_3\}$)
	M	($\{a_3\}$, $\{g_1^{a_1}\}$, $\{a_1, a_3\}$)



2CN-1LA		Styles and DMF models (D, G, C) held by each agent
agent a_1	LA	($\{a_1\}$, $\{g_1^{a_1}\}$, $\{a_1\}$)
agent a_2	CN	($\{a_2, a_3\}$, $\{g_1^{a_2}, g_1^{a_3}\}$, $\{a_2, a_3\}$)
agent a_3	CN	($\{a_2, a_3\}$, $\{g_1^{a_2}, g_1^{a_3}\}$, $\{a_2, a_3\}$)

agents act in a locally autonomous manner, (2) "All M/CD," one master controls two command-driven agents, (3) "All CN," all the agents collaborate in true consensus, (4) "2M/CD-1LA," two agents are in a master/command-driven relationship, and one agent acts in a locally autonomous manner, and (5) "2CN-1LA," two agents are in consensus and one agent acts in a locally autonomous manner.

These global frameworks are a subset of all valid *glb*DMFs for this system and are chosen to provide coverage of "interesting" cases (i.e. maximum participation in each

of LA, M/CD, and CN DMFs, respectively, as well as combinations of collaborative and locally autonomous frameworks).

3.3. Situation definition and notation

As situations change, agents operating under ADMF attempt to change their DMFs (resulting in a new *glb*DMF) to best fit the new situation. An agent designer may identify a subset of all possible situation characteristics as relevant for a given system or agent. The set of all possible values for all these relevant characteristics for every agent in a system composes the set of possible system states for that system.

Let X define the state space for all possible system states. Section 5.3 gives a concrete example in terms of the situations explored by experiments for this article. A state in this space, $x \in X$, is called a "situation." The space X has m dimensions, where each dimension is defined by a discrete set of possible values for a given state variable, yielding $X = \{X^i\}_{i=1}^m$, where each X^i is a discrete set.

3.4. System performance

Let p define a function that maps the spaces of X and F to a positive real number, given some set of initial conditions for the agents in the situation (indexed by i). The function $p(x, f, i)$ is a measure of the performance per unit time of all agents in achieving their goal in situation x under *glb*DMF f given initial conditions indexed by i.

The function, $p(x, f, i)$, is defined based on application domain characteristics and can be determined empirically. Section 5.5 below, describes how $p(x, f, i)$ is determined for the experimental problem domain considered by this article. The expected performance $p(x, f)$ for a given *glb*DMF f in a given situation x, can be estimated by taking an average over several observations.

3.5. Changes over time as system operates

Systems capable of ADMF are concerned with changes in both situations and DMFs at run time. This section presents the analytical model for these changes in the experimental system. Let a scenario, S, be a sequence of consecutive situations, x, of the form $S = (x_1, x_2, x_3, \ldots)$. $x_k \in \mathrm{X}$ is the situation that exists in a given scenario during the kth time interval in that scenario.

For $1 \leq k \leq K$, let τ_k^N represent the kth time interval, of fixed length N time units, in a given scenario, where K is the total number of time intervals in that scenario. τ_k^N is defined by the time interval $\tau_k^N = \left[t_{N\cdot(k-1)}, t_{(N\cdot k)-1}\right]$. N is constrained for a given system such that $N \geq N_0$, which is the minimum number of time units agents need to perform interaction in τ_k^N. Let a scenario of length K, S^K, be defined as a scenario consisting of K time intervals,τ_k^N, where the value of N is given separately by the context in which a particular S^K is presented.

During any scenario, agents must use some *glb*DMF to make decisions about how to achieve their goals in each situation they encounter. Let $f_k \in F$ be the *glb*DMF that agents are using to determine decision-making interactions during time interval τ_k^N. Let A^K be a sequence, of length K, of these *glb*DMFs.

Changes in A^K from f_k to f_{k+1} are controlled by the DMF policy the agents are using. Several types of DMF policies are compared for these experiments:

- *Static decision-making frameworks*—the agents do not change their decision-making frameworks during system operation. A "Static f" policy is denoted by $A_{\text{STATIC } f}$ such that $f_{k+1} = f_k = f$. For example, a "Static All LA" policy is denoted by $A_{\text{STATIC AllLA}}$ such that $f_{k+1} = f_k = \text{AllLA}$.
- *Adaptive decision-making frameworks*—the agents use the ADMF capability to modify their decision-making frameworks during system operation. The "ADMF" policy is denoted by A_{ADMF} such that

$$f_{k+1} = f : best_{f \in \left\{\hat{\text{F}}(x_{k+1}) \cup f_k\right\}} \left(p(x_{k+1}, f_{k+1})\right).$$

  This equation indicates that the achievable *glb*DMF that gives the best expected performance, for the time interval when x_{k+1} exists, is chosen as f_{k+1}.
- *Random decision-making frameworks*—the agents choose and implement DMFs randomly during system operation. The "Random" policy is denoted by A_{RANDOM} such that $f_{k+1} \in \text{F}$, given random selection with uniform distribution in F. For A_{RANDOM}, each agent must choose and implement the same *glb*DMF at the same time, without depending on communication. Note that the A_{RANDOM} policy is used strictly for experimental comparison. Random implementation of DMFs could not occur in practice because agents could never agree to establish a randomly selected DMF without experimental control. A_{RANDOM} requires agents to be seeded with the same initial random seed and use the same random number generation algorithm to select *glb*DMFs. If each agent were allowed to choose a *glb*DMF independently at random, a single, valid *glb*DMF could not be established. Although the implementation of A_{RANDOM} could not be used in practice, exploring this policy provides evidence of whether it is simply changing *glb*DMFs or changing *glb*DMFs in a reasoned manner under ADMF that can improve agent performance.

Let P define a function that maps the spaces of S^K and A^K to a positive real number, given the sequence of initial conditions I^K. The function, $P\left(S^K, A^K, I^K\right)$, is the penalty per unit time that agents incur while operating across an entire scenario, S^K, given A^K and I^K. $P\left(S^K, A^K, I^K\right)$ can be formulated in terms of $p(x, f, i)$, such that

$$P\left(S^K, A^K, I^K\right) = \frac{\sum_{k=1}^{K} p(x_k, f_k, i_k)}{K}.$$

The expected performance of a given ADMF policy $P\left(S^K, A^K_{\text{name}}\right)$ can be estimated by applying the ADMF policy over several scenarios, S^K, observing $P\left(S^K, A^K_{\text{name}}, I^K\right)$, and averaging the resulting performance observations.

3.6. Analytical representation of experimental hypotheses

Given the analytical definitions of the concepts presented in this section, we can now more formally describe our hypotheses. For hypothesis 1, the experiments explore whether *the glb*DMF ($f \in \text{F}$) *under which agents perform best* (determined by empirically estimating $p(x, f)$) *differs as the situation* (x) *that the agents encounter differs*. An empirical estimate of $p(x, f)$ is made for all combinations of $f \in \text{F}$ and $x \in \text{X}$. Then, for each $x \in \text{X}$, it is determined for which $f \in \text{F}$ the performance measure $p(x, f)$ is best (maximum or minimum, depending on whether the performance measure is a positive measure or a penalty measure). Let the function $bestDMF(x)$ map $x \in \text{X}$

to $f \in \mathrm{F}$ such that the performance measure $p(x, f)$ is best. The hypothesis indicates that $bestDMF(x_1)$ should not equal $bestDMF(x_2)$ for all x. Further, the performance difference should be significant between $p(\mathrm{x}, bestDMF(x))$ and $p(x, f)$, where $f \neq bestDMF(x)$.

For hypothesis 2, the experiments explore whether *agents operating under ADMF* (A_{ADMF}) *can perform significantly better* (based on $P\left(S^K, A^K_{\mathrm{name}}\right)$) *than agents operating under static* ($A_{\mathrm{STATIC}\,f}$) *or random* (A_{RANDOM}) DMFs *given run-time situation changes* (S^K). A set of multi-agent simulations compares the performance of agents operating under ADMF in various scenarios to the performance of agents operating under static and random DMFs for the same scenarios.

