1

Emotion-Based Control of Cooperating Heterogeneous Mobile Robots

Robin R. Murphy, Christine Lisetti, Russ Tardif, Liam Irish, Aaron Gage

Abstract— **Previous experiences show that it is possible for agents such as robots cooperating asynchronously on a sequential task to enter situations where one robot does not fulfill its obligations in a timely manner due to hardware or planning failure, unanticipated delays, etc., leaving the other robot in an infinite wait state. Our approach is derived from a formal multilevel process theory of emotions where emotions both modify active behaviors at the sensory-motor level and change the set of active behaviors at the schematic level. The resulting implementation of a team of heterogeneous robots using a hybrid deliberative/reactive architecture produced the desired emergent cooperative behavior. Data collected at two different public venues illustrate how a dependent agent selects new behaviors (e.g., stop serving, move to intercept the refiller) to compensate for delays from a subordinate agent (e.g., blocked by the audience). The subordinate also modifies the intensity of its active behaviors in response to feedback from the dependent agent. The agents communicate asynchronously through KQML via wireless Ethernet.**

I. Introduction

This article describes an approach to multi-agent control for interdependent tasks which embues the agents with emotions, allowing a satisfactory societal behavior to emerge. The use of emotions allows individual robots with different roles in the team to dynamically adapt their local active behaviors and select new behaviors in order to accomplish the overall mission. The robots respond to the environment and task progress individually without centralized planning and only a minimal amount of communication to maintain synchronization for cooperation.

For the purposes of this article, *interdependent tasks* are those where *one or more robots execute a tightly coupled sequence with a cyclical dependency.* Robots working cooperatively on an interdependent task are said to be interdependent. Examples include a robot assistant resupplying another robot in the field or one robot docking with another to be recharged or transported to a new site. Note that in interdependent tasks one robot must wait upon a real resource to be transferred from one robot to the other. In the case of resupply, the robot in the field must receive a resource before continuing; the use of a second robot to transport that resource introduces the inter-robot dependency. The delivery robot cannot make a delivery until the field robot requests it and provides a rendevouz location, completing the cycle. In the case of docking, the docking robot is the resource; the robot serving as the docking station cannot complete its task (recharging, transportation) until the docking robot has been physically coupled. Likewise, the docking robot cannot dock if the docking station robot is not ready.

Cooperation between robots to accomplish a common goal does not necessarily mean the robots are interdependent. For example, cooperative box pushing is a task that appears to require interdependence, with the box being the shared resource. However, the two best-known approaches to multi-agent box pushing, Mataric et al. [1] and Parker [2], [3], use the technique of pushing the left and then the right sides of an elongated box. In [1], both can push on the box at the same time, breaking any cyclic dependency between robots. Parker introduces an explicit dependency between robots by subdividing the task into `push-right` and `push-left`, then assigning these subtasks to each robot. However, when only one robot is available, both methods degenerate to one robot pushing alternate sides, eliminating true inter-robot dependency. In the definition of interdependency for this article, one robot cannot perform the other one's task.

Interdependent tasks are interesting to the behavior-based robotics community for at least three reasons. First, there is the possibility that one robot will fail to meet its obligations in a timely manner due to hardware failure, planning failure, or an adverse environment; the other robot could be kept waiting forever, rendering both robots useless. An open issue is how to detect and respond to, or break, this cyclic dependency in a distributed team where centralized reasoning is not supported. Second, cooperation may dynamically improve performance, beyond the elimination of unexpected waiting. In the case of resupply, the dependent agent may experience a higher than expected demand for its resources, which should lead to a faster servicing from its subordinate. The subordinate must have some mechanism for modifying its behaviors in response to the needs of the dependent agent without losing autonomy. Robots which optimize or opportunistically accelerate the transfer of the shared resource should complete the mission faster. An open issue is how these performance improvements would arise in a distributed system. Third, interdependent tasks in practice may be accomplished with heterogeneous robot teams. In the example of docking, the "mother" robot is physically and behaviorally different than the "daughter" robots. This introduces the open issue of control mechanisms which can be realistically implemented on a heterogeneous team.

The approach taken in this article is to investigate the application of a formal cognitive model of emotions, whereby the intensity and choice of behaviors of each robot is self-regulated by their emotional state. For the purposes

Murphy, Tardif, Irish, and Gage are with the Dept. of Computer Science and Engineering, University of South Florida, Tampa FL 33620, USA; murphy, irish, tardif, agage @csee.usf.edu, Lisetti is with Dept. of Computer Science, University of Central Florida, Orlando, FL, USA; lisetti@cs.ucf.edu

of this article, a robot is said to have emotions if, to paraphrase Zajonc, it has the capacity to distinguish and adapt to its environment which may be harmful or beneficial to it or its multi-robot system. [4], [5] The primary benefit of emotions is that it enables adaptation to harmful conditions without having to reason about the cause.

The motivation for using emotions is our position that newly gained knowledge on emotional intelligence will lead to robots capable of representing and learning affective knowledge thereby rendering the team more autonomous and efficient. We acknowledge that the same results may be obtained without the use of cognitive models (see Sec. II). On the other hand, we believe that it is premature to rule out any possible techniques, and that cognitively-oriented approaches provide an interesting source for comparison and contrast with engineering solutions.

The work reported in this article is the first that we know of to use a formal cognitive model of emotions to improve performance of multiple robots, either homogeneous or heterogeneous, working cooperatively on interdependent tasks. Sec. II describes the work of other researchers in applying emotions to robots as well as discusses solutions to the cyclic dependency. Sec. III describes a distributed control scheme using a formal multilevel model of emotions following [6] compatible with hybrid deliberative/reactive architectures [7]. Sec. IV describes the implementation on a pair Nomadic Technologies Nomad 200 robots with heterogeneous sensor and behavioral suites and demonstrations at the the 2000 AAAI Mobile Robot Competition in Austin, Texas, where it won numerous awards, and at the Museum of Science and Industry (MOSI) in Tampa, Florida. The data collected at those venues and an analysis of the results are reported in Sec. V, followed by a discussion of the remaining open issues (how to specify a set of emotions for a task, scalability, etc.).

