Modeling and Simulating Human Teamwork Behaviors Using Intelligent Agents

Xiaocong Fan, John Yen
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA16802
Email: {zfan, jyen}@ist.psu.edu

Abstract

Among researchers in multi-agent systems there has been growing interest in using intelligent agents to model and simulate human teamwork behaviors. Teamwork modeling is important for training humans in gaining collaborative skills, for supporting humans in making critical decisions by proactively gathering, fusing, and sharing information, and for building coherent teams with both humans and agents working effectively on intelligence-intensive problems. Teamwork modeling is also challenging because the research has spanned diverse disciplines from business management to cognitive science, human discourse, and distributed artificial intelligence. This article presents an extensive, but not exhaustive, list of work in the field, where the taxonomy is organized along two main dimensions: team social structure and social behaviors. Along the dimension of social structure, we consider agent-only teams and mixed human/agent teams. Along the dimension of social behaviors, we consider collaborative behaviors, communicative behaviors, helping behaviors, and the underpinning of effective teamwork—shared mental models. The contribution of this article is that it presents an organizational framework for analyzing a variety of teamwork simulation systems and for further studying simulated teamwork behaviors.

I. INTRODUCTION

Teamwork is joint work toward common performance goals. Researchers from different disciplines have defined "team" from different perspectives. Katzenbach and Smith defined team as "a small number of people with complementary skills who are committed to a common purpose, set of performance goals, and working approach for which they hold themselves mutually accountable [1]." They distinguished teams from working groups, which typically have no significant performance needs, no true interdependency, and no shared accountability. Salas et. al. [2] defined team as "... a distinguishable set of two or more people who interact dynamically, interdependently, and adaptively towards a common and valued goal/objective/mission, who each have been assigned specific roles or functions to perform, and who have a limited life-span of membership." A more operational definition of team is given by Cohen, Levesque and Smith [3] as "a set of agents having a shared objective and a shared mental state". The simplicity of this definition is attributed to the notion of joint intention, which requires an agent to commit to informing other team members whenever it detects the common goal is already achieved or becomes impossible or irrelevant. Drawn from these definitions, the common characteristics of teams include team accountability/responsibility, common objective/commitment, performance monitoring, and within-team interdependence.

Contemporary research on teamwork spans a variety of disciplines, including psychology, cognitive science, artificial intelligence, organization science, concurrent engineering, and business management [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Among researchers in multi-agent systems, there has been growing interest in using intelligent agents to model and simulate human behaviors. To only mention a few: heterogeneous intelligent agents simulating firefighters, commanders, victims and volunteers conduct search and rescue activities in a virtual disaster world [14]; agents model and assist operational cells in Network-centric Warfare [15]; cognitive agents as a model of human operators are employed to interact with simulations of human-oriented systems in order to repetitively evaluate the design of the systems [16]; interface agents, empowered with a model of users' attention, can collaborate better with users [17]; rule-driven autonomous agents are used in RBSim (Recreation Behavior Simulator) to simulate human recreation behaviors [18].

In this article we review the state-of-art of research on modeling and simulating human teamwork behaviors using intelligent agents. But before that, we would like to answer two questions. The first question is why modeling and simulation of human behaviors are needed. Generally, human-behavior simulation is important for at least four reasons. (1) Simulations offer sufficient practice for human training. Simulation tools provide a virtual but realistic environment for human trainees to learn effective resource allocation and activity scheduling, to develop their individual and cooperation skills, to understand and improve their multitasking capacity, to learn how to manage their attentional focus in dynamic domains, and to practice coordination skills during group decision-making process. (2) Simulation is a practical solution to improving readiness and lowering costs. For instance, military training exercises involving real entities are expensive and logistically complex [19]. Simulations can save money and time by cutting the need to deploy actual equipment and forces, which may even involve multiple countries. Simulations are particularly useful now for training military leaders because military organizations can afford fewer training hours and smaller expenditures during field exercises [19]. (3) Simulations can be used for conducting "what-if" scenarios. In complex domains, human users can leverage simulation tools to watch scenarios unfold on-screen, to explore alternative hypotheses, to rehearse an event so that they can be better prepared for surprises, etc. (4) Simulations of what actually occurred can be used for after-the-fact analysis [19]. Noise being removed, simulation data can be further generalized into reusable experience for guiding scenario unfolding in later simulations. The importance of human-behavior simulation is also reflected in the increasing demands of comprehensive simulation tools in site security, disaster relief, hostage rescues, crowd control, police raids, as well as battlefield exercises (military maneuvers). Janus, JTS, and JCATS [19], ModSAF [20], Brahms [21], NSS [22] are representatives of the range of such simulation tools.

The second question is "why use agents to model and simulate human behaviors"? The answer lies in four aspects. (1) Agent technology offers a new paradigm for conceptualizing, designing, and implementing complex software systems. In particular, agents are often implemented using mentalistic notions such as beliefs, desires, and intentions that are more usually applied to humans. Being artifacts with properties like autonomy, social ability (communicating via speech acts protocol) and human-like mental states, agents are the best way to model and simulate human behaviors that require spatially, functionally, and temporally distributed processing. (2) It is well-recognized that aggregated units are difficult to simulate due to the complexity of coordination (e.g., fluid entity relationships within aggregated unit [19]) and information exchange within and between aggregated units. This can be addressed gracefully using multi-agent systems or team-based agents [23], [24]. For instance, TaskableAgents are used to simulate the behaviors of a battalion tactical-operations-center (TOC), decomposing the activities of TOC into multiple agents that represent different functional areas [25]. (3) Usually, modeling and simulating human behaviors are not the ends. Simulation systems need to further *support* human users of these agents in pursuing their analysis goals. Being able to learn from interactions with human users and adapt to dynamic changes, agents can be used for implementing adjustable intelligence to better support humans in their team activities. (4) It is desirable that a well-developed simulation system with adequate intelligence can work seamlessly with real-world equipment or even human peers. The agent approach is a natural choice because agents can not only encapsulate modularized intelligence, they also adopt a societal view of computation [26]. There have been several attempts in using cognitive agents as virtual teammates in mixed human/agent teams [27], [8], [28], [29].

Teamwork behaviors often involve two or more team members. Examples of teamwork-related behaviors include communication (sharing of information or a mental model such as intentions), coaching, collaborative problem solving (e.g., decision making, planning), task allocation (work-load distribution, contracting or delegation), needs anticipation, cross monitoring, and backup behaviors [30]. Rather than cover a wide spectrum of teamwork behaviors, in this paper we choose to focus on four of them: shared mental models, collaborative behaviors, communicative behaviors, and helping behaviors. The remainder of the paper is organized as follows. In Section II and Section III we give an overview on the agent research in the field of artificial intelligence and research about human teamwork behaviors in cognitive science, respectively. Teamwork modeling in agent-only teams and mixed human/agent teams are surveyed in Section IV and Section V respectively through identifying some representative systems in the literature and inspecting how shared mental models, collaborative behaviors, communicative behaviors, and helping behaviors are modeled or implemented in the systems. Section VI briefly discusses the recent trends and future directions regarding the modeling of human teamwork behaviors and Section VII summarizes the paper.

II. Agent Research in Artificial Intelligence

The term "agent" has been identified as a hardware or software system with certain properties such as autonomy, social ability, reactivity and pro-activity [31]. Researchers in different disciplines have used agents to refer to different entities or artifacts. Generally, the different notions of agency in the literature can be classified into four categories: software agent, intelligent agent, robotic agent and cognitive agent.

In mainstream computer science, software agents are taken as software components that communicate with other peers via message passing [32]. Agents in such a sense can be as simple as subroutines or as complex as active objects with complicated computing logics. Research in this area is typically concerned about agent-oriented engineering or methodologies—how to rapidly develop well-behaved large agent systems while avoiding the notorious pitfalls described in [33].

Intelligent agents are often specified, designed and built using mentalistic notions [34], [35]. As intentional systems, they are widely used in the AI field to represent modularized intelligence. Researchers in this field are interested in formally characterizing properties of agents using modal logics [36], in understanding the relationships between different mental attitudes (beliefs, goals, intentions) [37], in investigating how social intelligence can be built on top of individual intelligence, and in exploring how intelligent agents can form coalitions to solve problems collaboratively and effectively, etc.

A robotic agent is an autonomous sensor-motor-control system that is capable of physical perception, manipulation and processing. In this line of work, researchers try to investigate intelligence bottom-up in a behavior-oriented way [38].

Cognitive agents [39] are intelligent human-like systems developed using insights from cognitive science. They represent computational models of human thought and reasoning: they perceive information from the environment, assess situations using knowledge obtained from human experts, and act to affect the external or internal environment. The most notable cognitive agent architectures include Soar [40], [41], ACT-R [42], iGEN [43] and Cougaar [44]. They have been used to develop agents for cognitive tutoring systems, human-computer interaction, and performance support systems.

A group of individual agents form a MultiAgent System (MAS) when they co-exist and act (compete or cooperate) in an environment. Multiagent systems, being a subdiscipline of distributed artificial intelligence (DAI), study networked systems composed of physically or logically distributed agents (humans, robots, computational problem solvers) that can deliberate, predict, communicate, and cooperate. MASs can be characterized as: (1) having no explicit global control, (2) having distributed resource, expertise, intelligence, and processing capabilities, (3) typically working in an open environment full of uncertainties, and (4) emphasizing social agency and social commitments [45].

Modularity, distribution, abstraction, and intelligence are four major techniques for dealing with the size and complexity of domain problems [46]. MAS, which reflects the combination of all four of these techniques, provides both principles for developing complex systems and mechanisms for collaborative problem solving. First, MASs are the best way to specify and/or design systems that require spatially, functionally, or temporally distributed processing [47], [48], [45]. Second, MASs are a natural way to simulate, train, and support collaborative behaviors in human teams. Examples of such domains include Robo-Rescue, Network-Centric Warfare, air-traffic control, and on-line trading. Third, MASs encourage parallelism, robustness, scalability, understandability, maintainability, and reusability [45], [48], which are typically the desired features of system designs from the perspective of software engineering. As a new paradigm for conceptualizing, designing, and implementing software systems [31], MASs have been successfully used in a variety of areas including applications for distributed situation awareness and assessment, distributed resource planning and allocation, distributed expert systems, workflow, supply chain management, Web services composition, and collaborative information processing [47], [23], [24].

Research on agent teamwork in the field of DAI emerged in the early 1990s [7]. Team-based agent systems are a special kind of MASs that emphasize cooperativeness and proactiveness in pursuing certain common goals. The Joint Intentions theory [49], [7] and the SharedPlans theory [50], [51] are two widely accepted formalisms for modeling teamwork; each has been successfully applied in guiding the design and implementation of multi-agent systems, such as GRATE* [52], STEAM [23], COLLAGEN [53], CAST [24].

Teams are high-performing cooperative MASs because team members are often deeply committed both to each other's individual growth and to the whole team's success. But on the other hand, a team requires more cost than a MAS due to maintaining the common vision and mission. In particular, an inherent overhead is associated with establishing, monitoring, and disbanding a team [3]. How to reduce such overhead without sacrificing too much team performance has been and will continue to be a key issue in teamwork research [23], [24].

III. Human Teamwork Behaviors

We review the cognitive and psychological studies regarding human teamwork behaviors from three aspects: communicative behaviors, collaborative behaviors and helping behaviors.