4. ADMF implementation

The capability of ADMF developed by this research allows agents to dynamically change the way they participate in DMFs to best fit their current situation. The implementation of ADMF requires a computational representation of DMFs, a reasoning process to select desirable DMFs during system operation, and a mechanism for changing DMFs while the system is running. Section 3 has provided the required computational representation. This section presents the desired characteristics of a system meeting the remaining requirements and describes how these characteristics are realized in the experimental system.

4.1. Selecting desirable DMFs

The ADMF requires agents to be able to determine, with reasonable success, what DMF should be preferred in their current situation to achieve the best possible performance. Agents could use various reasoning techniques to perform this determination, ranging from simple rule-based selection to case-based reasoning to selection based on a reinforcement-learning paradigm. In general, the agents should select the DMF that provides the most improvement in expected performance over the DMF that is already in place. However, a tradeoff usually exists between (1) the performance improvement expected under a different DMF and (2) the costs associated with abandoning the current DMF and establishing a new DMF. These costs may include communication costs as well as any penalties associated with dissolving a current DMF (see discussion of DMF commitments below in Section 4.3). Therefore, once a new potential DMF for a particular situation is identified, agents should use some form of cost-benefit analysis to determine whether a change should be made. Beyond this cost-benefit reasoning, the agent must consider the feasibility of the potential DMF, that is, whether the DMF can even be established in the current situation. Factors to consider in this evaluation may include (1) whether communication is available with any other agents involved in the potential DMF (so they can agree to participate) and (2) an estimate of whether or not the other agents involved in the potential DMF would be likely to agree to participate in that DMF. Even when the benefits of a DMF change outweigh the costs, an agent may decide not to pursue the change if it is not feasible.

For the experiments presented in this article, agents have access to a case-base of previously recorded performance measurements corresponding to each possible situation encountered and each possible DMF they can establish. This exhaustive case-base

identifies the DMF with the optimal performance in each situation. From this previously recorded data, agents can determine the expected value of system performance in their current situation for each possible DMF. The benefit of a potential DMF is calculated as the difference in expected performance between the potential and currently established DMFs. Benefit is normalized by the maximum absolute value of expected performance. The cost of changing to a new potential DMF is calculated as the normalized weighted sum of DMF commitment penalties (see Section 4.3) as well as the cost of resources (e.g. communication bandwidth for message passing) used to make the transition. The weights in this sum can be used to adjust the relative importance of one cost component over the others and to adjust the relative importance of the cost components versus the normalized benefit measure. The net benefit is calculated by subtracting cost from benefit. For these experiments, the estimated feasibility of establishing and potential DMF is 1 if communication with all involved agents is available and 0 otherwise. The feasibility measure acts as a gating function on the desirability of the potential DMF, where

$$desirability = feasibility(benefit - cost).$$

Using the mechanism for DMF change presented in the upcoming section, an agent will attempt to make a change to the potential DMF with the greatest *desirability* rating greater than 0. If no change to a potential DMF has a *desirability* rating greater than 0, the agent will remain in the currently established DMF for that goal.

This method of estimating expected benefit, using such a large set of previously recorded performance measurements, should be interpreted as part of the experimental design for exploring the motivation for ADMF as described in Section 6.2 rather than as a suggested selection mechanism for use in deployed systems. Additional experiments reported in [35] show that using a simple rule base to estimate benefit can produce results similar to those reported here. Therefore, a large amount of prior knowledge is not a requirement for the implementation of ADMF.

4.2. Mechanism for DMF change

The ADMF requires that agents be able to change the DMFs in which they participate as the system operates. The defined mechanism to meet this requirement is straightforward and uses the DMF representation defined in Section 3.1. Each agent maintains an intentional model that describes its own intended goals as well as its beliefs about the intended goals held by other agents. DMF instances can be applied to the goals in this model. This process is called making a *DMF assignment*. Once a DMF has been assigned to a given goal or set of goals in the agent's intentional model, the agent is then constrained to make decisions, with respect to that goal or set of goals, using the now-established DMF. By making different DMF assignments to goals as the system operates, agents can dynamically change the DMFs in which they participate.

Because DMFs may involve more than one agent, steps must be taken to ensure that multiple agents can model any and all DMF changes coherently. In order for productive decision-making interaction to occur, each agent participating in a given DMF must have a view of the DMF that is consistent with the views of all other participating agents. By using *DMF agreements* agents can come to a mutual understanding about collaborative DMFs. The DMF agreement process developed by this research

allows agents to form a mutual understanding that a particular DMF instance has been established. A brief description of the communication protocol is provided here:

Agents use instances of the DMF representation developed in Section 3.1 to communicate to other agents which DMF they desire to establish at any given time. If more than one agent participates in a potential DMF, a complete specification of this DMF is communicated to these agents using the DMF representation. A three-phase protocol is used in which the first step involves proposing and counter-proposing desired DMFs, the second step involves accepting or rejecting these proposals based on desirability ratings, and the final step involves a confirmation of success or failure to establish a DMF. A single agent is identified as the conversation manager, and this agent handles the coordination among multiple agents for a given negotiation. Once a particular DMF instance has been agreed upon through the DMF agreement protocol, the associated DMF assignments are made in each participating agent's intentional model.

There is no requirement for the DMF agreement protocol to establish perfect common knowledge about DMF changes. The agents are able to propose/accept/confirm and then go forth under the assumption that all parties understand the agreement. If this assumption breaks down at some point (i.e. communication failure before the receipt of a confirm message), then the agents can recover by using ADMF to establish a new DMF or try again to establish the same DMF.

The communication protocol for forming DMF agreements is fully documented in [6], including expected messages and protocol properties. This communication protocol is specifically designed to support conversations among two *or more* agents participating in a given negotiation to establish a DMF. The requirement to support more than two agents participating in a given negotiation arises whenever agents desire to establish a DMF in which more than two agents participate, which happens often in systems with three or more agents.

4.3. Managing commitments to DMFs

Allowing agents to establish DMFs through DMF agreements does not ensure that these agents will actually participate in any DMF once it is established. Just as agents must form a commitment to their goals to ensure pursuit of these goals, this research proposes that agents should model and enforce commitments to their established DMFs. Overall, the concept of agent commitment allows agents to form expectations about the behavior of other agents. The DMF commitment motivates agents to actually participate as specified in their established DMFs. By committing to a DMF as it becomes established, an agent agrees to participate, as specified, in the DMF or to pay a penalty for reneging on the established DMF.

Agents must also pay a commitment penalty for dissolving an established DMF in order to form a new one. Since only one DMF can be established for a goal at any given time, to establish a new DMF, agents must concurrently dissolve any conflicting DMFs. Only the agent(s) initiating the dissolution of an established DMF must pay the penalty for reneging on its DMF commitment. Other agents participating in that DMF are not penalized. For the current ADMF implementation, any agent can completely dissolve an established DMF simply by informing the other involved agents (through messages associated with the DMF agreement protocol) that it is no longer participating. No further agreement or permission to dissolve a DMF is

required. Each agent is bound only by its DMF commitment to participate in an established DMF.

Commitment penalties can be implemented with or without requiring explicit payments. In the ADMF implementation used for this article's experiments, the commitment penalties are considered implicitly by including them in the cost of establishing a new DMF. The penalty associated with reneging on a commitment increases with the level of the commitment. Variable levels of commitment have been discussed previously for use in multi-agent systems and extended contract net protocols [21, 44]. DMF commitment is modeled as an integer, c, in the range 0–4, inclusive. A commitment value is associated with each DMF assignment (one DMF assignment is made by each agent involved in any established DMF). The integer, c, represents the cost for the agent of breaking the associated DMF commitment (reneging on the established DMF agreement). For the purposes of the experiments presented in this paper, $c = 1$ for every established DMF that involves only one agent, and $c = 2$ for every established DMF that involves two or more agents.