II. Related Work

The topics most closely related to our efforts are cooperative multiple robot teams, explicit use of emotions for control, computational models of emotions, and deadlock in multi-agent systems. Cooperation between mobile robots has been the subject of much work, though as noted in the Introduction, true interdependencies do not appear to have been examined. Emotions and emotion-based control have been considered for robotics and agent applications. However, as will be shown below, these efforts have largely considered emotions for control of a single robot or agent. As a result, a coherent framework for implementing emotions in heterogeneous robots is missing. Cognitive science provides many possible computational models, of which Leventhal and Scherer's multilevel process theory of emotions [6] shows the most useful degree of correspondence to hybrid robotic architectures. While the motivation for our work is to explore emotions, non-cognitive solutions exist and they are summarized below.

A. Cooperative Teams and Independent Tasks

The aims of Mataric [1] and Parker [8] are most closely related to those reported in this paper, in terms of distributed coordination of robots. However there are significant differences, particularly in the backgrounds and approaches. In particular, the architectures are different, leading different implementation details. Their task focus is also different.

Research efforts directed by Mataric are based on the subsumption architecture, with robots programmed in the Behavior Language. Mataric started on simple homogeneous robots (the Nerd Herd) and built up to more complex systems. In the early system cooperation emerged from the structure of behaviors. There was no awareness of interaction. Later systems such as [9] include communication in order to minimize interference. In contrast, the emotional model used here involves communication, as well as an explicit representation, and awareness of and reaction to emotional stimulus.

Parker's work with ALLIANCE has also been limited to robots working on independent tasks, and also uses the Behavior Language for implementation [2], [8], [3]. While these tasks can be broken into subtasks with ordering dependencies, all tasks are available to all robots involved. If one robot fails a task, the other can take over. ALLIANCE uses two continuous functions with emotional labels, *impatience* and *acquiescence*, but these are not directly based on cognitive science. While it may be possible to do so, the system is not designed to deal with interdependence. It is unclear how ALLIANCE would respond if one robot was waiting on another for a resource. More research must be done as to when one robot has access to a resource which another robot needs. Our work proposes a solution to this resource problem.

The works of Mataric and Parker, while related to the work in this article, do not provide a sufficient implementation framework for the problem of interdependence in hybrid architectures, since they are committed to reactive, subsumption-style [10] architectures. It is interesting to see the incorporation of emotional labels in ALLIANCE, which seems to continue the tradition in behavior-based systems to use biological metaphors, such as motivation. The work in this article is different in scope (it focuses on interdependent tasks, not purely cooperative), architectural assumptions (it assumes a hybrid deliberative/reactive architecture rather than be limited to a purely reactive architecture [7]), and mechanism (it explicitly explores biological emotions as a control mechanism).

B. Application of Emotions to Robots

There have been several attempts to model emotions in software agents [11] and robots[12], [13], [14], [15], [16], [17] and to use these models to enhance functionality. Our work is most similar to Velasquez, Breazeal, and Michaud as described below.

Velasquez [12], [13] is concerned with autonomous agents, and in particular robots, for which control "relies on, and arises from emotional processing." The work describes an emotion-based control framework and focuses on affect programs which are implemented by the integration into specific circuits of several systems that mediate perception, attention, motivation, emotion, behavior, and motor control. These range from simple reflex-like emotions, to facilitation of attention to emotional learning. Although the approach is different, its motivation is similar to ours, and we focus on multi-robot cooperation.

Breazeal [14], [15] also involves robot architectures with a motivational system which associates motivations this time with both drives and emotions. Drives in this architecture help in maintaining an adequate homeostatic regime in three dimensions: social (over the lonely to asocial spectrum), stimulation (over the bored to confused spectrum), and fatigue (leading to exhausted). Emotions are implemented in framework very similar to that of Velasquez's work but Breazeal's emphasis is on the function of emotions in social exchanges and learning with a human caretaker. Our approach is different from Breazeal's in that it is currently less focused on social exchanges than in efficiency of behavior toward goals, and of the use of emotions to control multiple robots in addition to a single agent.

Michaud [16], [17] uses the guidelines of a hybrid-reactive-deliberative architecture, building on top of behavior-producing models connecting sensory inputs to commands. Emotions are largely considered in terms of helping an agent to adapt to limitations, to manage social behavior, and to communicate with others. At the implementation level, emotions monitor the accomplishment of the goals and goals are represented as motives.

In Michaud's work, emotions per se are not represented in the model, but emotional capability is achieved by incorporating it into the control architecture as a global background state. Our approach which chooses to represent the emotional system explicitly (as discussed later) differs from Michaud's in that respect. Although both Michaud's and our approach revolve around the notion of emotion as monitoring progress toward goals, our work explicitly represents emotion and corresponds to a formal cognitive model. Our approach follows Frijda's evolutionary theory of emotions (Frijda, 1986) and associates action tendencies with emotional states (see Table I). For example, the emotion HAPPY is associated with the action tendency to activate freely, CONFIDENT to continue normal activity, FRUSTRATED to change current strategy and so on.

C. Computational Models of Emotions

Most of the computational approaches to emotion theory have used cognitive theories of emotion with implementation designs involving rule-governed symbolic constructs, such as production systems, semantic nets, frames, etc. While there have been quite a few AI models of cognition in which cognition is enhanced by emotion (see Pfeifer [18] for a survey), few AI models have taken emotions as the central focus, and fewer of these have actually been accompanied with a computer program. With the exception of Leventhal and Scherer [6], these models do not have a direct correspondence with structures found in behavior-based robots.