Communication is crucial in high-performing human teams, and it is the basis for developing teamwork skills such as active listening, questioning to clarify uncertainties, elaborating, persuasion, and group decision makings. In general, there exist two styles of communication: reactive and proactive. Reactive communication refers to the ask/reply mode, also known as *master-slave assumption* [54]. Proactive communication refers to the mode where a person provides information without being asked. Both modes are widely used by humans in everyday life. For instance, team members typically tend to seek new information to achieve their joint goals [55]. Some psychological studies have identified one key characteristic of effective human teams as those in which members proactively offer information needed by teammates based on the anticipation of others' needs [56], [57], [58].

Proactive information delivery complements reactive communication in cases where a person does not know whom to ask when information is needed or he/she simply does not realize his/her information needs due to limited knowledge. Proactive information delivery occurs more frequently when human teams need to filter and fuse an overwhelming amount of information and to make critical decisions under time pressure. One of such dynamic domains is Battlespace Infospheres, which often require a large number of intelligent agents and human agents to form a team to cooperate effectively in information gathering, information fusion and information delivering for making better group decisions.

Being able to anticipate teammates' information needs is the key to proactive communication. In the ATOM model proposed by Dickinson and McIntyre [56], "recognize other members' need for certain information" is listed as one of the teamwork dimensions. In a study by Yen, Fan and Volz [59], the concept of information-needs is formally characterized and a framework for anticipating teammates' information needs is provided. The framework connects the anticipated information-needs to potential commitments so that agents can choose appropriate communicative actions to satisfy teammates' information-needs.

The scope of collaborative behaviors is very broad, including collaborative learning, collaborative planning, collaborative decision-making, collaborative conflict handling, and collaborative situation awareness, all of which have been actively investigated. Using collaboration as one of the critical explanatory concepts, Ellis et al [60] examined the effects of cognitive ability, agreeableness, workload distribution, and structure on team learning. This study supported the hypothesis that high levels of agreeableness may actually be detrimental to effective collaboration because premature consensus has a negative effect on group problem solving.

Collaborative (e.g., mixed-initiative) planning is a group activity frequently adopted in human society. For instance, members of a conference organizing committee negotiate about the schedule of conference program; a family makes a plan used to contact each other in an emergency situation; two people collaboratively work out a process for preparing a meal together [50]. Collaborative planning in its nature is incremental, hierarchical and exploratory, and requires a shared understanding of the planning objective and process. Grosz and Sidner adopted a mental-state view of collaborative planning; a group of collaborating agents have a shared plans if and only if they work towards establishing and maintaining a particular collection of beliefs and intentions [54]. In a study by Lochbaum [61], human discourse understanding was investigated from the perspective of collaborative planning, leveraging the notion of SharedPlans [54].

Most command and control teams need to make decisions under time pressure. For instance, the scenario given by Hollenbeck et al. [62] involves a four-person team assigned the task of monitoring the airspace in the vicinity of an aircraft carrier battle group. The team members, having different areas of expertise, have to collaborate in a timely manner to identify threats from the approaching aircrafts.

The diversity of resources, knowledge, and ideas produces conflict. However, conflict in work teams is not necessarily destructive [63]. Cooperative conflict can contribute to effective problem solving and decision making by motivating people to examine a problem; energizing people to seek a superior solution; and fostering integration of several ideas to create high-quality solutions [64]. The Team Resolution Process [65] is recommended by Rayeski and Bryant to handle conflict when it occurs in human teams. The Five-P's of Conflict Management [66] is identified as patterns common to all controversies. Negotiation is the most effective approach to solving conflict. It is recommended [67] that in negotiation the assistance of a third party can help the two conflicting parties establish a mutual definition and understanding of the problem.

Situation awareness (SA) is a psychological construct that is not directly observable [68]. Endsley defined SA as being composed of three parts: the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [69]. In a distributed environment, situation awareness must be viewed as a collective property (a type of collective consciousness) because of the distribution of awareness and knowledge. Collaborative situation awareness helps create and maintain a high level of shared collective knowledge, which can be leveraged to reconcile differences, construct a common picture, and plan joint activities. For instance, in Network-centric Warfare [15], shared battlespace awareness is deemed to be one of the vital factors in the conduct of military operations in the new information age.

In a word, all the above collaborative behaviors can be covered by the term "collaborative problem solving", with the 'problem' varied.

Helping behaviors can come in many different forms, such as emergency aids, philanthropic acts, etc. Pearce and Amato have developed an empirically-derived taxonomy of helping behaviors [70]. The model has a threefold structure of helping: (1) doing what one can (*direct help*) vs. giving what one has (*indirect help*); (2) spontaneous help (informal) vs. planned help (formal); and (3) serious vs. non-serious help. These three dimensions correspond to the type of help offered, the social setting where help is offered, and the degree of need of the recipient. In a study by Anderson and Williams, helping behavior is examined among supervisor-subordinate dyads from a leader-member exchange (LMX) perspective [71]. As a special kind of helping behavior, backing-up behavior has been defined as helping other team members perform their roles [72]. It is recognized that the central characteristic of backing-up behavior is that it reflects both the recognition and correction of a workload distribution problem [73]. Porter et al. [73] developed a model devoted specifically to understanding backing-up behavior in teams. The model predicts that backing up behaviors are a function of characteristics of the team's task, those team members who can provide the back up, and those team members who receive the back up. There have also been studies suggesting that helping behavior in human teams is enabled by some "overlapping shared mental models" that are developed and maintained by members of the team [74], [75], [76], [77], [73].

IV. Agent-only Teams

In this section we survey some notable agent systems developed for or used in teamwork settings. For each of the system, if applicable, we identify and analyze the underlying shared mental model, communicative behaviors, collaborative behaviors, and helping behaviors.

A. Existing Work

The existing systems modeling and simulating human teamwork using agent technologies can be classified into three categories: team-oriented agent architectures, team-aware cognitive systems, and agent-based teamwork testbeds.

For team-oriented agent architectures, we consider OAA [78], STEAM [23], CAST [24], GRATE* [52], and RETSINA [79]. OAA (Open Agent Architecture) [78] is a facilitator-based architecture. OAA adopts a blackboard-based framework that allows individual agents to communicate by means of goals posted on blackboards controlled by facilitator agents. Basically, when a local facilitator agent determines that none of its subordinate agents can achieve the goal posted on its blackboard, the facilitator propagates the goal to a higher-level blackboard controlled by a higher-level facilitator. This process continues until some competent agent takes the responsibility. Having knowledge regarding the capabilities of a collection of agents (and sub-facilitators), facilitator agents play an important role in (1) establishing connections between service requesters and providers; (2) providing a general notion of transparent delegation, (3) offering a mechanism for organizing an agent society in a hierarchical way.

STEAM (a Shell for TEAMwork) [23] is a hybrid teamwork model built on top of the Soar architecture [40]. STEAM borrows from the strengths of both the Joint Intentions theory [7] and the SharedPlans formalism [50]. It uses joint intentions as a building block to hierarchically build up the mental attitude of individual team members, and to ensure that team members pursue a common solution path. TEAMCORE [80] is an extension of STEAM, taking the perspective of team-oriented programming. TEAMCORE realized a wrapper agent that handles domain-independent team expertise and domain-specific team-level knowledge separately, and hides the details of the coordination behavior, low-level tasking, and re-planning from programmers.

CAST (Collaborative Agents for Simulating Teamwork) [24] supports teamwork using a richer computational shared mental model, which includes adaptive team processes represented by predicate-transition nets (PrT nets) as well as team structures and the capability of each team member. Using such a shared mental model, every CAST agent can reason about what other teammates are doing, what the preconditions of the teammate's actions are, whether the teammate can observe the information required to evaluate a precondition, and hence what information might be potentially useful to the teammate. As such, agents can figure out what information is needed to be proactively delivered to teammates.

GRATE* (Generic Rules and Agent model Testbed Environment) [52] is a general-purpose cooperation framework implemented using the Joint Responsibility model. A rule-based approach is adopted in GRATE*, where the rules (including situation assessment rules and cooperative interaction rules) provide an operational semantics for Joint Responsibility. One key task of applying GRATE* to domain problems is to determine those events which could cause the situations related to responsibility problem to happen and to encode them explicitly into appropriate rules. Such an explicit representation of cooperation enables a group of GRATE* agents to behave both when joint activity is progressing satisfactorily and when it runs into difficulty.

RETSINA-MAS architecture [79] is extended from the RETSINA (Reusable Environment for Task-Structured Intelligent Networked Agents) agent architecture [81]. RETSINA-MAS agents interact with each other via capability-based and team-oriented coordination. Initially all agents share a common partial plan for fulfilling a team task (goal). Each agent then matches its capabilities to the requirements of the overall team goal within the constraints of its authority and other social parameters. This produces a set of candidate roles for the agent, who can select some and communicate to its teammates as proposals for its role in the team plan. Once the team members have reached a consensus that all plan requirements are covered by the role proposals without any conflicts, they can commit to executing the team plan.

COGNET/BATON and Team-Soar are two team-aware cognitive systems able to mimic the way that humans think and solve problems. COGNET/BATON is a system for modeling and simulating human work performance in complex environments, where COGNET (COGnition as a NEtwork of Tasks) [82] refers to a modeling framework for representing the internal body of expertise and BATON (Blackboard Architecture for Task-Oriented Networks) [83] refers to an emulation of the internal information processing mechanisms which process and execute expertise. Recently, COGNET/BATON was extended [84] to support cooperative and teamwork behaviors. *Self-awareness blackboard*, proactive, reactive, and introspective controls are the primitive building blocks for modeling teamwork processes in COGNET/BATON. It is illustrated [84] that various types of cooperation could be simulated with controls that work from the declarative knowledge on the self-awareness blackboard.

Team-Soar [85], built on top of Soar [40], is a computational model implemented specifically for testing a theory of team decision making called “multilevel theory” [86]. It simulates a naval command and control team consisting of four members who have different expertise and need to cooperate on the task of identifying the threat level of incoming aircraft. It is shown [85] that the experiments conducted using Team-Soar supports the same hypotheses of the multilevel theory that the human team experiment supported [86].

In essence, teamwork testbeds are not agent systems by themselves. Rather, they provide a simulated environment for studying teamwork-related issues. RoboSoccer [87] and Robocup-Rescue [14] are two such testbeds.

The RoboCup Soccer Simulator is a multi-agent system testbed where two teams of 11 simulated robotic soccer players can compete against each other. Since 1997, RoboCup international competitions and conferences have been held annually covering simulation league, small-size robot league, middle-size robot league, four-legged robot league, and humanoid league. The ultimate goal of the RoboCup project is to develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer by 2050 [87], [88].

Built upon the success of the RoboCup Soccer project, the intention of the RoboCup-Rescue project is to promote research and development in disaster rescue domains at various levels involving multi-agent teamwork coordination, physical robotic agents for search and rescue, information infrastructures, etc [14]. The project has two camps: the simulation project, and the robotics and infrastructure project. The simulation project provides an integrated simulation of disaster situations including the spread of fire, the collapse of buildings, traffic jams, life line cut, etc. By following common protocols, autonomous virtual entities such as civilians, fire brigades, ambulance teams and police forces can be developed to act in the simulated world. This not only offers a rich source of research for AI planning, learning, and resource sharing, but in the long run it could also provide emergency decision support and actually save people in case of large-scale disasters [89].