In general, agents can be designed to favor strong or weak DMF commitments. Stronger commitments (higher c) result in more stable DMFs over time because penalties are larger (agents are less willing to make a DMF change). Modeling DMF commitments and their associated costs allows agents to account for tradeoffs related to the stability of the organizational structure versus changing DMFs.

5. Experimental setup and application domain

This section describes the setup for the experiments reported in this article and maps the analytical representations provided in Section 3 to concrete examples in the experimental domain.

5.1. Experimental domain and agent responsibilities

The experiments presented in this article are performed using the Sensible Agent Testbed [8] in the application domain of Naval Radar Interference Management (NRIM). A *naval radar* is modeled as a radar on board a military ship. Radar *interference* is any form of signal energy detected by a radar that comes from some source other than a reflection of its own emitted wave, but which is indistinguishable from actual return signals. Radar interference decreases the signal to noise ratio of a "victim" radar, thereby making it more difficult for this radar to detect targets.

Each ship in this NRIM application carries one radar, and one agent is associated with each radar. The goal for each agent in the NRIM application for these experiments is to control the frequency of its radar such that radar interference in the system is minimized.

Radar interference occurs primarily when two radars are operating in close geographical proximity at similar frequencies. Interference experienced by each radar is calculated using a straight-line approximation of the typical distance-frequency relationship shown in Fig. 6. Radar interference increases along the diagonal arrow in the figure. The environment simulator therefore models levels of interference along a linear scale from 0 (at and beyond the threshold line in Fig. 6) to 100 (an arbitrary value for maximum interference at the origin in Fig. 6).

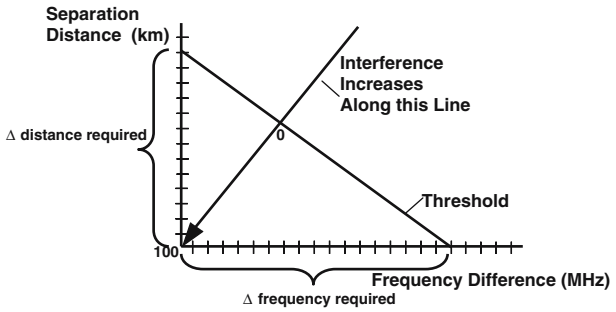


Fig. 6 Straight-line approximation model for interference

The total interference experienced by a victim radar is calculated as the sum of interference experienced from every possible source. The maximum value for interference experienced from one source radar is 100.0. However, because there is no upper limit on the number of sources of interference for a given victim, there is no upper limit on the total amount of interference that can be experienced by a given radar. The total amount of interference in the system, *system interference*, is defined as the total amount of interference experienced by all radars in the system.

As an extension to this basic frequency/distance separation model for interference, the environment simulator for this NRIM application also models directional interference. Radar characteristics can differ from one radar to the next, and radar interference is not necessarily reciprocated. The environment simulator uses a simplified model of non-reciprocal interference by representing *interference reciprocation* as "regular" (uniform in all directions) and "north" (radars interfere only with other radars that have a greater position value along the y-axis as defined for the simulator's model of the environment).

Systems consisting of three agents are used for the experiments reported here. (Summaries of additional experiments investigating 4–8 agent systems [35] are also discussed.) One radar per ship is modeled, and one agent controls each radar's frequency assignment. For these experiments, system radars are always arranged geographically with a regular distribution around a central point on a two-dimensional "ocean" surface. Figure 7 shows this geographical configuration for three agents with a central non-agent entity.

Each system ship is located at distance r from the center of the configuration. A non-system ship (shown smaller than system ships in Fig. 7) is located at the center. This ship represents a *non-agent entity* which controls a radar whose frequency is not

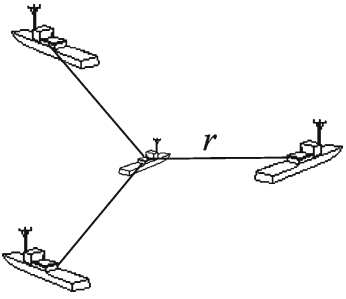


Fig. 7 Relative geographical positions of radars for experiments

controlled by an agent in the system (approximates a fishing boat or other source of non-system radar energy). The radar strength associated with the non-agent entity is half the radar strength associated with agent-controlled radars.

In the system considered by the experiments presented here, issues of safety, trust, and capability differences are eliminated by considering a system of homogeneous software agents with no human participation, thus showing that the ADMF capability can be motivated by performance improvements alone.

5.2. DMFs explored by experiments, *F*

For the experiments presented in this article, *F* contains five elements. These *glb*DMFs apply to the three goals in the system, one intended by each agent of the three agents, as shown in Table 4. The set, $F = \{\text{AllLA, AllM/CD, AllCN, 2M/CD1LA, 2CN1LA}\}$. Refer to Table 3 for the DMF representations used to apply these *glb*DMFs.

The following paragraphs describe variations in the agents' decision-making behavior for this experiment given established DMFs resulting in particular decision-making interaction styles. The concept of a *processing phase* is used in these descriptions to make the point that the agent is constantly and repeatedly attempting to achieve its goals to minimize interference. During a single processing phase, an agent may attempt (once) to determine an acceptable way to achieve its goals. Each of an agent's processing phases is separated by three other periods in this order: (1) a period for actions to be taken, (2) a period for the environment to be updated, and (3) a period for sensor readings to be taken.

Locally Autonomous (*LA*): An agent who is using a locally autonomous decision-making interaction style makes decisions alone about how to manage the frequency of its own radar. If the agent is locally autonomous for its frequency-management goal, it must select a frequency for its own radar. Because changing frequencies is a more expensive action than remaining at the same frequency, an agent has a one in four chance of choosing to remain at the same frequency setting for its own radar during any given processing phase if it is locally autonomous, even if it is experiencing interference. In general, these agents attempt to choose frequencies that minimize system interference. However, to help prevent the system from getting stuck at a local minimum in the global solution space, agents may choose a solution that increases interference slightly in hopes of even better future solutions. Specifically, the agent has a one in two chance of adopting locally autonomous solutions for its own radar if that solution results in not more than 15% greater system interference than the current solution. Only one agent-processing phase is required for a locally autonomous agent to decide on a new frequency and implement the associated frequency change.

Master/Command-driven (*M/CD):* Only the master makes decisions in a master/command-driven relationship. A command-driven agent simply waits for task

Table 4 Agent goals for a 3-agent system

Goal	Goal name	Intended by
$g_1^{a_1}$	Minimize_interference_thru_frequency_management a_1	Agent a_1
$g_1^{a_2}$	Minimize_interference_thru_frequency_management a_2	Agent a_2
$g_1^{a_3}$	Minimize_interference_thru_frequency_management a_3	Agent a_3

allocations from its master. During every processing phase, an agent acting out a "master" decision-making interaction style (a "master agent") attempts to find better frequencies for every agent's radar for which it is making decisions. If every agent's radar that a master agent is making decisions for has already zero interference, then the master agent will not attempt to select new frequencies. Because there is more central control in this *glb*DMF, a master agent will not attempt to avoid local minimum's in the global solution space by adopting any solution that results in greater or equal system interference compared with the current level of system interference. Once the master decides to adopt a solution, it must (1) implement its own frequency change, if any, and (2) send messages to all of its command-driven agents ordering them to change their frequencies to adopt the new solution. Command-driven agents must carry out these orders or pay the penalty for reneging on the DMF agreement that established the master/command-driven relationship. The implementation of an adopted solution takes two agent processing phases in a master/command-driven relationship. During the first processing phase, the master agent can determine a solution, change its own frequency, and send messages. During the second phase, the command-driven agents will receive these messages from the master agent and the command-driven agents will implement their own frequency changes.