AI approaches to emotion have, for the most part, been based upon Paulhan's (1887) conflict theory [19], in which emotions are thought to occur when an ongoing tendency is interrupted. The conflict approach emphasizes the need to simulate systems with limited resources in an unpredictable world, and with multiple goals and plans which can conflict with each other, and which therefore, must be able to be interrupted. Simon's [20] argument that emotions have a counterpart in computational systems that work with multiple goals in finite time with limited resources is indeed related with Paulhan's theory. This theory provides motivation for our use of emotions to mediate resource conflicts. Other approaches with computational components include Sloman [21], Swagerman [22], Pfeifer [18], and Ortony [23].

A promising theory of emotions which lends itself more directly to computational modeling is the one presented by Leventhal and Scherer: *the multilevel process theory of emotions* [6]; this theory provides the framework for the work reported in this article. The central idea of this theory is that adult emotions are complex behavioral reactions which are constructed from the activity of a hierarchical multi-component processing system. The *sensory-motor level* is activated automatically without deliberate planning by a variety of external stimuli and by internal changes. Emotion reactions based on "pure sensory-motor" processes are mostly of short duration and reflex-like. The *schematic level* integrates sensory-motor processes with prototypes of emotional situations having concrete representations. The *conceptual level* is deliberative and involves reasoning over the past and projecting into the future to avoid repeating emotional disturbances. The relationship of this theory to hybrid architectures will be discussed further in Sec. III.

D. Interference and Deadlock

Researchers have noted that robots working on a mission, or goal for the entire robot team, can end up in conflict. Mataric [24] terms this conflict *interference*. Interference is further subdivided into more complex conflicts, "including goal clobbering, deadlocks and oscillations" [24] Resource competition can be over *space*, *information*, and *objects*. Mataric's work in interference appears to be primarily focused on spatial competition, rather than object competition that occurs in the tasks of resupply and docking. For example, in [25] Goldberg and Mataric use a caste system which gives robots priority to act in given territories in order to solve spatial interference issues.

The undesirable effects of cyclic dependencies in teams of robots are often treated as though they were a deadlock, where no robot is able to make progress. Previous work on deadlocks in robot applications fall into several categories. Lin and Hsu [26] approach deadlocking from the classical operating system perspective. Hartonas-Garmhausen [27], Qutub [28], and Svestka [29] represent the problem as a graph with centralized planning, which is incompatible with the desire for a distributed, reactive solution. Fukuda

[30] and Kube [31] utilize changes in behaviors to break deadlocks. Hara [32] removes the deadlock from the system by changing the physical configuration of the robots. None of the works studied uses emotions to mediate deadlocking. Kube uses an index akin to impatience in ALLIANCE (but not addressed as such) to trigger behaviors. Only Lin, Fukuda, Kube, and Hara deal with objects as the source of deadlocks; the other efforts are concerned with spatial deadlocking.

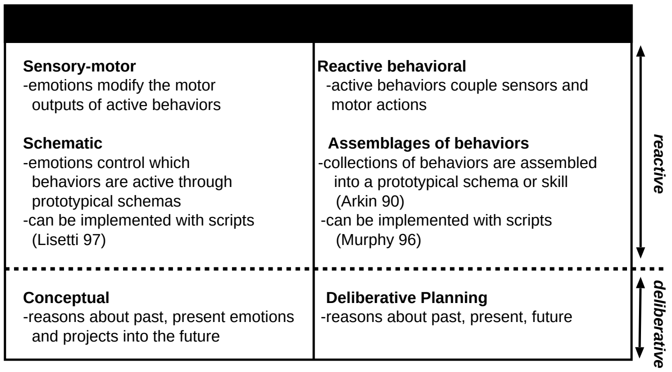
Lin and Hsu [26] apply a classical operating systems approach to handle resource conflicts in an object sorting task. In the object-sorting task, robots are used as resources. A robot can call on another robot to help sort. However, if Robot A calls on Robot B, while Robot B calls on Robot A, deadlock can result. Based on the standard mechanisms for solving deadlock, the paper proposes three solutions: *deadlock detection*, in which if a set of agents is waiting for too long, they are assumed to be in deadlock; *object priority*, in which objects and agents are given priority (such as proximity), and this priority is used to determine who will help with each object, preventing deadlock; and *feasible sequence*, in which an algorithm searches over possible sequences to determine which ordering of tasks will avoid deadlock. Our approach is most similar to deadlock detection, where the use of emotions allows the robots to discover that they are deadlocked and then adapt or change their behavior. Object priority and the method of feasible sequencing require centralized planning, which is at odds with a distributed solution.

Kube and Zhang [31] examine the issue of stagnation in robot teams. Stagnation occurs when a team of robots work on a task but cease to make progress. They use a behavior-based approach to solve the problem of stagnation for multi-robot box pushing. Their solution is to provide a set of exception behaviors that enact different strategies to break the stagnation. *Realignment* causes the robot to change the direction it is pushing. *Reposition* causes the robot to move its place on the box. These strategies have thresholds which increase as lack of progress is perceived. The longer the stagnation, the more drastic the behavior activated. The team was implemented in simulation, but the experiment was focused on the number of team members, rather than issues involved in the success of the behavior strategy in solving deadlocking. As such, it provides no foundation for creating and controlling behaviors for interdependent tasks.