B. Shared Mental Models

A shared mental model produces a mutual awareness, which is the key for supporting many interactions within a team that lead to its effectiveness and efficiency [6]. The scope of shared mental models covers shared ontology [90], common knowledge (beliefs) [36], joint goals/ intentions/ responsibility [91], [7], [52], shared team structure [24], common recipes [52], shared plans [50], etc. The notion of shared mental models (SMM) has been put forward to explain certain coordinated behaviors of human teams [77], [74].

There have been several theoretical studies on SMM in the AI field. In particular, the Joint Intentions theory [7] introduces a notion of joint mental attitude (i.e., joint intention) based on the concept of joint persistent goal. The theory requires a team of agents with a joint intention each to not only commit to its part in achieving the shared goal, but also to commit to informing others when the goal has been accomplished, becomes impossible to achieve, or becomes irrelevant. It thus indicates that robust multi-agent systems can be implemented to work in a dynamic environment if agents can monitor joint intentions and rationally react to changes in the environment. Along this line of research, Jennings [52] proposed the Joint Responsibility model with the explicit representation of cooperation using common recipes. The Joint Responsibility model, which refines Cohen and Levesque's work on joint intentions [7], explicitly captures different causes of recipe failures and clearly specifies the conditions under which an agent involved in a team activity should reconsider both its joint goal commitments and common recipe commitments. In addition, a series of research has been focused on the formal reasoning of common knowledge [36].

The notion of shared mental models in an implemented system may be implicit. For instance, to be able to communicate directly, agents in any MASs have to agree on a shared ontology, which is often implicitly embedded in the system. It is also implicitly assumed that agents in a MAS have common knowledge on the domain tasks to be solved, the communication protocols to be used, the social laws or social normatives to follow, etc.

TABLE I

SHARED MENTAL MODELS OF THE SYSTEMS DESCRIBED IN SECTION IV-A

Types	Systems	Components of SMM
Blackboard-based	OAA	hierarchically organized facilitators shared by agent community, Triggers
	COGNET/BATON	team goal, meta-model of team process and teammates
Joint Mental Attitudes	GRATE*	Joint Responsibility, Common recipe
	STEAM	Hierarchy of joint intentions, team state
	RETSINA-MAS	joint intentions, shared plans
	CAST	shared plans, shared team structure & progress
Rule-based	Team-Soar	SOAR rules for interactions
	RoboCup Soccer RoboCup Rescue	common communication protocols, common constraints

Table I summarizes the SMM implemented implicitly or explicitly in the systems described in Section IV-A, where the systems belong to the categories of blackboard-based and rule-based have only implicitly embedded common knowledge (i.e., structures, rules), and the systems belong to the category of joint mental attitudes have a stronger notion of computational SMM.

Both OAA [78] and COGNET/BATON [84] use a blackboard structure. OAA triggers provide a general mechanism for requesting that some action be taken when some set of conditions is met. The installation of a trigger within an agent can be viewed as that agent's commitment to evaluating the conditional part of the trigger [78]. Hence, an agent can share commitments with other agents by installing the same triggers. OAA blackboards controlled by facilitator agents provide a global data store for its client agents. A request (e.g., information need) posted on a blackboard implicitly establishes a shared goal for those agents who can contribute to the request. For instance, an agent can ask to be notified of certain changes in certain predicates by placing a trigger on the blackboard [92]. In contrast, the blackboards in COGNET/BATON are not shared; each agent has its own blackboard as a structure for realizing self-awareness. In its blackboard, an agent not only records its own affair, but also models the states of other team members and the relationship of its local activities to the larger goals and team process. This establishes a cooperative awareness of team affairs. Such self-awareness establishes an inherent need for collaboration between individuals and tasks [84]. Thus, for a group of COGNET agents to achieve effective teamwork, the contents of their blackboards should overlap.

Agents in rule-based systems typically share the rules and knowledge governing their cooperative behaviors. All the members in a Team-Soar team share knowledge such as the interrelations between environmental attributes, the accessibility of attributes, and other members' expertise (the ability to evaluate attributes, combination rules) [85]. Agents developed for acting in the RoboCup Soccer or RoboCup Rescue simulators also need to conform to common communication protocols and constraints (e.g., all agents must decide an action within half a second).

GRATE*, STEAM, RETSINA, and CAST are systems with explicitly shared mental models. GRATE* is a computational interpretation of the Joint Responsibility model, which distinguishes two types of joint commitment: joint goal commitment and joint recipe commitment. The semantics of joint commitment are encoded as generic cooperation rules, which are shared by all team members. GRATE* agents use common recipes, developed via the distributed planning protocol [52], to guide team activities. In addition, knowledge of others' capabilities maintained in each agent's acquaintance model could also be overlapped.

STEAM agents can build up snapshots of the team's mental state through forming joint intentions at different levels of abstraction [23]. The intention hierarchy established jointly by team members through the establish-commitments protocol ensures that the team pursues a common solution path in execution. The SMM in STEAM also includes the prescription of team operators (e.g., decomposition), team state of active team operators (e.g., team members in the team, subteams, pre-determined team leader, etc.), and the explicit declaration of information-dependency relationships among actions.

Both RETSINA-MAS and CAST borrow from the strength of the SharedPlans formalism [54], [50]. A shared plan is characterized in a mental-state view as a particular collection of beliefs and intentions. An agent is said to have a shared plan with others if and only if the agent works towards establishing and maintaining those required mental attitudes, and it believes the other agents do so likewise [54]. Each shared plan is associated with a shared recipe, of which team members may have different partial views. The SMM of a team of RETSINA-MAS agents includes team plans—high-level description of tasks, and state of team affairs (assignment of team roles, authority relations, team goals).

CAST [24] has implemented a richer shared mental model compared with the systems mentioned above. The SMM includes joint goals, organizational structure (i.e., team membership, sub-team relations, roles each member can play, capability requirements on each role), information-needs graphs, the structure of team plans, as well as team members' capabilities and observability. In addition, the monitoring of teamwork progress is also very important for agents to gain awareness of the current context. Thus, the shared mental model of CAST also contain the dynamic status of team process as built-in information needs. Treated as such, an CAST agent can not only better anticipate others' relevant information needs and reduce the cost of communicating irrelevant information, it can also dynamically adapt team structure and team process to external changes.

C. Collaborative Behaviors

In Section III, we mentioned that collaboration is needed when a human team pursues a joint goal but the achievement of the goal is beyond the capability, knowledge, or capacity of any individual. Examples of collaboration include collaborative learning, collaborative planning, collaborative conflict handling, and collaborative situation awareness. From the literature, we can identify the collaboration behaviors that have been or could be simulated in the systems described in Section IV-A. However, not much insight can be gained from such identification alone because collaboration behaviors are typically domain-dependent. Rather, we analyze the capability of the systems for collaboration behaviors from three abstract levels: atomic team actions, coordinated tasks and planned team activities.

Atomic team actions refer to those atomic actions that cannot be done by a single agent and must involve at least two agents. Typically, a team action is associated with certain constraints describing the prerequisite conditions for doing the action. Examples of such constraints include the number of performers required, constraints for choosing performers, quality of performance, etc. Before doing a team action, the associated preconditions should be satisfied by all the involving agents, and the agents should synchronize when performing the action. Atomic team actions can be found everywhere in the physical world. For example, two people shake hands with each other; several robots lift a heavy object. However, most of the simulation systems lack the support for atomic team actions, partly because collaboration at this level is hard to model because it requires every performer to continuously monitor its partners' performance at a small granularity and respond to their subtle changes in a timely fashion, and partly because modeling collaboration at a higher level can more easily expose issues attractive to a wider research community. Among the above-mentioned systems, only CAST provides a full support for atomic team actions. CAST has a language-level construct called TeamOperator, with which agent developers can specify team actions identified from the domain under concern. Preconditions and constraints on the number of performers can also be specified for a TeamOperator. A TeamOperator, called *co_fire* [93], is used in an experiment studying the teamwork behaviors supported by CAST. Although it is still unclear whether there is a generic support for team actions in RoboCup Soccer and RoboCup Rescue, concrete examples do exist in these two real-world simulators (e.g., ball-passing, collaboratively moving a heavy object).

TABLE II

COLLABORATION CAPABILITIES OF THE SYSTEMS DESCRIBED IN SECTION IV-A

Systems	Atomic team action	Coordinated task	Planned team activity
OAA		Triggers with compound goals	
COGNET/BATON			team process in goal-trees
GRATE*			common recipes
STEAM		AND/OR-combination, role dependency	team operators
RETSINA-MAS			team plans in HTN
CAST	TeamOperator	Joint-DO	MALLET process
Team-Soar			hard-coded team decision making process
RoboCup Soccer	e.g. ball-passing	e.g. arrange a goal shot	
RoboCup Rescue	e.g., move heavy objects	e.g. several agents pass a narrow road	depend on the client agents

By coordinated tasks we refer to those short-term (compared with long-term plans) activities involving multiple agents. Executing a coordinated task often requires the involved agents to establish joint and individual commitments to the task or sub-tasks, to monitor the execution of the task, to broadcast task failures or task irrelevance whenever they occur, and to replan doing the task if necessary. A coordinated task is typically composed of a collection of temporally or functionally related sub-tasks, the assigned doers of which have to synchronize their activities at the right time and be ready to backup others proactively. An OAA Trigger with a compound goal posted on a blackboard launches a coordinated task for all the subordinate agents, who need to work together in their pursuit of the compound goal. STEAM uses role-constraints to specify the relationship between sub-tasks of a coordinated task [23]. AND-combination is used when the success of the task as a whole depends on the success of all the sub-tasks; OR-combination is used when any one sub-task can bring success to the whole task; and role-dependency can be used when the execution of one sub-task depends on another. Complex joint team activities can be specified by using these role-constraints combinatively and hierarchically. CAST uses the notion of joint types to specify execution constraints on a coordinated task. The Joint-Do construct has three joint types: AND, OR, and XOR [24]. A Joint-DO statement with joint type AND succeeds if and only if all of the sub-tasks are executed successfully. A Joint-DO statement with joint type XOR succeeds if and only if exactly one sub-task succeeds. A Joint-DO statement with joint type OR succeeds if at least one sub-task succeeds. It is required that all the involved team members synchronize before and after executing a Joint-Do statement. Arranging a goal shot in RoboCup Soccer and several platoon agents passing a narrow road in RoboCup Rescue are concrete examples of coordinated tasks.

Planned team activities refers to common recipes that govern the collaboration process of a team in solving complex problems. A planned team activity is a long-term process that often involves team formation, points of synchronization, task allocation, execution constraints and temporal ordering of embedded sub-activities. COGNET/BATON uses goal-trees to represent team process, where each goal is associated with triggering knowledge and priority knowledge. GRATE* has a recipe language, which allows trigger-conditions, roles to be filled, and structure of sub-operations to be specified for a recipe. STEAM [23] uses the notion of team operator (this notion is different from the notion of TeamOperator in CAST, where a TeamOperator defines an atomic team action) to prescribe the decomposition of task structures. Shared team plans in RETSINA-MAS are represented in the form of an HTN planning structure. Team-Soar only has one hard-coded team process for making decisions. Comparatively, CAST has a team-process encoding language called MALLET. A MALLET plan specifies which agents or agent variables (agent selection can be done dynamically based on the evaluation of certain teamwork constraints) under what pre-conditions can achieve what effects by following which process, and, optionally, under what conditions the execution of the plan can be terminated. A complex MALLET process can be composed hierarchically using constructs such as sequential (SEQ), parallel (PAR), iterative (WHILE, FOREACH, FORALL), conditional (IF), and choice (CHOICE). The CHOICE construct, composed of a list of alternatives (each of which invokes a plan and is associated with preference conditions and a priority information), can be leveraged to support adjustable team processes. CAST agents can respond to execution failures by backtracking to the nearest choice point and making a better decision on the next course of actions based on the current situation.