True Consensus (*CN*)*:* Each agent involved in consensus interaction plays an equal part in determining frequency assignments for the agent's radars associated with every decision maker. For these experiments, each agent in consensus independently carries out the same process for frequency selection and solution adoption as the master does in a master/command-driven relationship (treating the other agents in consensus as if they were command-driven). However, once a given agent acting in consensus determines a desired solution, it sends a message to all other consensus agents proposing its set of frequency assignments to the group. Along with this solution proposal, each agent sends a message to every other agent in the consensus group describing its view of the declarative model for its own agent (including position information, which may be more accurate than other agents' models). An agent receiving this model information will update its own agent's model of the sending agent. Once a given agent has received proposed solutions and model updates from all agents in the consensus group, it chooses, as its preferred solution of all those proposed, the set of frequency assignments that results in the lowest system interference according to its models. An agent may also decide to prefer the current system state (i.e. rejecting all solutions). Each consensus agent sends a message containing a vote for its preferred solution to all other agents in the consensus group. Each member of the consensus group uniformly adopts the solution with the most votes, and each agent in the group implements the frequency assignment in this solution associated with its own radar. Voting ties are broken deterministically. At least three agent-processing phases are required to realize a consensus solution. During the first processing phase, each agent can determine its own solution proposal and send messages to other agents detailing this proposal and its model of its own radar. During the second processing phase, each agent can use the model updates to determine its preferred solution proposal and send messages to other agents relaying its vote. During the third phase, each agent can tally all the votes and implement the winning solution.

Each decision-making interaction style described here has strengths and weaknesses, which can be identified based on a qualitative evaluation of these descriptions. Observations about different DMFs, such as those that follow, are similar to the observations that initially motivated the investigation of the ADMF capability, described

in Section 2.6. Locally autonomous behavior results in the fastest possible solution adoption (one processing phase), but has limited coordination with other agents. Master/command-driven operation is almost as fast as locally autonomous behavior (two processing phases) and enforces a more global solution approach, but it relies heavily on communication and on the accuracy of the world model maintained by the master agent. A true consensus framework is the slowest to adopt a solution (three processing phases) and most reliant on communication, but it also allows more agents to suggest solutions, allows agents to share information more readily as peers with two-way communication during the decision-making process, and, as a result, allows agents to possibly discover a better global solution.

The processes presented here and used by this research to reach solutions under each type of decision-making style are not the only possible decision-making implementations for each decision-making interaction style. For example, consensus agents could use a negotiation paradigm rather than a voting paradigm. The decision-making implementations chosen for this research highlight the differences among types of DMFs. The remainder of this article shows that when significant differences do exist in the performance of available decision-making frameworks across situations, the capability of ADMF can improve system performance. Determining the most appropriate decision-making implementation within a given framework is left to the individual system designer.

5.3. Situations defined for experiments, X

The defined state space for possible system states for these experiments contains four dimensions, $X = \{X^1, X^2, X^3, X^4\}$. These four dimensions address, respectively, (1) communication availability in the system, (2) position-sensing capabilities of the agents, (3) radar interference reciprocation profiles, and (4) distance of the radars (r) from the center of their geographical configuration. Therefore, $X = \{\text{comm, possen, recip, radius}\}$.

$X^1 = \text{comm}$ refers to "communication" and contains two possible values, UP and DOWN. Therefore, $|X^1| = 2$, and the situation variable $x^1 \in \{x^{1,0}, x^{1,1}\} = \{\text{UP, DOWN}\}$. A value of $x^1 = \text{UP}$ indicates that every agent can communicate with every other agent, and a value of $x^1 = \text{DOWN}$ indicates that no agent can communicate with any other agent.

$X^2 = \text{possen}$ refers to "position sensing" and contains two possible values, YES and NO. Therefore, $|X^2| = 2$, and the situation variable $x^2 \in \{x^{2,0}, x^{2,1}\} = \{\text{YES, NO}\}$. A value of $x^2 = \text{YES}$ indicates that every agent can sense the positions of other radars in the system with 100% certainty. A value of $x^2 = \text{NO}$ indicates that every agent senses the positions of other radars in the system as "very far away" with 0% certainty.

$X^3 = \text{recip}$ refers to "interference reciprocation" and contains two possible values, REGULAR and NORTH. Therefore, $|X^3| = 2$, and the situation variable $x^3 \in \{x^{3,0}, x^{3,1}\} = \{\text{REGULAR, NORTH}\}$. A value of $x^3 = \text{REGULAR}$ indicates that radar interference is reciprocated symmetrically. A value of $x^3 = \text{NORTH}$ indicates that radars interfere only with other radars that are located geographically to the north or at the same north-south position.

$\text{X}^4 = \text{radius}$ refers to distance from center. $\text{X}^4 = \text{radius}$ contains nine possible values which are chosen to give a range of problem difficulty (from most difficult at $r = 0.0$ to least difficult at $r = 58.0$ where radars are so far apart that every frequency

Table 5 Summary of possible state space for experiment situations for three agents

Name	X^i	Possible values for x^i
comm	X^1	$x^1 \in \{\text{UP}, \text{DOWN}\}$
possen	X^2	$x^2 \in \{\text{YES}, \text{NO}\}$
recip	X^3	$x^3 \in \{\text{REGULAR}, \text{NORTH}\}$
radius	X^4	$x^4 \in \{0.0, 5.0, 9.0, 10.0, 27.0, 36.0, 44.0, 49.0, 58.0\}$

assignment is a zero-interference solution). Therefore, $|X^4| = 9$, and the situation variable $x^4 \in \{x^{4,0}, x^{4,1}, x^{4,2}, x^{4,3}, x^{4,4}, x^{4,5}, x^{4,6}, x^{4,7}, x^{4,8}\}$.

Given these specifications, the total state space for situations is $X=\{X^1, X^2, X^3, X^4\}$. The cardinality of X, $|X|$, is therefore 2*2*2*9 = 72. Table 5 summarizes the possible state space.

5.4. Situation changes over time

As defined above, in Section 3.5, a scenario, S^K, is composed of a sequence of K situations, x, where each $x_k \in X$. Each situation, x_k, exists for a time interval, τ_k^N, of fixed length N. For these experiments, transitions from x_k to x_{k+1} in a given scenario are controlled by fixed transition probabilities or fixed transition rules. The transition probabilities and rules chosen for these experiments for each situation state variable yield a uniform distribution over the situation state space. Table 6 shows the

Table 6 Transition probabilities and rules for situation components

Situation component	Elements	Transitions
$X^1 = \text{comm}$	$x^1 \in \{\text{UP}, \text{DOWN}\}$	Transition probability is $\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$
$X^2 = \text{possen}$	$x^2 \in \{\text{YES}, \text{NO}\}$	Transition probability is $\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$
$X^3 = \text{recip}$	$x^3 \in \{\text{REGULAR}, \text{NORTH}\}$	Transition probability is $\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$

$X^4 = \text{radius}$	$x^4 \in \{x^{4,0}, x^{4,1}, x^{4,2}, x^{4,3}, x^{4,4}, x^{4,5}, x^{4,6}, x^{4,7}, x^{4,8}\}$	Transition rules defined by if
		(*in_out_flag* == OUT)
		if ($x^4 \neq x^{4,8}$)
		$x^4 = x^{4,\text{current}+1}$
		else
		$x^4 = x^{4,8}$
		in_out_flag = IN
		end
		else
		if ($x^4 \neq x^{4,0}$)
		$x^4 = x^{4,current-1}$
		else
		$x^4 = x^{4,0}$
		in_out_flag = OUT
		end
		end

chosen transition probabilities or rules for each situation state variable. Transition probabilities related to X^1, X^2, and X^3, respectively, indicate that the state variables x^1, x^2, and x^3 are binary random variables (e.g., for comm, $x^1 = \text{UP}$ or $x^1 = \text{DOWN}$ is chosen by a coin toss for every situation change). The radius state variable x^4 changes such that ships move out from the center of their geographic configuration, $x^{4,0}$, to the outer limit of their radius, $x^{4,8}$, then back to the center, and so forth. Over long scenarios, this transition rule allows agents to spend an equal amount of time at each radius.

These transition probabilities and rules are used to change the situations faced by agents in the scenarios through experimental control. At the end of every time interval, τ_k^N, in a given scenario S^K, a new value is chosen for each situation variable using these transition probabilities and rules. In addition to the changes of the situation variables, the radar frequencies in the system are reassigned to random frequencies at the beginning of every new situation. This prevents the system from settling into a steady state and forces the agents to continue problem solving during each situation encountered.