III. Approach

The objectives for using emotions to control teams of robots working on interdependent tasks are to:

- Enable the individual robot to *dynamically adapt* to the context. For example, a robot which is not making sufficient navigational progress might attempt to go faster or take more chances. Note this is conceptually similar to *homeostatic control* [33].
- Enable the individual robot to escalate its response to the point of *changing its behavior altogether* in order to avoid deadlock-like types of situations or task failure. For example, a robot waiting for a resource might choose to stop waiting and proactively navigate to another depot of the resource. Note that this assumes that there are alternative behaviors or depots available.



Sensory-motor -emotions modify the motor outputs of active behaviors	**Reactive behavioral** -active behaviors couple sensors and motor actions
Schematic -emotions control which behaviors are active through prototypical schemas -can be implemented with scripts (Lisetti 97)	**Assemblages of behaviors** -collections of behaviors are assembled into a prototypical schema or skill (Arkin 90) -can be implemented with scripts (Murphy 96)
Conceptual -reasons about past, present emotions and projects into the future	**Deliberative Planning** -reasons about past, present, future

Fig. 1. Relationship of the multilevel process of emotions to the commonly accepted levels of organization in hybrid deliberative/reactive architectures.

For the reasons given in Sec. II-C, Leventhal and Scherer's *multilevel process theory of emotions* [6] serves as the basis for this distributed cooperative control scheme, where task cooperation avoids deadlock-like situations through emergent societal interactions. The central idea is that adult emotions are complex behavioral reactions which are constructed from the activity of a hierarchical multi-component processing system. Failure to make progress on tasks or goals changes the emotional state of the agent, which then produces a multilevel response.

In this hierarchy there are three levels comparable to the commonly accepted groupings of activity in a hybrid deliberative/reactive robot architecture [7]. The three levels are shown in Fig. 1 and are described below:

Sensory-motor. At this basic level, sensed or internal events lead to emotions which then modify the motor output of active behaviors. This is similar to the reflexive behaviors, where the output of a behavior is proportional to the inputs. In this case, the inputs would be perception plus the emotional state.

Schematic. This level integrates sensory-motor processes with prototypes of emotional situations with concrete representations. It corresponds to assemblages of behaviors, or skills.

Conceptual. This level is deliberative since it involves reasoning over the past and projecting into the future. As such, it is beyond the scope of a purely behavioral-based system and so is outside the scope of this project at this time.

[34] has proposed that the schematic level of emotions can be implemented according to a *script* [35]. Scripts have also been used by the robotics community for assemblages of robotic behavioral schemas into larger schemas representing a stereotypical set of behaviors [36], [37], [38], [39]. In particular, Murphy [38] and Rowe et al. [39] have used scripts for robots in a manner consistent with robotic

SCRIPT	EMOTIONAL	BEHAVIORAL
Goal	Action tendency	Task
Places	Social context	Environment
Actors	Emotions	Behaviors
Props	Internal percepts	Percepts, Cues
Causal Chain	Sequence of Events/Beliefs	Sequence of Behaviors
Subscripts	Exception Handling	Exception Handling

TABLE I

AI script compared with emotional, behavioral scripts.

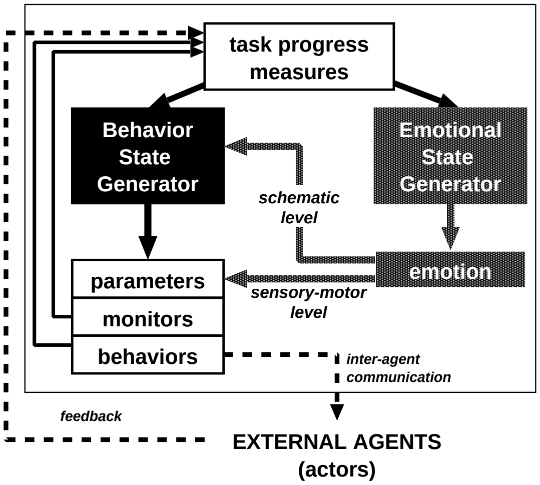


Fig. 2. Layout of a causal chain showing the relationship between the BSG, ESG, and task progress measures.

schema theory [40]. Table I shows the relationship of script components in [35] to the emotional theoretic [34] and behavioral implementations [38], [39] .

Our initial approach was to add the emotional component to the existing behavioral script developed in [38]. The resulting single script is shown in Fig. 2, and consists of two FSAs forming a causal chain dynamically modified by the emotional state.

The script is instantiated with any relevant *a priori* parameters, such as the places, actors, and props. The causal chain is created by the interaction of two finite state machines: a *behavioral state generator (BSG)* and a *emotional state generator (ESG)*. Both the BSG and ESG accept measures of task progress as inputs. The BSG is responsible for selecting the appropriate behaviors and associated parameters and monitors based on task progress and the output of the ESG.

The use of a finite state machine for the emotional influence captured by the ESG was only a first attempt. While satisfactory for this domain, we are currently exploring other mechanisms such as fuzzy logic which can be better integrated with other aspects of emotions and personality.

Task progress metrics come from three sources: *monitors, individual behaviors,* and *inter-agent communication.* Monitors are perceptual schemas which look for conditions or events. Monitors can be further subdivided into *releasers*, after innate releasing mechanisms, which signal the need to activate a new behavior, or *performance monitors*, which use short-term memory to compute script-specific task progress. Individual behaviors often act as releasers for other behaviors. For example, the termination of one behavior usually leads to the activation of another, as per the BSG. An example of a performance monitor is the time spent on the task so far compared to the expected time for the task (passed as a "prop" at script instantiation).

Another source of information about task progress is communication from an external agent, either a *command* (e.g., "hurry") or *data* (e.g. "I'm at location x, y"). A received communication is processed by the task progress monitor component of the script and distributed to both the BSG and ESG pathways. The communication is not confirmed since we are interested in solutions that will work in the presence of failures, including comms failures. The sender has no guarantee that the receiver has heard the message because the lack of a response is ambiguous: it could have heard the message, is responding, but was unable to communicate, or is just not functioning.