D. Communicative Behaviors

Inter-agent communication is rooted in research on speech act theory, which can be traced to work by Searle [94] and further to the seminal work by Austin [95]. In early 1990, Cohen and Levesque proposed the idea of "performative-as-attempt" [96] which has been adopted as the standard way of assigning mentalistic semantics to communicative acts (e.g., Arcol [97], KQML [98], and FIPA's ACL [99]). Maintaining group awareness and achieving effective coordination depend on communication. For instance, communication plays an essential role in dynamic team formation, in implementing team-oriented agent architectures, and more theoretically, in forming, evolving, and terminating both joint intentions [7] and shared plans [50].

Implemented systems often apply the Joint Intentions theory [7] in deriving inter-agent communications. The Joint Intentions theory requires that all the agents involved in a joint persistent goal (JPG) take it as an obligation to inform other agents regarding the achievement or impossibility of the goal. The Joint Responsibility model [52] extends the Joint Intentions theory by clearly specifying the conditions under which an agent involved in a team activity should reconsider its commitments. Specifically, in addition to communications for dropping a joint intention, an agent should also endeavor to inform all the other team members whenever the desired outcome of the common recipe under concern is already available or the recipe becomes invalid, untenable, or violated.

Both reactive and proactive communication embodied in human discourse exist in agent systems simulating human teamwork. Because communication takes time and attention, resource-bounded agents often employ selective communication. We compare the communication modes implemented in the systems described in Section IV-A in terms of whether they allow reactive, proactive and selective communication and how they derive inter-agent communication (see Table

III).

As the basic communication mode, reactive communication (i.e., ask/reply) is supported in all the above-mentioned systems. In essence, the motivation of ‘ask’ depends on the speaker’s self-awareness of its needs while the replier’s response is triggered by the external request. Among the systems, OAA, COGNET/BATON and RoboCup Soccer Simulator employ a blackboard style of communication. In OAA, all communication between client agents must pass through blackboards; goals posted on a publicly-accessible blackboard establish service requests to all agents providing the relevant services. COGNET agents, via self-reflection, can actively remind (request) the other agents to provide information [84]. All players in RoboCup Soccer Simulator communicate by posting to a common blackboard (a single, unreliable communication channel): each agent monitors all messages from teammates to determine their internal states, even if the content of the message is intended for another teammate [100]. Communication in RoboCup Soccer is driven by collaboration needs. For example, a player can send a message via the blackboard to another player if the first player wants to pass ball to the second but does not know its location. RoboCup Rescue simulator has a local language defined for the agent-agent reactive communication (e.g., say, tell, listen) [14]. Communication in Team-Soar [85] is via hard-coded communication channel and driven by rules. Reactive communication is used in GRATE* when an organizer agent needs to establish a team and devise a common recipe for performing a joint activity.

TABLE III

COMMUNICATION STYLES OF THE SYSTEMS DESCRIBED IN SECTION IV-A

Systems	Reactive	Proactive	Selective	Driven by
OAA	yes			goals posted on public blackboard
COGNET/BATON	yes	yes		needs on self-awareness blackboard
GRATE*	yes	yes		joint responsibility, rules
STEAM	yes	yes	decision analysis	joint intentions, rules
RETSINA-MAS	yes	yes		joint intentions
CAST	yes	pro-tell	decision analysis	anticipated information needs
Team-Soar	hard-coded communication channel			rules
RoboCup Soccer	yes		e.g., locker-room agreements [100]	collaboration needs
RoboCup Rescue	yes		limited communication	situation

However, it is inadequate for agents in complex situations to only engage in reactive communication. First, an information consumer may not realize certain information it has is already out of date. If this agent needs to verify every piece of information before it is used, the system can be easily overwhelmed by the amount of communications entailed by such verification messages. Second, there are cases where an agent itself may not realize it needs certain information due to its limited knowledge (e.g., distributed expertise). For instance, a piece of information may be obtained only through a chain of inferences (e.g., being fused according to certain domain-related rules). If the agent does not have all the knowledge needed to make such a chain of inferences, it will not be able to know it needs the information, not to mention requesting for it. Proactive information delivery offers a better solution to the limitations exposed in reactive mode; it shifts the burden of updating information from the information consumer to the information provider, who has direct knowledge about the changes of information. Proactive information delivery also allows teammates to assist an agent who cannot realize it needs certain information due to its limited knowledge.

Among the systems, COGNET/BATON, GRATE*, STEAM, RETSINA-MAS and CAST support proactive communication, which is driven either by anticipated needs, or pre-determined rules, or the semantics of the underpinning teamwork model. In COGNET/BATON, proactive information exchange is modeled through two metacognitive processes [84]. As soon as an agent, A, realizes that a teammate, B, is beginning a task that might require information from A, it will post such a contingency on the self-awareness blackboard. Then, agent A will seek information periodically on how close B is to needing input, and send B the information input proactively as soon as it is available. Proactive communication in GRATE* is driven by the situation assessment rules, which encode the operational semantics of the Joint Responsibility model. The rules ensure that all team members are informed if a local agent reneges upon its joint commitment [52]. Proactive communication in STEAM [23] is driven by commitments embodied in the Joint Intentions theory, as well as explicit declaration of information-dependency relationships among actions. For example, a STEAM agent has to inform other team members before it terminates a team operator. RETSINA-MAS agents also derive proactive communication from joint intentions [79]. To form a shared mental model about each others' commitments to a team plan, every individual needs to inform others of its candidate roles. If an individual discovers that it is no longer capable of performing a role, the agent must communicate this knowledge to its teammates and superiors. The proactive information delivery behavior implemented in CAST is enabled by the anticipation of team members' information needs through reasoning about the current progress of the shared team process [24]. In particular, the communication of teamwork progress (control token information) is implemented in CAST as a kind of built-in information-needs. CAST agents need to inform appropriate teammates when they start a team plan, a team operator, a joint-action, or when the execution of a plan is terminated.

The strong requirement on communication among teammates entailed by the Joint Intentions theory is necessary to model coherent teamwork, but it is too strong in the real case (e.g., in time-stress domains) to achieve effective teamwork. Many practical strategies have been sought to achieve selective communication. For instance, to deliberate upon communication necessities, STEAM agents consider communication costs, benefits, and the likelihood that some relevant information may be already mutually believed. CAST employs a context-centric approach to anticipate others' information needs. For example, a team process may include choice (decision) points, each of which can specify several branches (potential ways) to achieve the goal associated with the choice point. CAST agents would not activate the information-needs emerging from a specific branch of a choice point until that branch is selected [101]. Similar to STEAM, CAST agents also use a decision theoretic approach to evaluating the benefits and costs of communication when they consider satisfying others' information needs. RoboCup Soccer Simulator provides a realistic communication environment; both communication range and communication capacity of client agents are limited (e.g., an agent can hear at most 1 message every 200 msec). Pre-determined protocols such as Locker-room agreements [100] have been designed to facilitate effective teamwork in RoboCup Soccer where communication is heavily restricted.

E. Helping Behaviors

Helping behavior in MultiAgent Systems can be defined as helping other team members perform their roles under conditions of poor workload distributions, asymmetric resource allocations, or encountering unexpected obstacles. Theoretically, helping behavior can be derived within both the Joint Intentions theory [7] and the SharedPlans formalism [50], [51]. For example, a team holding a joint commitment implies that any agent in the team will intend to help if it knows a team member requires its assistance [7]. The help axioms in the SharedPlans theory state that an agent will consider doing an action if the performance will contribute to the fulfillment of the agent's intention from which its teammates can benefit. Yen, Fan and Volz have proposed an axiom to characterize chains of helping (e.g., indirect help) that often occur in large agent teams [102].

As an alternative to Pearce and Amato's taxonomy [70], helping behaviors can simply be classified as *reactive* helping and *proactive* helping. Reactive helping is triggered by an external request and it can be easily realized by agents with the capability of multi-tasking. For instance, in RoboCup Rescue, when a fire is getting out-of-control the working team can ask another fire-fighting team for help. Helping behaviors are often domain-dependent and there is no doubt that all the 9 systems can or can be easily extended to model reactive helping behaviors in various domains (e.g., in the mission scenario simulated by RETSINA-MAS [79], one platoon requests reinforcement from other team members).

Some of the systems also support proactive helping. Proactive helping is not initiated by help requests but by the anticipation of others' needs from certain shared mental models—even if such needs are not directly expressed [103]. To help others often requires the agent to monitor others' performance. For instance, a COGNET/BATON agent determines whether its teammates need help by launching a proactive control to periodically assess their performance. The agent will offer guidance or help in a proactive manner once a team member is judged to be in trouble [84]. One notable helping behavior in STEAM is the way of dealing with teamwork failures. STEAM uses role-monitoring constraints (AND, OR, dependency) to specify the relationship of a team operator and individual's (or subteam's) contributions to it. When an agent is unable to complete actions in its role (via tracking teammates' role performance) and the embedding team operator is still achievable, the remaining agents will invoke a repair plan accordingly.

Broadly speaking, proactive information exchange featured by COGNET/BATON, GRATE*, STEAM, RETSINA-MAS and CAST can also be taken as helping behavior (e.g., helping others gain global situation awareness, enhance coordination, achieve coherent and effective teamwork).

V. MIXED HUMAN/AGENTS TEAMS

Similar to Section IV, we examine some notable mixed human/agent systems in the literature. For each of the systems, when applicable, we identify and analyze the underlying shared mental models, communicative behaviors, collaborative behaviors, and helping behaviors.

A. Existing Work

Intelligent agents are better at numerical or symbolic processing (e.g., information fusion, data mining), while humans are experts in cognitive computing such as exception/emergency handling, multiple meta-level reflections, and comprehensive situation assessment. It is claimed that agents, if they could collaborate with human peers effectively, can allow humans to pay their limited attention to more important activities [29], can help build and maintain situational awareness without overloading human teammates [104], and can make better decisions using information at a greater accuracy and finer granularity [8].

Recent research has been highlighting the trend of supporting various dimensions of human-agent interactions [105]. In this section we focus our investigation on three developing fields: mixed-initiative user interfaces, human-centered teamwork and human-guided teamwork. In particular, for mixed-initiative user interfaces we consider the efforts at Microsoft [5], [106], [17] and MERL [53], [107], [108]; for human-centered teamwork we examine KAoS+Brahms [29], [109] and MokSAF [8], [110]; for human-guided teamwork we consider proxy-based coordination [111] and the ongoing R-CAST project [112].