For these experiments, agents cannot predict when a situation change will occur. Therefore, the situation change x_k to x_{k+1} in S^K must trigger the *glb*DMF change f_k to f_{k+1} in A^K. In addition, because many DMF changes require the establishment of DMF agreements among agents, the change f_k to f_{k+1} may occur several time units after the change x_k to x_{k+1}. For these experiments this lag-time is assumed to be negligible because each situation lasts orders of magnitude longer than this lag time.

5.5. System performance

System performance in this domain is measured as Average System Interference (ASI) the average amount of system interference per unit time per agent. The ASI performance measure is of the general form $p(x, f, i)$ described in Section 3.4. System interference is the sum of interference experienced by all agent-controlled radars in the system. The experimental results for each observation, $asi(x, f, i)$, reflect the amount of total system interference experienced by agents averaged over the entire simulation execution time for a given situation ($N = 150$). The measure $asi(x, f, i)$ directly assesses how well the agents are achieving their goal to minimize interference through frequency management. The function $asi(x, f, i)$ should be *minimized* for best performance. Therefore, the performance measurement function of the form $p(x, f, i)$ is a "penalty" function in this case, in the sense that higher values will correspond to worse performance. The expected performance $asi(x, f, i)$ for a given *glb*DMF, f, in a given situation, x, can be estimated by taking an average over several observations $asi(x, f) = \sum_{i=1}^{m} asi(x, f, i)$. The average system interference per unit time that agents incur while operating across an entire scenario, of the form $P\left(S^K, A^K, I^K\right)$, is $ASI\left(S^K, A^K, I^K\right)$.

6. Experimental results

An analysis of the results from multi-agent simulation in the NRIM domain support the experimental hypotheses introduced in Section 1.2.

6.1. Significant performance differences across DMFs

To explore whether the *glb*DMF under which agents perform best differs significantly across situations, we recorded the performance of agents operating under each possible *glb*DMF in each possible situation defined for the system. For each situation/*glb*DMF case, the performance was observed for 200 simulations using different initial conditions (different initial frequencies assigned to each radar and different initial random seeds for the planner used to select random frequencies). Given the situation space, X, and the *glb*DMF space, F, described in the previous section, a total of $|F|\,|X| = 5 \times 72 = 360$ cases were simulated.

Figure 8 presents a description of the relative performance of each *glb*DMF in each situation. Although the absolute performance differences are also reported in [35], a relative description is the preferred view because the absolute performance for any single *glb*DMF across initial frequency assignments, i, can vary more than the absolute performance across *glb*DMFs within a single initialization for the same experimental case. For example, if the initial frequency assignment is such that interference is already minimized, then all *glb*DMFs can perform equally well and otherwise significant performance characteristics can be masked. Therefore, the performance values should be corrected for variation related to problem difficulty. We use $h(x,f,i)$, the normalized difference from the average performance for each initialization, to make this correction as follows: let $h(x,f,i) = \frac{p(x,f,i)-q(x,i)}{q(x,i)}$, where $q(x,i) = \frac{1}{5}\sum_{j=1}^{5} p(x,f_j,i)$, the mean performance for a given initialization, i, and a given situation, x, across all possible *glb*DMFs.

The expected normalized relative performance in each situation/*glb*DMF case, $h(x,f)$, is calculated from 200 observations across different initial conditions, $h(x,f) = \sum_{i=1}^{200} \frac{p(x,f,i)-q(x,i)}{q(x,i)}$. The graphs in Fig. 8 present these results for $h(x,f)$, plotted for each possible combination of situations ($x \in X$) and *glb*DMFs ($f \in F$). The relative penalty values for ASI, where $p(x,f,i) = asi(x,f,i)$, are shown for all situation/*glb*DMF cases ($|F|\,|X| = 360$), and each point on the graphs represents one such situation/*glb*DMF case, (x,f). Because $asi(x,f)$ should be minimized for best performance, the lowest values of $h(x,f)$ also reflect best performance. Negative values of $h(x,f)$ are reported for *glb*DMFs that performed *better* than average. Due to visualization constraints, the situation/*glb*DMF cases are plotted on eight separate graphs. Recall that the situation state space $x \in X$ has four sub-spaces {comm, possen, recip, radius}. Each individual graph presents values corresponding to one possibility for comm, possen, and recip, and all nine possible elements of radius.

The graphs in Fig. 8 show a great deal of performance variation across *glb*DMFs for a given situation. The mapping for $bestDMF(x)$ can be read directly from these graphs, identifying the *glb*DMF corresponding to the minimum value of $h(x,f)$ in each situation (compared vertically across *glb*DMFs for each value of r in each graph). The $bestDMF(x)$ varies among All LA, All CN, All M/CD, and 2M/CD-1LA, depending on the situation. Therefore, these results support the experimental hypothesis by showing that $bestDMF(x_1)$ does not equal $bestDMF(x_2)$ for all x.

Each point on the graphs in Fig. 8 reflects a mean value. The 95% confidence intervals for each point (based on a standard error calculation [37]) are on the order of the line width or the size of the marker symbol for each point. Therefore, the average difference is statistically significant in most cases, and the *glb*DMF that performs best on average does differ significantly across situations. These results support the