This processing may result in a change in behavior through the BSG pathway; for example, Robot A receives a message to do "come here" and the BSG instantiates a move-to-goal behavior. Note that in this case the command acted directly as a releaser. However, the processing may also change the emotional state of the robot via the ESG pathway simultaneously, which then may change or modify the robot's behavior. For example, Robot A receives a message "hurry." The message causes the emotional state to shift to a more negative valence, say from "confident" that it was making good progress to "concerned." As a result of becoming concerned, the ESG triggers adaptation of active behaviors to make better progress. Rather than simply changing the velocity (which is technically not the release of a new behavior), the robot is free to do other things such as reduce the sensitivity to obstacles, coming closer, and change the acceleration of its motions (becoming more jerky and aggressive which might affect positive changes in the environment such as scattering humans from its path) to produce the change. The use of emotions and the ESG pathway breaks the potential master/slave coupling that would occur by using only a BSG pathway, thereby preserving the independence of the robots.

There are two major differences between the emotional script and the abstract behavior script. The first is an implementation detail: the interface with the sensory-motor level. The multi-level hierarchy suggests that the implementation is through separate mechanisms, with the script reserved for only the schematic level; this implementation combined both following the standard usage of scripts in robotics. The second difference is in the lack of a social context which requires situational awareness.

IV. Multi-Agent Implementation

Structures corresponding to the sensory-motor and schematics levels of the multilevel process have been implemented on a pair of heterogeneous robots running under the SFX hybrid deliberative/reactive architecture[41].

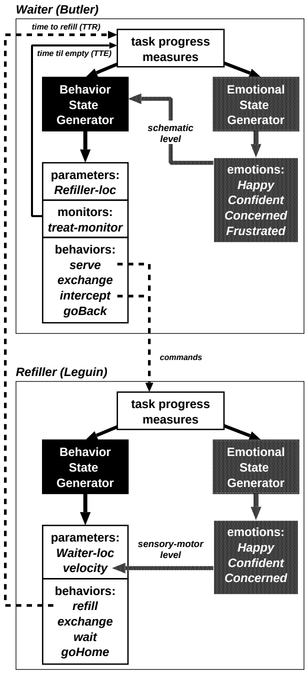


Fig. 3. Implementation showing emotions. Some behaviors and communication links not shown.

These levels are sufficient to meet the objectives of dynamic adaptation and change of behaviors to make task progress. This section describes in detail the task domain, the robotic equipment, and the individual behaviors and emotions on each robot.

The current implementation deviates from the ideal approach in two ways. Both deviations are due to time constraints in programming. First, the WaiterScript implements only the schematic level link between the ESG and BSG and the RefillerScript implements only the sensory-motor level link between the ESG and relevant behavioral parameters. There is no conceptual or practical problem with implementing the two links in the same ESG; rather the implementation reused behaviors and basic scripts developed in 1999, some of which did not have parameters or logic to support both. Second, the BSG and ESG are represented as a single finite state machine. Future instances of the ESG and BSG will enforce a strict separation to permit a direct comparison of the team with and without emotions.

A. Cooperative Task

The AAAI Mobile Robot Competition's annual *Hors D'Oeuvres, Anyone?* event was chosen as the target domain, extending the USF 1999 entry which introduced a novel cooperative robot approach. The goal of the event was for fully autonomous robots to serve finger-foods at a reception, maximizing area covered.

The cooperative society created for *Hors D'Oeuvres, Anyone?* domain consists of two robots: the *Waiter* robot whose task it is to serve items to an audience and a subordinate robot upon which it depends to bring a tray of refills upon request (*Refiller*). It can be shown, that by using a Refiller robot assistant, the Waiter can substantially increase the time on task, i.e. *serving*, since the non-productive travel time to and from the serving station is eliminated. The tray of refills is the resource that can lead to a situation in which the Waiter robot is waiting for the *Refiller* robot to resupply her, but the Refiller is blocked or otherwise unable to meet the request. Emotions are used to adapt or change the team behavior to minimize this non-productive time.

There are three distinguishing features of the society from more traditional cooperative, heterogeneous teams. First, it is assumed that if the Waiter returns to the refill station, it will be refilled regardless of whether the Refiller is there or not. The primary advantage of the society is to eliminate the need for the Waiter to return to the station. Second, the two robots are heterogeneous physically and also in their roles. Therefore, the Refiller cannot perform the Waiter's task and vice versa. Third, the two robots do not operate in a strict master/slave relationship; they are distributed and decentralized.

Fig. 3 shows the basic organization of the reactive layer of SFX. Each robot uses a script, called the `WaiterScript` or `RefillerScript` scripts respectively, representing the strategic task, plus employed tactical behaviors (e.g., avoid-obstacle). The robots use wireless Ethernet to communicate with each other following the KQML agent communication language. The communication is either in the form of a *command* from the Waiter to the Refiller ("refill," "hurry," "intercept," "go home"), or location *data*.

B. Equipment

The multi-agent team consisted of two heterogeneous, fully autonomous Nomadic Technologies Nomad 200 robot bases. Both robots used the Sensor Fusion Effects (SFX) hybrid deliberative/reactive architecture.[41] Both robots run under RedHat Linux version 3.0.3 and are coded in a combination of Lisp, C, C++, TCL-tk, KQML, Perl, lex, and yacc. The Waiter robot (a.k.a. Butler) is equipped with two sonar rings, a Sick planar laser ranger, a thermal probe, and dual Hitachi color video cameras on a pan-tilt head. She is controlled by two on-board processors: a 233 MHz Pentium MMX and 133 MHz Pentium MMX. The Refiller robot (a.k.a. Leguin) has one sonar ring, and dual Hitachi color video cameras on a pan-tilt head. She is controlled by a 233 MHz and a 166 MHz Pentium MMXs. Both robots utilized internal shaft encoders to estimate location relative to a common global coordinate system. The robots can communicate with each other and local workstations using wireless Ethernet.