With the long term goal of allowing people to interact with computing systems in a manner similar to the way that people interact with one another, the Attentional User Interface (AUI) project at Microsoft has been seeking to build computing systems that can recognize the attentional cues from human users and use such cues to guide the timing of interactions with the users. Horvitz et al. employed Dynamic Bayesian networks for inferring attention from multiple streams of information and used Hidden Markov Models for reasoning about users' states of attention under uncertainty [5]. The attentional models have been applied in developing several interface agents. For instance, Bayesian Receptionist [106] is a system that models dialogs in a joint-activity situation; it uses Bayesian models for inferring a user's communication goal and uses decision-theoretic control policies for determining when to progress to the next level of analysis.

Collagen (Collaborative Agent) [53] is a software system built on top of the same principles that underlie human collaboration from research in discourse theory [54], [50]. It realizes a standard mechanism for maintaining the flow and coherence of interactions between a human user and an intelligent agent. The Collagen architecture provides data structures and algorithms for representing and manipulating goals, actions, recipes, and SharedPlans. Collagen has been used in developing interface agents for air travel planning [53], simulation-based training [108], and thermostat controlling [107].

Human-centered teamwork takes the perspective that humans are the crucial elements in the system while agents should be adapted to serve humans' needs [29]. It thus requires agents to have a detailed understanding of how people actually work.

KAoS is a collection of agent services in the form of adjustable policies, which assure that agents will always operate within the boundary of established behavioral constraints and will be responsive to human control [109]. Brahms is a multi-agent modeling and simulation environment for understanding the interactions between people and systems [21], [113]. Brahms allows human behaviors to be represented and studied from the perspective of activity theory and work "practice", thus helping to uncover the roles people actually play in a complex work space. The integration of Brahms and KAoS allows Brahms agents, guided by the teamwork policies explicitly incorporated in KAoS, to renegotiate roles and tasks among humans and agents when new opportunities arise or when breakdowns occur [29]. This idea of policy-based adjustable autonomy is under testing in a NASA funded project, where human-robot teamwork onboard the International Space Station is modeled using Brahms+KAoS [29], [109]. The model includes a detailed conceptual representation of onboard life, executions of daily planned tasks, crew's activities, etc. The model helped uncover some unexpected opportunities for assistance from robots [29].

MokSAF (a simplified version of ModSAF, which is the acronym of Modular Semi-Automated Forces) is a computer-based simulation system developed to evaluate how humans can interact and obtain assistance from agents within a team environment. Each commander has a route-planning agent (RPA) and an interface agent (MokSAF) with an embedded Geographic Information System (GIS). A MokSAF agent can monitor a human's progress, and reallocate or modify tasks if the human fails to perform a task [28]. In experimental studies [8], [110], the task consists of three military commanders each assembling a platoon and planning routes so that all three platoons arrive at a given rendezvous by a specified time without violating any physical or intangible constraints. The commanders must evaluate their plans from a team perspective and iteratively modify the plans until an acceptable team solution is developed.

In human-guided teamwork, humans' strengths are largely leveraged in guiding agents to cooperate towards a desirable solution path. In a study by Scerri et al [111], the idea of proxy-based coordination (i.e., agents coordinate with others through their proxies) was used to create heterogeneous teams for an urban disaster recovery domain. The proxies, based on a quantitative model of teammates' capabilities, allow a team to allocate and reallocate roles (including coordination tasks) to the most willing and capable team members. To explore how different styles of coordination among team members may affect team performance, an experiment was conducted using an extended RoboCup Rescue simulator, where agents work in the virtual city while people and robots act in the physical world. The preliminary experiment involves a human fire chief and a number of agent-controlled fire brigades. The human fire chief is given a high-level view of the fire-fighting progress, the chief thus can make better role-allocation decisions to save trapped and injured civilians and limit damage. The results did confirm the hypothesis that exploiting human assistance in coordination can improve the overall team performance.

R-CAST [112] is a multi-agent system that supports human decision-making in a team context by using a collaborative RPD process (Recognition-Primed Decision model, see [114]) as part of the team members' shared mental model. Fuzzy clustering and case-based reasoning are employed to learn coordination autonomy rules regarding under what contexts the human user is interruptible and under what situation agents can take the authority to make decisions for the human. Humans' assistance in the collaborative RPD process may occur in three situations. First, R-CAST agents support decision making in time-stress domains. Human users can dynamically adjust the time interval for making a decision based on the soft real-time constraints. Second, a human user, based on his expertise, can input assumptions to explore potential alternatives, and suggest new information-fusion rules to link pieces of information. Third, human teammates can handle exceptions emerging from the decision-making process by seeking help from other agents or human team members.

B. Shared Mental Models

The interactions within mixed human/agent teams come in one of three modes: agent-agent interactions, human-human interactions, and agent-human interactions. Accordingly, the forms of shared mental models governing the joint activities of a mixed human/agent team also vary with the characteristics of the team members. From Section IV-B we know that the shared mental models for agent-agent teams may be blackboard-based, rule-based, or based on joint mental attitudes. Such representations still apply to agent-agent interactions in mixed human/agent teams. How shared mental models among human members in a team are established is far beyond our current understanding. However, this has nothing to do with the ability of humans to form high-performing teams. Therefore, one remaining but more important issue faced by mixed human/agent teams is how a human and an agent represent and reason about their mutual awareness. This, in turn, will depend on breakthroughs in a variety of fields including Artificial Intelligence, speech recognition, and natural language understanding. As a short-term solution, it is not surprising to note that in the six projects described in Section V-A interface agent technology is unanimously employed to represent the shared understanding between agents and their human users. Table IV summarizes the shared mental models identified for these systems (in the table we omit the specific shared goals of the systems because all the teams implicitly take joint tasks as their shared goals).

TABLE IV

EXAMPLE SHARED MENTAL MODELS OF THE SYSTEMS DESCRIBED IN SECTION V-A

Types	Systems	Examples of SMM
Mixed-initiative	AUIs: attention modeling	signals about attention, predefined costs of disruptions, probabilistic user model of attention, task abstraction hierarchy [106]
user interfaces	Collagen: discourse theory	mutual beliefs (recipe trees, history list), SharedPlans, discourse state
Human-centered	Brahms+KAoS	humans' daily behavioral patterns, working context, teamwork policies
Teamwork	MokSAF	shared domain knowledge (e.g., terrain map, intangible constraints)
Human-guided	Proxy-based coordination	team plans, team state, roles (shared responsibility)
Teamwork	R-CAST	team process and progress, interruption policy

The AUI project at Microsoft has developed several mixed-initiative interface agents that can share with users natural signals about attention [17]. For instance, the LookOut system [106] maintains a pretrained probabilistic user model of attention that can be incrementally refined through continual learning. The user model contains assertions about the cost of disruption for each alert modality (i.e., each alert may have different visual displays), conditioned on the user being in different attentional states. To enhance the value of services and alerts, LookOut also considers the temporal pattern of a user's attentional focus. The Bayesian Receptionist system [106] can construct a task abstraction hierarchy to facilitate the understanding of a user's goals.

The task abstraction hierarchy guides the conversation between the agent system and the user toward shared understanding about uncertainties and misunderstandings at progressively greater detail. It thus leads to a more tractable, level-specific modeling and inference of user's intention.

The elements that are shared between Collagen agent [53] and its human user include mutual beliefs such as recipe trees, SharedPlans, and focus of discourse attention (also called discourse state [107] for tracking shifts in the task and conversational context). Collagen agents also maintain a memory of recent interactions with users. Such "shared" short-term experienc allows users to make efficient and natural references to objects and services in the most relevant contexts. Collagen applies focus-guided plan recognition to the utterances and actions in order to maintain a finer shared mental model between user and the agent [107].

The shared mental models among humans and robots in the human-centered teamwork project [29], [109] contain two parts: humans' daily behavioral patterns are learned, modeled using Brahms, and shared at the group level while teamwork policies are encoded using KAoS. In the empirical studies using MokSAF [8], [110], what is shared between commanders and their route-planning agents is domain knowledge such as terrain map and intangible constraints which are represented through MokSAF interface agents.

The shared mental model of the human-agent team employing proxy-based coordination [111] includes team plans and various roles derived from the team plans (i.e., shared responsibility). It also includes a memory of team state, which allows a proxy to quantitatively estimate team members' dynamic capabilities in role allocation and reallocation. R-CAST agents [112] share with their human team members the decision-making process and the dynamic progress (for anticipating relevant information needs), as well as interruption policies (for achieving reflexive adjustable autonomy).

C. Collaborative Behaviors

The collaborative behaviors of the systems described in Section V-A can be examined according to the taxonomy given in Section IV-C. Although there is no salient evidence that these systems support atomic team actions that have to be performed jointly by humans and agents, examples of coordinated tasks and planned team activities can be easily identified. For instance, route-planning is a coordinated task between the *autonomous* RPA (route-planning agent) and its human user; the RPD process shared among all the members of a R-CAST team is a planned team activity.

We examine how collaborations are implemented in the systems. The mixed-initiative collaboration implemented in the LookOut system [5] has three modes: manual mode, basic automated-assistance mode, and social-agent mode. In manual modality, collaboration is invoked directly by the user; in basic automated-assistance mode, the agent can request additional information from user considering the cost of disruptions; in social-agent modality, the agent can engage the user in a dialog using an automated speech-recognition system and a text-to-speech system. By maintaining the conversation history and focus of discourse segments, a Collagen agent can collaborate with its human user to solve domain problems such as planning a travel itinerary [53]. Collaborations in the teams modeled by Brahms+KAoS are governed by teamwork policies encoded in KAoS [29]. The boundary between the initiative of humans and that of agents can be refined through adjusting the policies. This allows agents to collaborate with humans at a desired level of autonomy.

The MokSAF project [110] involves three route-planning agents (RPA), which reflect three levels of human-agent collaboration. *Naive* RPA can analyze the route drawn by a commander by checking the violation of known physical constraints and suggest ways to improve the route. *Cooperative* RPA is self-initiative: it can modify the route to avoid violating the physical constraints. *Autonomous* RPA takes more responsibility; it can guide a commander throughout the route-planning task, using its knowledge of both the physical constraints and intangible constraints and seeking to find the shortest path. Through their proxies, agents in the human-agent disaster-responding team [111] can do role allocation and reallocation collaboratively. In the RPD decision making process, humans and R-CAST agents [112] can collaborate in establishing shared situation awareness, in monitoring an expectancy, etc.

D. Communicative Behaviors

It is much more challenging to maintain mutual awareness in mixed human-agent systems than in agent-only systems. An agent should not only allow its human teammate to understand what has happened and why, but it also should allow human to adjust its autonomy and help people predict what will happen [29], [115]. Thus, effective solutions to supporting human-agent communications are highly needed in mixed human-agent teamwork.

All the systems described in Section V-A employ interface agent technology. Collagen agents are built on top of human discourse theory. MokSAF, RAP Proxy, and R-CAST all feature an interface containing a domain map for facilitating situation awareness. In addition to interfaces, Brahms+KAoS and AUIs (attentional user interfaces) also employ speech recognition and natural language understanding techniques to facilitate direct interactions between humans and agents. For instance, in the mobile agent project [109], people can interact with their personal agents using a speech dialog system integrated with Brahms. However, robust solutions to the speech target problem will continue to be a challenge in allowing users to effectively interact with multiple devices and people [17].