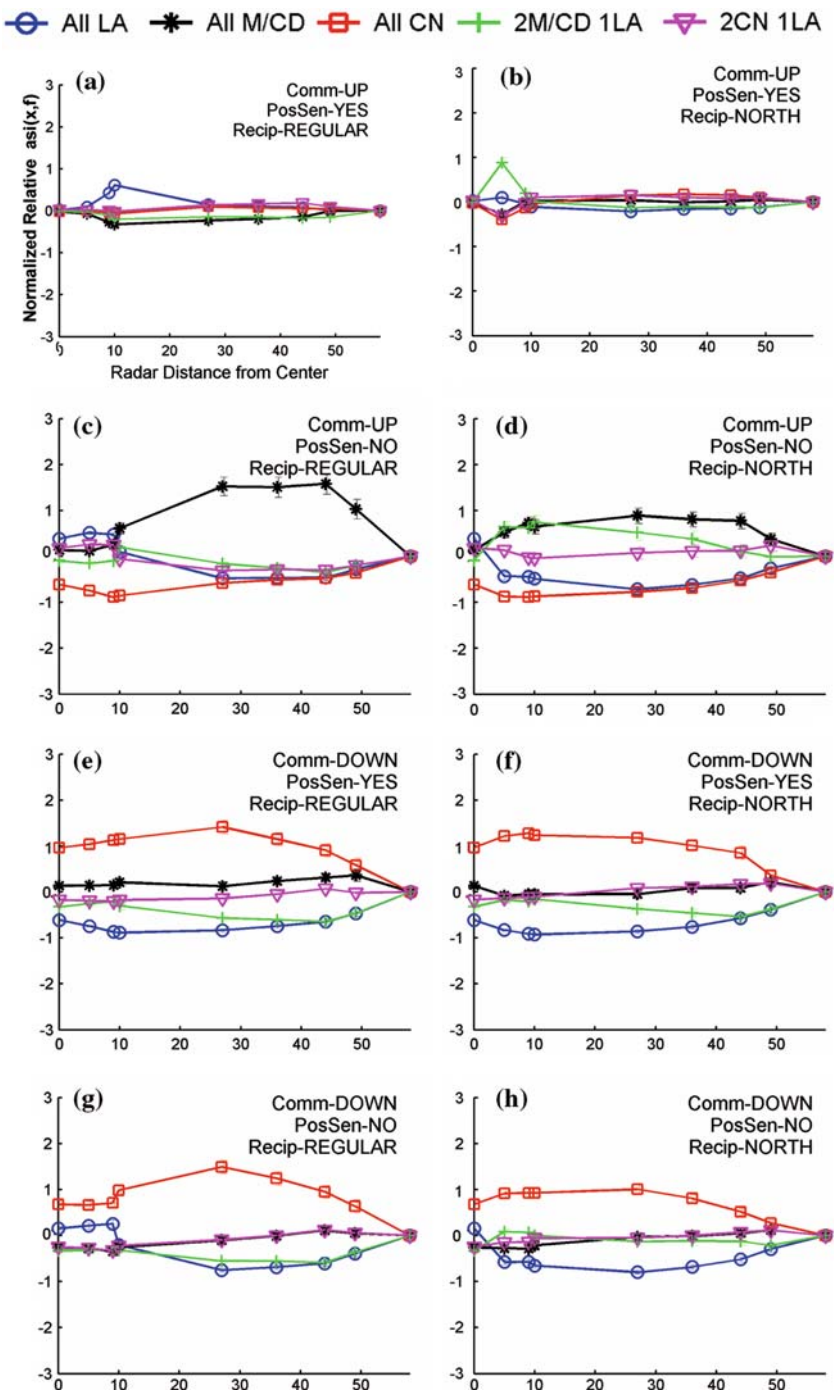


Fig. 8 Normalized relative ASI for each situation and *glb*DMF

Table 7 Percentage of situations in which each global DMF is best

Number of agents	Global decision-making framework (Note: For three agent systems, the CN and M/CD framework is CN and LA)					
	All LA %	All M/CD %	All CN %	M/CD and LA%	CN and M/CD%	None/Tie %
Three agents	44	4	13	13	0	26
Four agents	44	1	18	7	0	29
Five agents	33	0	19	0	8	39
Six agents	29	4	21	0	11	35
Seven agents	47	4	21	1	4	22
Eight agents	31	6	21	3	4	36

experimental hypothesis, *the global decision-making framework under which agents perform best differs as the situation that the agents encounter differs.*

Similar results were observed with systems ranging from 4 to 8 agents [35]. Table 7 shows the percentage of situations in which each *glb*DMF performs best as the number of agents varies. The only significant trend is that as the number of agents in the system increases, the All CN *glb*DMF tends to become more prevalent as the best *glb*DMF in more situations. However, the magnitude of the increase is small, and the situational differences seem to dominate (e.g., between All LA when communication is down and All CN when communication is available).

The results verify the assumption that the DMF can be a controlling variable in the performance of a multi-agent system in a given situation, even though these performance results do not generalize directly to other systems with other situation types and other decision-making implementations. For example, in another system, a poorly performing negotiation mechanism may be used for consensus decision-making, such that any DMF with more than one agent acting as a decision maker (in *D*) will always perform poorly. The results presented here could not be used to predict that outcome and should not be seen as a prescription for what kind of DMF performs best in any particular type of situation for another system. However, the results do show that given a set of characteristics and decision-making implementations for a particular system, the DMF can be a controlling variable in the system's performance.

6.2. ADMF performance across changing situations

These results provide motivation for implementing ADMF by highlighting the opportunity to improve performance in a given situation through changing DMFs. The ADMF capability may allow agents to change from a *glb*DMF that does not perform well in a given situation to a *glb*DMF that performs better. However, the question remains whether this potential performance improvement will outweigh the overhead and relative system instability associated with changing *glb*DMFs. These costs are reflected in the ASI performance measure by elevated interference levels that continue to exist as agents abandon solutions that are forming under one *glb*DMF to start over forming solutions under another *glb*DMF.

The next step is to show whether these performance differences can be exploited to improve overall run-time performance across dynamically changing situations. To do so, another set of multi-agent simulations compares the performance of agents

operating under ADMF in various scenarios to the performance of agents operating under static and random decision-making frameworks (as defined in Section 3.5) for the same scenarios.

Recall from Section 3.5 that ADMF is defined analytically as a policy for changing the *glb*DMF, f, in effect from one time, k, to another, where $f_{k+1} = f$: $best_{f \in \left\{\hat{\mathrm{F}}(x_{k+1}) \cup f_k\right\}} \left(p(x_{k+1}, f_{k+1})\right)$. Using the previously recorded performance observations across all situations and *glb*DMFs as a case base, the *glb*DMF to select in a given situation under ADMF can be defined as $f_{k+1} = f : \min_{f \in \left\{\hat{\mathrm{F}}(x_{k+1}) \cup f_k\right\}} \left(asi(x_{k+1}, f_{k+1})\right)$. This equation indicates that the achievable *glb*DMF that gives the minimum expected ASI, for the time interval when situation x_{k+1} exists, is chosen as f_{k+1}.

Recall also from Section 3.5 that a scenario S of length K, S^K, is defined as a scenario consisting of K time intervals, τ_k^N, such that $S^K = (x_1, x_2, \ldots, x_K)$. Twenty scenarios with $k = 27$ and $N = 150$ were randomly generated according to the transition rules given in Section 5.4, and each DMF policy was simulated for each of these scenarios. Figure 9 shows the results for these simulations, comparing DMF policies with respect to the observed values of ASI, the average system interference across the scenario per unit time. The ASI values are normalized by the largest observed value of ASI.

Since ASI should be minimized, the DMF policies with lower values of ASI perform better. The bars in Fig. 9a show that, on average, the ADMF policy out-performs every static DMF policy as well as the random DMF policy. The "**x**" marks across each bar in Fig. 9a represent the actual data values for normalized ASI for each of the twenty scenarios tested. The distribution of these "**x**" marks directly reflect the distribution of the data, and the scatter in the data values arises from varying difficulty across scenarios. Figure 9b repeats the information in Fig. 9a and connects each data point for individual scenarios tested, across DMF policies. Using this relative penalty information for each individual scenario, one can see that the ADMF policy outperforms each of the other policies in every single scenario tested, as well as on average. Therefore, the scatter in the data does not reduce the significance of the performance improvement realized by ADMF.

Figure 10 presents a clearer view of the relative performance of the ADMF policy compared to the performance of all other policies. For this chart, all reported values from Fig. 9 are divided by the observed ASI for the ADMF policy in the same scenario. Again, the "**x**" marks indicate the distribution of the data. Figure 10 shows that ADMF clearly outperforms the other DMF policies. These results clearly support the experimental hypothesis that *agents operating under ADMF can perform significantly better than agents operating under static decision-making frameworks given run-time situation changes.* Similar results were observed for system ranging from 4 to 8 agents [35]. Increasing the number of agents in the system, up to eight agents, does not seem to reduce the significance of the performance improvement gained by ADMF.

7. Conclusions

This article shows that adding the capability of ADMF to a multi-agent system can result in significantly improved system performance across run-time situation changes. Specifically, ADMF can result in improved and more robust performance over the use of a single static DMF. The results reported in this article first show that the

DMFs used by agents during system operation can be a controlling variable in the performance of a multi-agent system. By definition, a DMF identifies a set of agents and specifies the set of interactions exercised by these agents as they determine how a goal or set of goals should be achieved. This article provides a computational representation for DMFs, and shows that shows how they can be manipulated by the ADMF capability. The experimental results show that one DMF does not perform best for all situations. The reported results further show that this varying performance of DMFs across situations can be exploited to improve overall run-time performance if agents are enabled to switch from an existing DMF to one that performs better in the current situation. The experiments show that dynamically adapting DMFs to changing situations through ADMF improves overall system performance across changing