C. Scripts and Emotions

The `WaiterScript` consists of four strategic behaviors (`serve, exchange, intercept, goBack`), two monitors (`treat-monitor`, `tray-watch`), and four emotions (HAPPY, CONFIDENT, CONCERNED, FRUSTRATED). The `serve` behavior uses a *find-face* sub-behavior to control the pan-tilt head and make eye-contact with the audience while `serve` is active.

ACTION TENDENCY	AT for Waiter/Refiller	EMOTION
FreeActivate	Serve/Joke	Happy
ContinueNormalActivity	Serve;CallforRefill/Serve	Confident
MonitorProgressClosely	AsktoHurry/IncreaseSpeed	Concerned
ChangeCurrentStrategy	Intercept/GoHome	Frustrated

TABLE II

Action tendency table: generic and robot case.

The `RefillerScript` consists of four strategic behaviors (`refill, exchange, wait, goHome`), no internal monitors, and three emotions (HAPPY, CONFIDENT, CONCERNED).

Task progress monitoring was based strictly on inter-agent communication requests from the Waiter. The `wait` behavior also uses an instance of the *find-face* sub-behavior to interact with the audience while waiting at the serving station. Both robots also used a tactical *avoid-obstacle* behavior; tactical, or "survival" behaviors run automatically in parallel to the strategic behaviors as per [41].

The partitioning and labeling of emotions was based on Frijda's theory of emotions [42] which focuses on the functional aspect of emotions. Frijda associates some of the primary emotions (emotions found at the sensory level in Leventhal and Scherer [6]) with distinct and elementary forms of action tendency. Each emotion calls into readiness a small and distinctive suite of actions that has been selected as appropriate to take in that specific emotional state. Thus, in broadly defined recurring circumstances that are relevant to goals, each emotion prompts both the individual and the group in a way that has been evolutionarily more successful than alternative kinds of prompting.

Depending on the physiology of the individual and its current environmnent, evolution has selected and associated specific action tendencies with each emotion. In our case, our two robots have been programmed with the HAPPY, CONFIDENT, CONCERNED and FRUSTRATED emotional states which correspond respectively to the action tendencies shown in table II. The choice of labels for the four states was ad hoc, but the labels themselves have no impact on emotional control.

Table III shows the emotional states for each robot and how the emotions were generated from task progress metrics. The Refiller, Leguin, was to have had a fourth state of FRUSTRATED which would have led to the schematic-level response of changing her behavior to `goHome`, or give up, but was not implemented due to time constraints. The role of personality can be seen as the Waiter could be typlified as "aggressive," since its response to frustration was to change its behavior to a pro-active intercept, while the Refiller's response to frustration was "meek," responding to frustration by giving up. Time constraints did not permit the addition of the FRUSTRATED emotional state on the Refiller.

Butler's emotional state was governed by the changing relationship of the rate of treat consumption, `time til empty (TTE)` to the anticipated time to be refilled, `time to refill (TTR)`. TTR is the time in should take for a refill if the assistant is moving at the expected speed. This is the Euclidean distance divided by the rate. Two modifiers were used, `caution, C`, and `patience, P`, acting as thresholds. The output of the emotions was at the schematic level, leading to changes in the set of active behaviors.

EMOTION	TASK PROGRESS MEASURES	
	WAITER	REFILLER
Happy	$TTE \geq (TTR + C)$	no pending requests
Confident	$TTE < (TTR + C)$	`refill` request
Concerned	$TTE < TTR$	`hurry` request
Frustrated	$TTE < (TTR - P)$	n/a

TABLE III

Table showing the task progress measures for each robot.

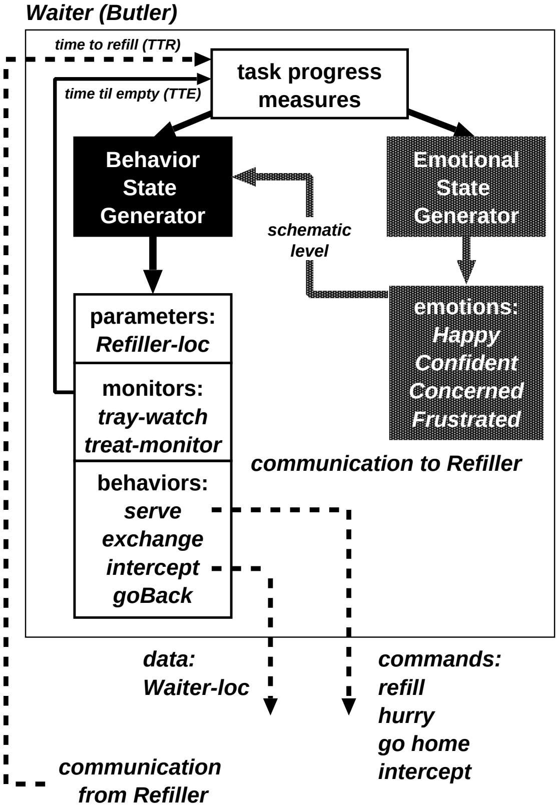

Waiter (Butler)
time to refill (TTR)
time til empty (TTE)
task progress measures
Behavior State Generator
Emotional State Generator
schematic level
parameters: Refiller-loc
monitors: tray-watch treat-monitor
behaviors: serve exchange intercept goBack
emotions: Happy Confident Concerned Frustrated
communication to Refiller
data: Waiter-loc
commands: refill hurry go home intercept
communication from Refiller

Fig. 4. Detail of the implementation of the `WaiterScript`.