Coming back to communication styles, all the systems support reactive communication. For example, being requested of a view of the current interaction history, a Collagen agent can directly display the content to its user [53]. It can also ask for clarification under ambiguity [107]. If the LookOut system is uncertain (i.e., at lower levels of confidence) about a user's intentions, it can engage in an efficient dialog with its user to get additional information [5]. In the disaster-responding team using proxies [111], an agent can ask the human fire chief to make role-allocation decisions through their proxies.

Based on the reported work proactive communication can only be identified from R-CAST and AUIs. For instance, being informed of the recognition result, R-CAST agents can help the decision-maker in detecting changes to the environmental variables relevant to the expectancies, and inform the changes proactively [112]. If the LookOut system is confident enough in its assessment of a user's goal, it will proactively display a schedule to the user [5].

TABLE V

COMMUNICATION STYLES (FROM AGENTS TO HUMANS) OF THE SYSTEMS IN SECTION V-A

Techniques	Systems	Reactive	Proactive	Selective
Based on User Interface	Collagen	yes		communication menu, plan recognition
	MokSAF	yes		
	RAP Proxy	yes		adjustable parameter: maxAsked
	R-CAST	yes	yes	user's cognition load analysis
Interface + speech recog./ natural lang. understanding	Brahms+KAoS	yes		notification policy
	AUIs	yes	yes	filtered by attention model and decision model

In human-agent interactions it is critical to ensure interruptions of humans by an agent be done judiciously and infrequently [29], [5]. For this reason, most of the above-mentioned systems have sought to achieve selective communications.

The AUI project uses an economic principle in attention-sensitive notification [17]. A decision-analytic model is employed to decide if, when, and how to alert a user about information from multiple message sources, considering the tradeoff between the expected value of information and the costs of potentially bothering a user needlessly. Bayesian Receptioist uses a Bayesian model to infer the probabilities of a user's communication goals at different levels of the goal hierarchy, and then employs value of information (VOI) analysis to identify the best questions to ask and the best visual observations to make [106].

To get around the complexity of natural language understanding, users are not allowed to make arbitrary communications with Collagen agents. Rather, users can only communicate with a Collagen agent by selecting from a dynamically-changing user communication menu, which is computed from the current discourse agenda [53]. Plan recognition technique (i.e., inferring intentions from actions) is also adopted in order to reduce the amount of communication required to maintain a mutual understanding of the shared plans between user and agent [107].

Brahms+KAoS supports selective communication through a so-called notification policy [29]. When an important event is signaled, the utility of various alternatives is evaluated in light of the current task and other contextual factors. If a notification is required, the Brahms agent will notify the human in a manner appropriate with respect to modality, urgency, and location of the human.

The disaster-responding team using proxy-based coordination strategy employs a decision-theoretic approach to communications [111]. In addition, *maxAsked*—an adjustable parameter in the role-allocation algorithm—can be used to control how many of the capable teammates to ask before the team gives up.

R-CAST also employs a decision-theoretic approach for deciding whether to inform others when obtaining a new piece of information. Additionally, the factor of human members' cognitive capacity is considered in making decisions on whether to send a human teammate multiple lower-level information or just send fused higher-level information [104], [112].

E. Helping Behaviors

Normally, humans can recognize the help needs of agents through observation or explicit requests from the agents while agents can approximately anticipate the help needs of humans based on certain probabilistic models and relevant contextual cues. One principle for guiding the design of mixed-initiative user interfaces, as suggested by Horvitz [5], is that it is preferable to “do less” but do it correctly under uncertainty. “Doing less” provides users with a valuable advance towards a solution and minimizes the need for costly undoing or backtracking [5]. This principle also applies to helping behaviors in general: although agents can watch “over the shoulders” of humans and jump in with advice or assistance, the agents have to be cautious to avoid helping humans needlessly. Thus, effective human-agent teamwork operating in a dynamic environment depends largely on how well agents can recognize the opportunities for providing help.

To the best of our knowledge, most of the systems described in Section V-A lack an explicit model of helping behaviors and lack a generic approach to recognizing assistance opportunities. We thus choose to simply identify some examples of helping from the systems.

The probabilistic user model of attention in the AUI project can be leveraged to estimate the opportunities of helping. To avoid poor guessing about the goals and needs of users, the LookOut system resorts to life-long learning to incrementally refine the user’s attention model and timing model. At lower-levels of confidence, LookOut can choose to display its guess to the user for further suggestions [5].

Upon request, a Collagen agent can show its user parts of the agent’s internal discourse state in the form of a *segmented interaction history*—a compact, textual representation of the past and present states of the discourse. The interaction history can not only help the user reestablish where in the problem solving process he/she was, it can also serve as a menu for history-based collaboration control—replaying earlier segments in a new context, allowing the user to return to earlier points in the conversation [53]. A concrete example of help exhibited by the air-travel planning agent is: the agent can propose adding new airlines when it realizes that the user has over-constrained the trip being scheduled [53].

In the mobile agent project [109], Brahms agents are deployed on different mobile systems over a wide-area wireless network. KAoS helps Brahms agents recover gracefully when communication among agents fails. The RPA agents in the MokSAF project can help a commander recover from errors in the route drawn by the commander [110]. In proxy-based coordination [111], the proxy of a human team member can decide whether to reject or accept an unassigned role on behalf of the human. Such an adjustable autonomy can help balance the human's workload, filtering out tasks that would be better performed by other agent team members. At the stage of action evaluation, R-CAST agents can monitor the expectancies recognized by the decision-making agent (and the human supervisor). This may help the whole team recover from inappropriate decisions at the earliest possible opportunity [112].

VI. Recent trends and future directions

In this article we evaluated the major works in the field of teamwork modeling, especially focusing on the modeling of human teamwork behaviors using BDI type systems. More recently, there has been significant interest in applying decision-theoretic principles to build team systems. For instance, work on decentralized Markov Decision Process (DEC-MDP) and partially observable Markov decision process (POMDP) [116], [117], [118], [119] has gained significant attention in recent agents conferences such as AAMAS. The major goal of these studies is to derive action or/and communication policies that can optimize some global value functions. A recent complexity result has shown that solving decentralized MDPs is NEXP-hard [120]. While there has been a long history of arguing on the criterion of decision making (satisficing or optimizing [121]) [1], there is no doubt that agents using the MDP approaches can help people teammates make decisions on well-defined problems.

Another key issue in teamwork is distributed resource allocation and multiagent planning [45]. Researchers have used auctions [123], distributed constraint optimization [124], and other techniques [125] to accomplish such distributed resource allocations. However, this topic deserves a survey paper in its own right and we choose not to expose it here. Interested readers may find the above pointers useful in exploring this topic.

In addition, although we have not mentioned, agent technology is particularly useful in designing self-learning or self-evolving systems. Take the field of language evolution as an

[1] For example, researchers in the camp of naturalistic decision making take the opinion that when making decisions, people usually do not know the probabilities of all the choices. They even do not know all the possible options. Communities dealing with time stress tasks often demand simulation systems with realistic (human-like) decision representation [122].

example. Recently, there is a growing interest in using mathematical and computational modeling to explore issues relevant for understanding the origin and evolution of language [126]. For instances, Hurford described computer simulations in which coordinated language systems emerge within populations of artificial agents through iterated learning across generations [126]; Perlovsky [127] suggested a computational mechanism that the mind might use for integrating cognition and language. A distant view in this area might be to employ advanced intelligent agent technologies in conducting certain realistic simulations to test hypothetical theories about the mechanisms underlying the evolution of language.

VII. SUMMARY

Teamwork modeling has spanned diverse disciplines from business management to cognitive science, human discourse, and distributed artificial intelligence. This article organized the existing work in this field along two dimensions: team social structure and social behaviors. Agent-only teams and mixed human/agent teams were considered along the dimension of social structure. Shared mental models, collaborative behaviors, communicative behaviors, and helping behaviors were considered along the dimension of social behaviors.

A list of systems for developing agent-only teams were examined first. The shared mental models of these systems were generalized into three categories: based on blackboards, based on rules, and based on joint mental attitudes. The collaborative behaviors were analyzed at three levels: atomic team actions, coordinated tasks, and planned team activities; most of the systems support planned team activities. Communicative behaviors of the systems were analyzed by examining whether they support reactive, proactive, and selective communications. Helping behaviors of the systems were examined from the perspective of activeness (reactive and proactive).

A list of systems for developing mixed human/agent teams were reviewed next. We grouped the systems into three categories: mixed-initiative user interfaces, human-centered teamwork, and human-guided teamwork. While it is hard to classify the shared mental models implemented in these systems as we did for agent-only teams, interface agent technology is unanimously employed in the systems to represent the shared understanding between agents and their human users. In human-agent interactions it is critical to ensure interruptions of humans by an agent be done judiciously and infrequently. Thus, most of the systems have sought to achieve selective communications. Domain-dependent collaborative behaviors and helping behaviors were also identified from the systems, with some principles highlighted.

Modeling and simulating human teamwork behaviors is still a research field in its infancy and with many open issues. A few such open issues are: while interface agents promise to achieve "adaptable division of labor between humans and computers [107]", it is still a challenge for one interface agent to effectively collaborate with multiple human teammates simultaneously. When agents work together with humans, one complex issue is how to ensure that "the cost of human-agent coordination (e.g., the monitoring of agents raises new tasks to humans) does not exceed the value of assistance offered by agents [29]." Rather than focusing on identifying challenges and open issues in this field, this article gives an organizational framework for analyzing teamwork simulation systems and for further studying simulated teamwork behaviors.

ACKNOWLEDGMENT

This research has been supported by AFOSR MURI grant No. F49620-00-1-0326. We would like to thank Dr. Frank Ritter at PSU for his valuable comments and suggestions.

REFERENCES

[1] J. R. Katzenbach and D. K. Smith, "The discipline of teams," *Harvard Business Review*, vol. 71, no. 2, pp. 111–120, 1993.

[2] E. Salas, T. Dickinson, S. Converse, and S. Tannenbaum, "Towards an understanding of team performance and training," in *Teams: Their training and performance*, R. Swezey and E. Salas, Eds. Norwood, 1992, pp. 3–29.

[3] P. R. Cohen, H. J. Levesque, and I. A. Smith, "On team formation," in *Contemporary Action Theory*, J. Hintikka and R. Tuomela, Eds., 1997.

[4] J. R. Katzenbach and D. K. Smith, *The Wisdom of Teams: Creating the High-Performance Organization*. Harvard Business School Press, 1993.

[5] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA, 1999.

[6] J. Orasanu, "Shared mental models and crew performance," in *Proceedings of the 34 Annual Meeting of the Human Factors Society*, Orlando, FL, 1990.

[7] P. R. Cohen and H. J. Levesque, "Teamwork," *Nous*, vol. 25, no. 4, pp. 487–512, 1991.

[8] T. L. Lennox, T. Payne, S. K. Hahn, M. Lewis, and K. Sycara, "Moksaf: How should we support teamwork in human-agent teams?" Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-99-31, September 1999.

[9] "Surviving the group project: A note on working in teams," Available at http://web.cba.neu.edu/ ewertheim/teams/ovrvw2.htm.

[10] D. O. Anderson, "Working in groups and teams," Available at http://www2.latech.edu/ dalea/instruction/teamwork.html, 1998.