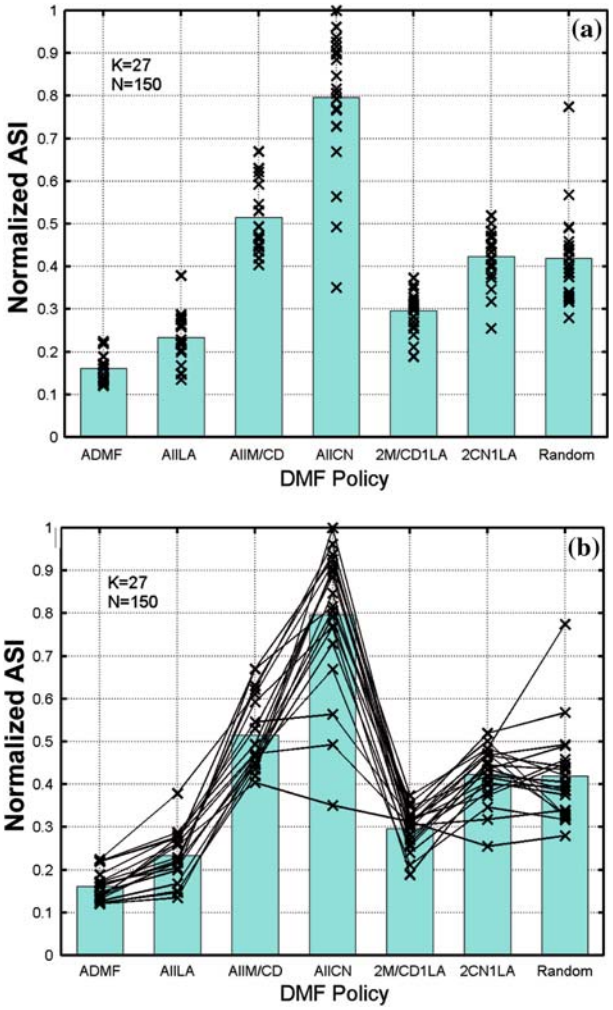


Fig. 9 The ASI compared across various DMF policies for the same 3-agent scenarios

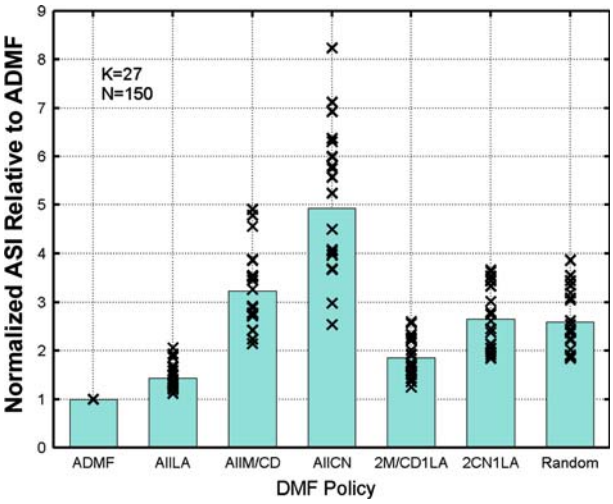


Fig. 10 Average System Interface of DMF policies relative to ADMF for 3-agent scenarios

situations, outweighing the overhead and additional system instability associated with making dynamic changes to DMFs.

The empirical results presented in this article therefore justify the implementation of ADMF in multi-agent systems to improve system performance across changing situations. This justification argument assumes that (1) agents encounter dynamic changes in their situation during normal system operation, (2) these dynamic changes do not occur faster than the agents in the system can respond to solve a problem, and (3) the DMF resulting in the best system performance (by a useful margin) varies across the possible situations that may be encountered by the agents. If these conditions hold, then the implementation of ADMF is justified. These characteristics would be required to transfer the conclusions from the presented experiments to another domain. Otherwise, the additional overhead required to support ADMF may not be warranted. However, if these conditions do hold, as they do for many complex, dynamic problem domains in which agents face uncertainty and unreliable communication, ADMF can provide significant performance improvements as indicated by the experimental results presented here.

Acknowledgements This research was supported in part by the Texas Higher Education Coordinating Board (no 003658452) and a National Science Foundation Graduate Research Fellowship.

References

1. Barber, K. S., Han, D. C., & Liu, T. H. (2000). Coordinating distributed decision making using reusable interaction specifications. In C. Zhang & V.-W. Soo (Eds.), *Design and applications of intelligent agents: Third pacific rim international workshop on multi-agents, PRIMA 2000, Melbourne, Australia, August 2000, Proceedings* (pp. 1–15). New York: Springer.
2. Barber, K. S., Liu, T. H., & Han, D. C. (2001) Strategy selection-based meta-level reasoning for multi-agent problem solving. In Ciancarini P., & Wooldridge M. (Ed.), *Agent oriented software engineering*, Vol. 1957, (pp. 269–284). Springer Verlag.

3. Barber, K. S., & Martin, C. E., (2001). Autonomy as decision-making control. In C. Castelfranchi, & Y. Lesperance (Ed.), *Intelligent agents VII: Agent theories architectures and languages* (pp. 343–345). Berlin: Springer.
4. Barber, K. S., & Martin, C. E. (2001). Dynamic adaptive autonomy in multi-agent systems: Representation and justification. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3), 405–433.
5. Barber, K. S., & Martin, C. E. (2001) The Motivation for dynamic decision-making frameworks in multi-agent systems. In Jiming Liu, N. Zhong, Y. Y. Tang, & P. S. Wang (Eds.), *Agent engineering* (pp. 59–91). Singapore: World Scientific.
6. Barber, K. S., Martin, C. E., & McKay, R. M. (2001). A communication protocol supporting dynamic autonomy agreements in multi-agent systems. In R. Kowalczyk, S. W. Loke, N. E. Reed, & G. Williams, (Eds.), *Advances in artificial intelligence: PRICAI 2000 workshop reader. Four Workshops held at PRICAI 2000, Melbourne, Australia, August/September 2000. Revised Papers*, Vol. LNAI 2112 (pp. 303–320). Berlin: Springer.
7. Barber, K. S., Martin, C. E., Reed, N. E., & Kortenkamp, D. (2001). Dimensions of adjustable autonomy. In R. Kowalczyk, S. W. Loke, N. E. Reed, & G. Williams (Eds.), *Advances in artificial intelligence: PRICAI 2000 Workshop Reader. Four Workshops held at PRICAI 2000, Melbourne, Australia, August/September 2000*, Vol. LNAI 211 (pp. 353–361). Revised Papers ed. Berlin: Springer.
8. Barber, K. S., McKay, R. M., Goel, A., Han, D. C., Kim, J., Liu, T. H., & Martin, C. E. (2000). Sensible agents: The distributed architecture and testbed. *IEICE Transactions on Communications. IECIA/IEEE Joint Special Issue on Autonomous Decentralized Systems*, E83-B(5) 951–960.
9. Bharatia, V. A., & Cook, D. J. (1995). Design and analysis of centralized, distributed, and group multi-agents coordination models, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, Technical Report.
10. Bond, A. H., & Gasser, L. (1998). An analysis of problems and research in DAI. In A. H. Bond, & L. Gasser (Eds.), *Readings in distributed artificial intelligence* (pp. 3–35). San Mateo, CA: Morgan Kaufmann Publishers Inc.
11. Briggs, W., & Cook, D., (1995). Flexible social laws. In *Proceedings of the fourteenth international joint conference on artificial intelligence* (pp. 688–693). Montreal, Que, Canada.
12. Brooks, C. H., Durfee, E. H., & Armstrong, A. (2001). An introduction to congregating in multiagent systems. In *Proceedings of the fourth international conference on multiagent systems (ICMAS-2000)* (pp. 79–86). Boston, MA.
13. Castelfranchi, C. (1995) Commitments: From individual intentions to groups and organizations. In *Proceedings of the first international conference on multi-agent systems* (pp. 41–48). San Francisco, CA.
14. Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence, 42*, 213–261.
15. Daft, R. L., & Marcic, D. (1998) *Understanding management* (2nd ed.). Fort Worth, TX: Dryden Press.
16. Davin, J., & Modi, P. J. (2005) Impact of problem centralization in distributed constraint optimization algorithms. In *Proceedings of the fourth international joint conference on autonomous agents and multi agent systems* (pp. 1057–1066). The Netherlands: Utrecht.
17. Decker, K., & Lesser, V. (1993). A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proceedings of the eleventh national conference on artificial intelligence* (pp. 210–216). Washington, DC.
18. Decker, K. S., & Sycara, K. P. (1997). Intelligent adaptive information agents. *Journal of Intelligent Information Systems*, 9(3), 239–260.
19. Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B., & Schreckenghost, D. (1998). Adjustable autonomy for human-centered autonomous systems on Mars, In *Proceedings Mars society conference*, (pp. 397–419). Boulder, CO.
20. Durfee, E. H. (1996). Planning in distributed artificial intelligence. In G. M. P. O'Hare, & N. R. Jennings (Eds), *Foundations of distributed artificial intelligence* (pp. 231–245). New York: John Wiley & Sons Inc.
21. Durfee, E. H., & Lesser, V. R. (1987). Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the tenth international joint conference on artificial intelligence* (pp. 875–883). Milan, Italy.
22. Excelente-Toledo, C. B., & Jennings, N. R. (2004). The dynamic selection of coordination mechanisms. *Autonomous Agents and Multi-Agent Systems, 9*, 55–85.