D. Waiter and the WaiterScript

The `WaiterScript` in Fig. 4 runs continuously. It has one external input, the communicated data about the location of the Refiller robot. There are six possible external outputs which are communicated to the Refiller, the five commands ("wait," "refill," "hurry," "intercept", "go home") and the position data ("Butler-loc"). The commands are generated by the active behavior in response to the events on the script. "wait" and "refill" are produced by `serve` when the script reaches a procedural milestone event. "hurry, " "intercept", "go home" are generated when an emotional event (state change) occurs. The "intercept" command only causes the Refiller to try to "hurry" if not already doing so. Internally, the script is responsible for computing the `time to refill (TTR)` and `time til empty (TTE)` task progress measures and either instantiating or modifying the set of active behaviors.

The typical scenario follows. When instantiated, the *WaiterScript* begins with the `serve` behavior and sends a command "wait" to the Refiller to ensure that it is in the correct starting state. Under the `serve` behavior, the Waiter navigates to a series of pre-specified waypoints. Her initial emotional state is HAPPY. `Serve` uses the sub-behavior `face-find` to direct the pan-tilt head to search for and track human faces. `find-face` operates in RGB space using a standard $O(n \times m)$ color region segmentation algorithm. If a skin-color affordance is found, the presence of a human would be verified with the thermal probe through behavioral sensor fusion. The `serve` behavior would then be attracted to the largest skin color region rather than the waypoint. `face-find` assumes that the largest blob is a face and directs the pan-tilt head to center the cameras on expected location of the eyes, estimated from the blob dimensions. Once a face has been detected, `serving` plays sound bites encouraging people to remove treats. If the people disappear from view, the robot resumes waypoint navigation.

The `treat-monitor` is also active whenever `serve` is active. It uses laser data to count the number of events; each event is assumed to be the removal of one treat. The laser plane extends over the extent of the tray, and the treat count variable is decremented each time the plane is broken.

While serving, the Waiter robot may communicate a "refill" request if the *time 'til empty* (TTE) is now less than the time required for the Refiller robot to navigate to her (TTR)(see Table III). This does not change the serving behavior on the Waiter. However, her emotional state does change; she is now CONFIDENT that she will receive a refill in time. `serve` terminates only under two conditions. Ideally, the Refiller reaches the Waiter triggering the `exchange` behavior. Under `exchange`, the Waiter does nothing until the `tray-watch` monitor sees the operator flash the empty tray in front of the cameras. When the `tray-watch` monitor returns true indicating that the tray was seen, the Waiter communicates a "go home" command to the Refiller. When `exchange` terminates, the `WaiterScript` re-instantiates `serve`, maintaining the list of waypoints visited, and the Waiter moves to the next on the list or is immediately captured by nearby human faces.

In less than ideal conditions, the Waiter may become CONCERNED and issue a "hurry" command to the Refiller. The Waiter may even become FRUSTRATED with the lack of a refill and activate the `intercept` behavior. When the Waiter issues an `intercept` data message, the Refiller moves into the "hurry" condition, in which she attempts to move at her maximum speed. `intercept` is essentially a move-to-goal behavior where the Waiter then uses the dynamically updated location of the Refiller (`Refiller-loc`) as the value of the goal parameter. Once the Waiter intercepts the Refiller, the `exchange` behavior is triggered and is carried out.

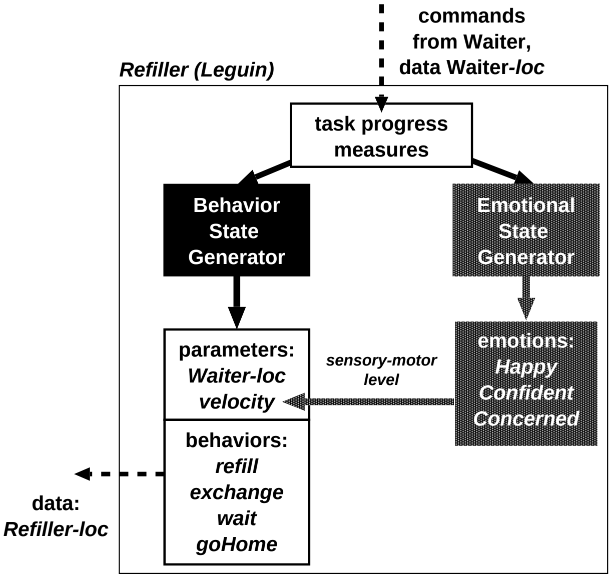


Fig. 5. Detail of the implementation of the `RefillerScript` for the Refiller robot, Leguin.

E. Refiller and the RefillerScript

Leguin, the Refiller robot, also has four behaviors under the `RefillerScript`, but only three emotions as seen in Fig. 5. The `RefillerScript` has only one source of external inputs, communications from the Waiter in the form of either commands ("wait," "refill," "hurry," "go home", "intercept") or the position data ("Waiter-loc"). It communicates only one output, the location of the Refiller in absolute coordinates after the Refiller has moved more than 3 cm from its last reported position.

The Refiller starts in the `wait` behavior where she loiters around the serving station telling jokes via pre-recorded audio sound bytes and mingling with the crowd. `face-find` runs to maintain a simple human-robot eye-contact interaction. When the Refiller receives a request for "refill" command from the Waiter, she instantiates her `refill` behavior with the parameter value of `normal speed` (60% of the maximum safe speed setting). `refill` is also a move-to-goal behavior where the location of the Waiter (`Waiter-loc` parameter, which is continually refreshed through inter-agent communication) is the value of the goal. If she receives a `hurry` command from the Waiter, she increases her navigational speed to the *maximum safe speed* or 100% of the maximum safe speed. Once within 1.5 meters of the Waiter, the `exchange` behavior is triggered. After the Waiter's command ("go home"), she returns to the serving station under normal speed with the `goHome` behavior.