[11] J. Scolaro and T. Santarelli, "Cognitive modeling teamwork, taskwork, and instructional behavior in synthetic teammates," in *Proceedings of the Eleventh Conference on Computer-Generated Forces and Behavior Representation*, Orlando, FL, 2002, pp. 315–322.

[12] S. Shina, "New rules for world-class companies," *IEEE Spectrum*, vol. 28, no. 7, pp. 22–37, 1991.

[13] E. Norling and F. E. Ritter, "Towards supporting psychologically plausible variability in agent-based human modeling," in *Proceedings of the third international joint conference on autonomous agents and multi-agent systems*. ACM Press, 2004, pp. 758–765.

[14] "Robocup rescue," http://www.rescuesystem.org/robocuprescue/.

[15] D. Alberts, J. Garstka, and F. Stein, *Network Centric Warfare*. CCRP, 2002.

[16] J. L. Eilbert, G. E. Campbell, T. Santoro, T. L. Amerson, and J. A. Cannon-Bowers, "The role of cognitive agents in the design of complex systems," in *Proceedings of I/ITSEC*, Orlando, FL, 1998.

[17] E. Horvitz, C. M. Kadie, T. Paek, and D. Hovel, "Models of attention in computing and communications: From principles to applications," *Communications of the ACM*, vol. 46, no. 3, pp. 52–59, 2003.

[18] H. Gimblett, R. Itami, and M. Richards, "Simulating wildland recreation use and conflicting spatial interactions using rule-driven intelligent agents," in *Integrating GIS and Agent based modeling techniques for Understanding Social and Ecological Processes*, H. R. Gimblett, Ed. Oxford University Press, 2000.

[19] D. E. Sackett, "Simulations to save time, money, and lives," *Science and Technology Review*, no. 11, pp. 4–11, 1996.

[20] Loral, *ModSAF User Manual (Version 1.5.1)*, Cambridge, MA: Loral Advanced Distributed Simulation, 1995.

[21] W. J. Clancey, P. Sachs, M. Sierhuis, and R. van Hoof, "Brahms: Simulating practice for work systems design," *International Journal of Human-Computer Studies*, vol. 49, pp. 831–865, 1998.

[22] A. A. R. Clemins, "Naval simulation system(nss): An assessment of it-21 warfighting value-added," USN, Briefing, 1999.

[23] M. Tambe, "Towards flexible teamwork," *Journal of AI Research*, vol. 7, pp. 83–124, 1997.

[24] J. Yen, J. Yin, T. Ioerger, M. Miller, D. Xu, and R. Volz, "CAST: Collaborative agents for simulating teamworks," in *Proceedings of IJCAI'2001*, 2001, pp. 1135–1142.

[25] Y. Zhang, L. He, K. Biggers, J. Yen, and T. Ioerger, "Simulating teamwork and information flow in tactical operations centers using multi-agent systems," in *Proceedings of the Tenth Conference on Computer Generated Forces and Behavioral Representation*, Norfolk, VA, 2001, pp. 529–539.

[26] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, 1993.

[27] J. Nerb, H. Spada, and A. M. Ernst, "A cognitive model of agents in a commons dilemma," in *Proceedings of the 19th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, 1997, pp. 560–565.

[28] T. Payne, T. L. Lenox, S. K. Hahn, K. Sycara, and M. Lewis, "Agent-based support for human/agent teams," in *CHI 2000 Conference on Human Factors in Computing Systems*. New York, NY: ACM Press, April 2000.

[29] J. Bradshaw, M. Sierhuis, A. Acquisti, Y. Gawdiak, D. Prescott, R. Jeffers, N. Suri, and R. van Hoof, "What we can learn about human-agent teamwork from practice," in *Workshop on Teamwork and Coalition Formation, Autonomous Agents and Multi-agent Systems (AAMAS 02)*, Bologna, Italy, 2002.

[30] T. Dickonson and R. McIntyre, "A conceptual framework for teamwork measurement," in *Team Performance Assessment and Measurement*, M. Brannick, E. Salas, and C. Prince, Eds. Mahwah New Jersey: Lawrence Erlbaum, 1997, pp. 19–44.

[31] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[32] M. Genesereth and S. Ketchpel, "Software agents," *CACM - Special Issue on Intelligent Agents*, vol. 37, no. 7, pp. 48–53, 1994.

[33] M. J. Wooldridge and N. R. Jennings, "Pitfalls of agent-oriented development," in *Proc. of the 2nd Int. Conf. on Autonomous Agents*, Minneapolis, USA, 1998, pp. 385–391.

[34] M. Georgeff and A. L. Lansky, "Reactive reasoning and planning," in *Proceedings of the sixth National Conference on Artificial Intelligence*, Seattle, WA, 1987, pp. 677–682.

[35] A. Rao and M. Georgeff, "Bdi-agents: from theory to practice," in *Proceedings of the First Intl. Conference on Multiagent Systems*, 1995.

[36] R. Fagin, J. Y. Halpern, Y. Moses, and M. Vardi, *Reasoning About Knowledge*. MIT Press, 1995.

[37] A. Rao and M. Georgeff, "Formal models and decision procedures for multi-agent systems," Australian Artificial Intelligence Institute, Melbourne, Australia, Tech. Rep. 61, 1995.

[38] A. Birk, H. Kenn, and T. Walle, "Robocube, a universal special-purpose hardware for the robocup small robots league," in *Distributed Autonomous Robotic Systems III*, D. Lueth and W. Dario, Eds. Springer, 1998.

[39] M. N. Huhns and M. P. Singh, "Cognitive agents," *IEEE Internet Computing*, no. 11&12, pp. 87–89, 1998.

[40] J. Laird, A. Newell, and P. Rosenbloom, "Soar: an architecture for general intelligence," *Artificial Intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[41] F. E. Ritter, Ed., *Techniques for Modeling Human Performance in Synthetic Environments: A Supplementary Review*, 2002.

[42] J. R. Anderson and C. Lebiere, *The atomic components of thought*. Mahwah, NJ: Erlbaum, 1998.

[43] R. Nibbelke and P. Emmerson, "Review of the software iGEN," *Ergonomics in Design*, vol. 8, no. 3, pp. 29–31, 2000.

[44] *Cougaar: Cognitive Agent Architecture*, http://www.cougaar.org/, 2004.

[45] K. Sycara, "Multiagent systems," *AI Magazine*, vol. 19, no. 2, pp. 79–92, 1998.

[46] M. N. Huhns and L. M. Stephens, "Multiagent systems and societies of agents," in *Multiagent systems: a modern approach to distributed artificial intelligence*, G. Weiss, Ed. MIT Press, 1999, pp. 79–120.

[47] V. R. Lesser, "Multiagent systems: an emerging subdiscipline of ai source," *ACM Computing Surveys*, vol. 27, no. 3, pp. 340–342, 1995.

[48] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[49] H. J. Levesque, P. R. Cohen, and J. Nunes, "On acting together," in *Proceedings of AAAI-90*. Morgan Kaufmann Publishers, Inc., 1990, pp. 94–99.

[50] B. Grosz and S. Kraus, "Collaborative plans for complex group actions," *Artificial Intelligence*, vol. 86, pp. 269–358, 1996.

[51] ——, "The evolution of SharedPlans," in *Foundations and Theories of Rational Agencies*, A. Rao and M. Wooldridge, Eds., 1999, pp. 227–262.

[52] N. R. Jennings, "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions," *Artificial Intelligence*, vol. 75, no. 2, pp. 195–240, 1995.

[53] C. Rich and C. Sidner, "Collagen: When agents collaborate with people," in *Proceedings of the International Conference on Autonomous Agents (Agents'97)*, 1997.

[54] B. Grosz and C. Sidner, "Plans for discourse," in *Intentions in communication*, P. Cohen, J. Morgan, and M. Pollack, Eds. MIT Press, 1990.

[55] J. Grunig, “Information and decision making in economic development,” *Journalism Quarterly*, vol. 46, pp. 565–575, 1969.

[56] T. Dickinson and R. McIntyre, “A conceptual framework for teamwork measurement,” in *Team performance assessment and measurement: theory, methods and applications*, M. T. Brannick, E. Salas, and C. Prince, Eds. Hillsdale, NJ:Erlbaum, 1997, pp. 19–44.

[57] R. McIntyre and E. Salas, “Measuring and managing for team performance: emerging principles from complex environments,” in *Team effectiveness and decision making in organizations*, R. Guzzo and E. Salas, Eds. San Francisco: Jossey-Bass, 1995, pp. 149–203.

[58] K. A. Smith-Jentsch, J. H. Johnson, and S. C. Payne, “Measuring team-related expertise in complex environments,” in *Making Decisions Under Stress: Implications for Individual and Team Training*, J. A. Cannon-Bowers and E. Salas, Eds. American Psychological Association, 1998, pp. 61–87.

[59] J. Yen, X. Fan, and R. A. Volz, “Information-needs in agent teamwork,” *Web Intelligence and Agent Systems: An International Journal*, vol. 2, no. 3, 2004.

[60] A. Ellis, J. Hollenbeck, D. Ilgen, C. Porter, B. West, and H. Moon, “Capacity, collaboration, and commonality: A framework for understanding team learning,” in *6th International Command and Control Research and Technology Symposium*, Annapolis, MD, 2001.

[61] K. Lochbaum, *Using Collaborative Plans to Model the Intentional Structure of Discourse*. PhD thesis, Harvard University, Tech Report TR-25-94, 1994.

[62] J. Hollenbeck, D. Sego, D. Ilgen, D. Major, J. Hedlund, and J. Phillips, “Team decision making accuracy under difficult conditions: Construct validation of potential manipulations and measures using the TIDE^2 simulation,” in *New Directions in Team Measurement*, M. Brannick, E.Salas, and C. Prince, Eds. San Francisco: Jossey Bass, 1997, pp. 111–136.

[63] L. A. Coser, *The functions of social conflict*. Glencoe, IL: The Free Press, 1956.

[64] D. Tjosvold, “Cooperative and competitive goal approach to conflict: Accomplishments and challenges,” *Applied Psychology*, vol. 47, no. 3, pp. 285–313, 1998.

[65] E. Rayeski and J. D. Bryant, “Team resolution process: A guideline for teams to manage conflict, performance, and discipline,” in *Proceedings of The International Conference on Work Teams: Anniversary Collection*, M. Beyerlein and M. Bullock, Eds., 1994, pp. 215–221.

[66] J. B. Stulberg, *Taking charge / managing conflict*. Lexington, MA: Lexington Books, 1987.

[67] G. H. Varney, *Building productive teams: An action guide and resource book*. San Francisco, CA: Josey-Bass, 1989.

[68] J. Uhlarik and D. A. Comerford, “A review of situation awareness literature relevant to pilot surveillance functions,” DOT/FAA, Tech. Rep. AM-02/3, 2002.

[69] M. Endsley, “Towards a theory of situation awareness in dynamic systems,” *Human Factors*, vol. 37, pp. 32–64, 1995.

[70] P. Pearce and P. Amato, “A taxonomy of helping: A multidimensional scaling analysis,” *Social Psychology Quarterly*, vol. 43, no. 4, pp. 363–371, 1980.

[71] S. E. Anderson and L. J. Williams, “Interpersonal job, and individual factors related to helping processes at work,” *Journal of Applied Psychology*, vol. 81, pp. 282–296, 1996.