23. Falcone, R., & Castelfranchi, C. (2000). Levels of delegation and levels of adoption as the basis for adjustable autonomy. In E. Lamma, & P. Mello (Eds.), *AI*IA 99: Advances in artificial intelligence. Selected papers from proceedings of the sixth congress of the Italian association for artificial intelligence, Bologna, Italy, September 1999* (pp. 273–284). Berlin: Springer-Verlag.
24. Fox, M. S., Barbuceanu, M., Gruninger, M., & Lin, J. (1998). An organizational ontology for enterprise modeling. In M. J. Prietula, K. M. Carley, & L. Gasser (Eds.), *Simulating organization*, Menlo Park, CA: AAAI Press/The MIT Press.
25. Gasser, L., Rouquette, N. F., Hill, R. W., & Lieb, J. (1998). Representing and using organizational knowledge in DAI systems. In L. Gasser, & M. N. Huhns (Eds.), *Distributed artificial intelligence*, (Vol. 2, pp. 55–78). London: Pitman/Morgan Kaufman.
26. Haddadi, A. (1995). Towards a pragmatic theory of interactions. In *Proceedings of the first international conference on multi-Agents systems* (pp. 133–139). San Francisco, California.
27. Hirayama, K., & Toyoda J. (1995) Forming coalitions for breaking deadlocks. In *Proceedings of the first international conference on multi-agent systems* (pp. 155–162). San Franciso, CA.
28. Ishida, T., Gasser, L., & Yokoo, M. (1992). Organization self-design of distributed production systems. *IEEE Transactions on Knowledge and Data Engineering, 4*(2), 123–134.
29. Jennings, N. R. (1993). Commitments and conventions: the foundation of coordination in multi-agent systems. *The Knowledge Engineering Review, 8*(3), 223–250.
30. Jennings, N. R. (1993) Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare, and N. R. Jennings (Eds.), *Foundations of distributed artificial intelligence* (pp. 187–210). New York: John Wiley & Sons Inc.
31. Kortenkamp, D., Keirn-Schreckenghost, D., & Bonasso, R. P. (2000). Adjustable control autonomy for manned space flight. In *Proceedings of the IEEE aerospace conference* (pp. 18–25). Big Sky, MT.
32. Lawrence, P. R., & Lorsch, J. W. (1967). *Organization and environment: Managing differentiation and integration*. Boston: Harvard Business School Press.
33. Lerman, K., & Shehory, O. (2000). Coalition formation for large-scale electronic markets. In *Proc. Fourth international conference on multi agent systems (ICMAS-2000)* (pp. 167–174). Boston, MA.
34. Lin, Z. (1998). The choice between accuracy and errors: A contingency analysis of external conditions and organizational decision making performance. In M. J. Prietula, K. Carley, M., & L. Gasser, (Eds.), *Simulating organization*, Menlo Park, CA: AAAI Press/The MIT Press.
35. Martin, C. E. (2001). Adaptive decision-making frameworks for multi-agent systems. University of Texas at Austin, Austin, TX PhD Dissertation.
36. Mertens, P., Falk, J., & Spieck, S. (1997). Comparisons of agent approaches with centralized alternatives based on logistical scenarios. *Information Systems, 19*(8), 699–709.
37. Milton, J. S., & Arnold, J. C. (1990). *Introduction to probability and statistics: Principles and applications for engineering and the computing Sciences* (2nd ed.), New York: McGraw-Hill.
38. Moulin, B., & Chaib-draa, B. (1996). An overview of distributed artificial intelligence. In G. M. P. O'Hare, & N. R. Jennings (Eds.), *Foundations of distributed artificial intelligence* (pp. 3–55). New York: John Wiley & Sons Inc.
39. Musliner, D. J., & Krebsbach, K. D. (1999). Adjustable autonomy in procedural control for refineries. In *Proceedings of the AAAI 1999 spring symposium series: agents with adjustable Autonomy* (pp. 81–87). Stanford University, Stanford, Califonia.
40. Myers, K. L., & Morley, D. N. (2001). Directing agent communities: An initial framework. In *Proceedings of the IJCAI-2001 workshop on autonomy, delegation, and control: Interacting with autonomous agents* (pp. 81–88). Seattle, WA.
41. Noh, S., & Gmytrasiewicz, P. (1999). Implementation and evaluation of rational communicative behavior in coordinated defense. In *Proceedings of the third international conference on autonomous agents (Agents-99)* (pp. 123–130). Seattle, WA.
42. Noh, S., & Gmytrasiewicz, P. (1999). Towards flexible multi-agent decision-making under time pressure. In *Proceedings of the sixteenth international joint conference on artificial intelligence (IJCAI-99)* (pp. 492–498). Stockholm, Sweden.
43. Reed, N., & Scerri, P. (2000). Online control of agents using EASE: Implementing adjustable autonomy using teams. In *Proceedings of the first workshop on teams with adjustable autonomy* (pp. 1–8). Melbourne, Australia.
44. Sandholm, T., & Lesser, V. R. (1995). Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *Proceedings of the first international conference on multi-agents systems* (pp. 328–335). San Francisco, CA.
45. Sandholm, T. W., & Lesser, V. R. (1997). Coalitions among computationally bounded agents, *Artificial Intelligence, 94*(1), 99–137.

46. Scerri, P., Pynadath, D. V., & Tambe, M. (2001). Adjustable autonomy in real-world multi-agent environments. In *Proceedings of the autonomous agents* (pp. 300–307). Montreal, Canada.
47. Scott, W. R. (1992). *Organizations: Rational, natural and open systems* (3rd ed.). Englewood Cliffs, NJ: Prentice-Hall.
48. Sen, S., & Dutta, P. S. (2000). Searching for optimal coalition structures. In *Proceedings of the fourth international conference on multiAgent systems (ICMAS-2000)* (pp. 287–292). Boston, MA.
49. Shehory, O., & Kraus, S., (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence, 101*(1–2), 165–200.
50. Singh, M. P. (1990). Group ability and structure. In Y. Demazeau, & J.-P. Müller (Eds.), *Decentralized A.I. 2: Proceedings of the second european workshop on modelling autonomous agents in a multi-agent world, Saint Quentin en Yvelines, France, August 13–16, 1990* (pp. 127–45). Amsterdam: Elsevier Science.
51. So, Y.-P., & Durfee, E. H., (1998). Designing organizations for computational agents. In M. J. Prietula, K. M. Carley, & L. Gasser (Eds.), *Simulating organizations* (pp. 47–66). Menlo Park, CA: AAAI Press/The MIT Press.
52. Stephens, L. M., & Merx, M. (1989). Agent organization as an effector of DAI system performance. In *Proceedings of the ninth workshop on distributed artificial intelligence* (pp. 263–292).
53. Sycara, K. P. (1997). Multiagent systems. *AI Magazine, 19*, 79–92.
54. Tambe, M. (1997). Agent architectures for flexible, practical teamwork. In *Proceedings of the fourteenth national conference on aritificial intelligence* (p. 1092). Providence, Rhode Island.
55. Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research, 7*, 83–124.
56. Wagner, T., & Lesser, V. (2000). Relating quantified motivations for organizationally situated agents, In N. R. Jennings, & Y. Lesperance (Eds.), *Intelligent agents VI: Agent theories, architectures, and languages* (pp. 334–348). Berlin: Springer-Verlag.