[72] T. Dickinson and R. McIntyre, “A conceptual framework for teamwork measurement,” in *Team Performance Assessment and Measurement: Theory, Methods and Applications*, M. Brannick, E. Salas, and C. Prince, Eds., 1997, pp. 19–44.

[73] C. Porter, J. Hollenbeck, D. Ilgen, A. Ellis, B. West, and H. Moon, “Towards understanding backing up behaviors in

work teams: The role of personality and the legitimacy of need for backing up others," *Journal of Applied Psychology*, vol. 88, pp. 391–403, 2003.

[74] J. A. Cannon-Bowers, E. Salas, and S. A. Converse, "Shared mental models in expert team decision making," in *Individual and group decision making*, Castellan, NJ, 1993, pp. 221–246.

[75] J. R. Rentsch and R. J. Hall, "Members of great teams think alike: A model of team effectiveness and schema similarity among team members," in *Advances in interdisciplinary studies of work teams (Vol. 1. Series on self-managed work teams)*, M. M. Beyerlein and D. A. Johnson, Eds. Greenwich, CT: JAI Press, 1994, pp. 223–262.

[76] W. Rouse, J. Cannon-Bowers, and E. Salas, "The role of mental models in team performance in complex systems," *IEEE Trans. on Sys., man, and Cyber*, vol. 22, no. 6, pp. 1296–1308, 1992.

[77] K. Sycara and M. Lewis, "Forming shared mental models," in *Proceedings of the 13th Annual Meeting of the Cognitive Science Society*, 1991, pp. 400–405.

[78] D. Martin, A. Cheyer, and D. Moran, "The open agent architecture: A framework for building distributed software systems," *Applied Artificial Intelligence*, vol. 13, no. 1-2, pp. 91–128, 1999.

[79] J. Giampapa and K. Sycara, "Team-oriented agent coordination in the RETSINA multi-agent system," in *tech. report CMU-RI-TR-02-34, Robotics Institute, Carnegie Mellon University*, 2002.

[80] D. Pynadath, M. Tambe, N. Chauvat, and L. Cavedon, "Toward team-oriented programming," in *Agent Theories, Architectures, and Languages*. Springer, 1999, pp. 233–247.

[81] K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng, "Distributed intelligent agents," *IEEE Expert, Intelligent Systems and their Applications*, vol. 11, no. 6, pp. 36–45, 1996.

[82] W. Zachary, J. Ryder, L. Ross, and M. Weiland, "Intelligent human-computer interaction in real time, multi-tasking process control and monitoring systems," in *Human Factors in Design for Manufacturability*, M. Helander and M. Nagamachi, Eds., 1992, pp. 377–402.

[83] W. Zachary, J.-C. Le Mentec, and J. Ryder, "Interface agents in complex systems," in *Human interaction with complex systems: Conceptual Principles and Design Practice*, C. Ntuen and E. Park, Eds. Norwell, MA: Kluwer Academic Publishers, 1996.

[84] W. W. Zachary and J. L. Mentec, "Modeling and simulating cooperation and teamwork," in *Military, Government, and Aerospace Simulation*, M. J. Chinni, Ed., 2000, vol. 32, pp. 145–150.

[85] M. Kang, "Team-soar: A computational model for multilevel decision making," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 31, no. 6, pp. 708–714, 2001.

[86] J. R. Hollenbeck, D. R. Ilgen, D. J. Sego, J. Hedlund, D. A. Major, and J. Phillips, "The multi-level theory of team decision making: Decision performance in teams incorporating distributed expertise," *Journal of Applied Psychology*, vol. 80, pp. 292–316, 1995.

[87] "Robocup soccer," http://www.robocup.org/.

[88] E. Pagello, E. Menegatti, A. Bredenfel, P. Costa, T. Christaller, A. Jacoff, D. Polani, M. Riedmiller, A. Saffiotti, E. Sklar, and T. Tomoichi, "Robocup-2003: New scientific and technical advances," *AI Magazine*, vol. 25, no. 2, pp. 81–98, 2004, online at http://www.aass.oru.se/˜asaffio/.

[89] H. Kitano and S. Tadokoro, "Robocup-rescue: A grand challenge for multi-agent and intelligent systems," *AI Magazine*, vol. 22, no. 1, pp. 39–52, 2001.

[90] D. Fensel, I. Horrocks, F. Harmelen, D. McGuinness, and P. F. Patel-Schneider, "Oil: An ontology infrastructure for the semantic web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 38–45, 2001.

[91] R. Tuomela and K. Miller, “We-intentions,” *Philos. Stud*, vol. 53, no. 2&3, pp. 367–389, 1988.

[92] P. R. Cohen, A. J. Cheyer, M. Wang, and S. C. Baeg, “An open agent architecture,” in *AAAI Spring Symposium*, 1994, pp. 1–8.

[93] J. Yen, X. Fan, S. Sun, M. Miller, R. A. Volz, R. Wang, C. Chen, and K. Kamali, “Formal semantics and communication strategies for proactive information delivery among team-based agents,” in *Proceedings of the second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS03)*, 2003, pp. 1166–1167.

[94] J. Searle, *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969.

[95] J. Austin, *How to Do Things with Words*. Oxford University Press: Oxford, England, 1962.

[96] P. R. Cohen and H. J. Levesque, “Rational interaction as a basis for communication,” in *Intentions in Communication*. MIT Press, 1990, pp. 221–255.

[97] P. Breiter and M. Sadek, “A rational agent as a kernel of a cooperative dialogue system: Implementing a logical theory of interaction,” in *Proceedings of ECAI-96 workshop on Agent Theories, architectures, and Languages*. Springer-Verlag, Berlin, 1996, pp. 261–276.

[98] Y. Labrou and T. Finin, “Semantics and conversations for an agent communication language,” in *Readings in Agents*, M. Huhns and M. Singh, Eds. Morgan Kaufmann, San Mateo, Calif., 1998, pp. 235–242.

[99] *FIPA: Agent Communication Language Specification*, http://www.fipa.org/, 2002.

[100] P. Stone and M. Veloso, “Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork,” *Artificial Intelligence*, vol. 110, pp. 241–273, 1999.

[101] X. Fan, J. Yen, R. Wang, S. Sun, and R. A. Volz, “Context-centric proactive information delivery,” in *Proceedings of the 2004 IEEE/WIC Intelligent Agent Technology conference (to appear)*, 2004.

[102] J. Yen, X. Fan, and R. Volz, “Towards a theory for proactive information delivery in agent teamwork,” *Artificial Intelligence*, accepted.

[103] M. Marks, S. Zaccaro, and J. Mathieu, “Performance implications of leader briefings and team interaction training for team adaptation to novel environments,” *Journal of Applied Psychology*, vol. 85, pp. 971–986, 2000.

[104] J. Yen, X. Fan, S. Sun, T. Hanratty, and J. Dumer, “Agents with shared mental models for enhancing team decision-makings,” *Decision Support Systems*, p. (in press), 2004.

[105] M. Lewis, “Designing for human-agent interaction,” *Al Magazine*, vol. 19, no. 2, pp. 67–78, 1998.

[106] E. Horvitz and T. Paek, “A computational architecture for conversation,” in *Proceedings of the Seventh International Conference on User Modeling*, Banff, Canada, 1999, pp. 201–210.

[107] C. Rich, C. Sidner, and N. Lesh, “COLLAGEN: Applying collaborative discourse theory to human-computer interaction,” *Artificial Intelligence Magazine*, vol. 22, no. 4, pp. 15–25, 2001.

[108] J. Rickel, N. Lesh, C. Rich, C. Sidner, and A. Gertner, “Collaborative discourse theory as a foundation for tutorial dialogue,” in *LNCS-2363: Proc. of International Conference on Intelligent Tutoring Systems (ITS)*. Springer, 2002, pp. 542–551.

[109] M. Sierhuis, J. M. Bradshaw, A. Acquisti, R. van Hoof, R. Jeffers, and A. Uszok, “Human-agent teamwork and adjustable autonomy in practice,” in *7th International Symposium on Artificial Intelligence (I-SAIRAS)*, NARA, JAPAN, 2003.

[110] T. L. Lenox, “Supporting teamwork using software agents in human-agent teams,” Ph.D. dissertation, Westminster College, Market Street,New Wilmington, PA 16172, April 2000.

[111] P. Scerri, D. V. Pynadath, L. Johnson, P. Rosenbloom, N. Schurr, and M. Tambe, “A prototype infrastructure for distributed robot-agent-person teams,” in *AAMAS’03*, July 2003, pp. 433–440.

[112] J. Yen, X. Fan, S. Sun, M. McNeese, and D. Hall, "Supporting anti-terrorist analyst teams using agents with shared rpd process," in *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safty*, 2004, pp. 15–23.

[113] M. Sierhuis, "Brahms: A multi-agent modeling and simulation language for work system analysis and design," Ph.D. dissertation, University of Amsterdam, 2001.

[114] G. A. Klein, "The recognition-primed decision (rpd) model: Looking back, looking forward," in *Naturalistic decision making*, C. E. Zsambok and G. Klein, Eds., 1997, pp. 285–292.

[115] T. Erickson, "Designing agents as if people mattered," in *Software agents*, J. M. Bradshaw, Ed. Menlo Park, CA: AAAI Press, 1997, pp. 79–96.

[116] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman, "Transition-Independent Decentralized Markov Decision Processes," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems*. Melbourne, Australia: ACM Press, July 2003, pp. 41–48.

[117] H. Jung and M. Tambe, "Performance models for large scale multiagent systems: using distributed pomdp building blocks," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM Press, 2003, pp. 297–304.

[118] J. Shen, V. Lesser, and N. Carver, "Minimizing Communication Cost in a Distributed Bayesian Network using a Decentralized MDP," in *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, vol. AAMAS03. Melbourne, AUS: ACM Press, July 2003, pp. 678–685.

[119] R. Nair, M. Roth, M. Yokoo, and M. Tambe, "Communication for Improving Policy Computation in Distributed POMDPs," in *Proceedings of The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. AAMAS04. ACM Press, July 2004.

[120] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Math. Oper. Res.*, vol. 27, no. 4, pp. 819–840, 2002.

[121] H. Simon, "A behavioral model of rational choice," *Quarterly Journal of Economics*, vol. 69, pp. 99–118, 1955.

[122] J. Sokolowski, "Can a composite agent be used to implement a recognition-primed decision model," in *Proceedings of the Eleventh Conference on Computer Generated Forces and Beharioral Representation*, 2002, pp. 431–436.

[123] L. Hunsberger and B. Grosz, "A combinatorial auction for collaborative planning," in *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, 2000, pp. 151–158.

[124] R. Mailler and V. Lesser, "Using Cooperative Mediation to Solve Distributed Constraint Satisfaction Problems," *Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, 2004.

[125] R. Nair, M. Tambe, and S. Marsella, "Role Allocation and Reallocation in Multiagent Teams: Towards a Practical Analysis," in *Proceedings of The Second International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. AAMAS03. ACM Press, July 2003.

[126] M. H. Christiansen and S. Kirby, Eds., *Language Evolution, Studies in the Evolution of Language*. Oxford, 2003.

[127] L. Perlovsky, "Integrating language and cognition," *IEEE Connections*, vol. 2, no. 2, pp. 8–12, 2004.