V. Results

The emotional distributed control scheme was demonstrated at two venues: the annual AAAI Mobile Robot

Competition Aug 1-3, 2000, in Austin, Texas, and as part of the Educator's Open House exhibit at Museum of Science and Industry (MOSI), Sept. 10, 2000. Program trace data collected at MOSI clearly showed that emotions led to dynamic adaptations and changes in the robots behaviors, permitting the robots to continue to make progress on the resupply task where they would have otherwise been trapped in a wait state.

A. AAAI Performance

The *Hors D'oeuvres, Anyone?* event consisted of two rounds, an unscored preliminary round lasting about 30 minutes and a final, lasting 2 hours. Although the two other entries used emotional labels to describe how their robots' behaviors and interface were perceived by the audience, the USF team was the only one to use a formal model of emotions for either intra- or inter-agent activities. An error in the data logging program prevented the acquisition of quantitative data, but emotions did lead to the correct emergent behavior and several lessons where learned. The entry won the Nils Nilsson Integration Award, a general Technical Achievement Award, and third place overall.

At the preliminary event, the robots smoothly demonstrated modification of their individual behavior, changes to their behavior in response to their task progress, and overall emergent societal behavior to a small audience and local TV crews. The team members and audience set up non-productive scenarios, where the Refiller could not reach the Waiter before it was likely to run out of items. The Waiter changed its behavior to intercept the Refiller rather than wait empty-handed as had happened in 1999's event.

At the final event, the robots interacted with a larger crowd than at the Preliminary event. The emergent societal behavior occurred but was less discernible. An unforeseen task dependency was uncovered: the Refiller can't service the Waiter until she restocks. At the final event, the Waiter was often requesting a refill faster than the Refiller could navigate back to the serving station. The solution is for the Refiller to deny requests and communicate her estimated time to readiness to the Waiter, allowing the Waiter to immediately become frustrated rather than experience a short time delay as the negative emotion escalates.

B. MOSI Performance

The robots were exhibited to the public shortly after the AAAI competition at MOSI (see Fig. 6), duplicating the venue of a reception with large numbers of people interacting with the robots in unpredicatable ways as well as permitting data collection. The robots served USF pencils to over 100 museum attendees on the main floor for approximately 1 hour.

C. Overview

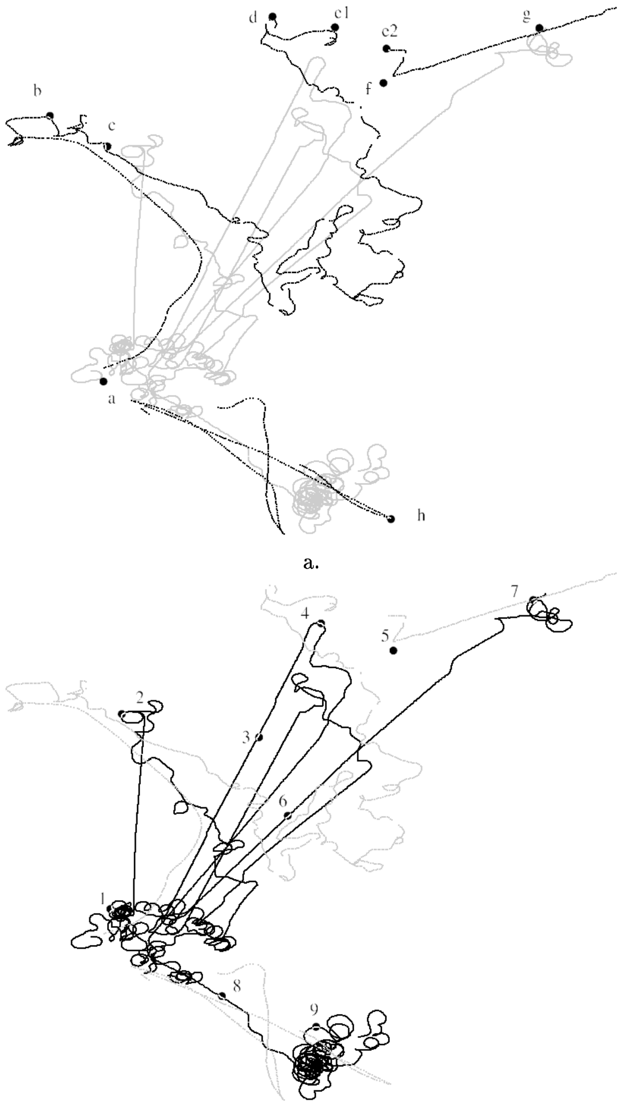
The Waiter ranged over 140 meters, and the Refiller traveled 142 meters. The waiter ran for 3504 seconds, while the refiller ran for 3454 seconds. Fig. 7 shows plots of the robots paths with important events superimposed.

Fig. 6. picture of MOSI

Figure 7 shows the paths taken by the Waiter and Refiller as they work the crowd at the Museum of Science and Industry in Tampa, Florida. The Waiter starts out in the serving area `a` and proceeds into a crowd of children and adults to hand out pencils. After running low on pencils, she calls on the Refiller for a refill, and continues to serve `b`. At `c` she rendezvouses with the Refiller, and is refilled. `d` represents another request for refill. However, the robot had to be restarted (`e1` and `e2` represent the downtime). The turn towards `f` is the Waiter deciding it must intercept the Refiller. At `f` she has been refilled, and goes back to serving. From `f` to `g` the robot transitions from to being in a dire need of refill, but she is refilled at point `g` before running too low. Finally at `h` the robot moves from serving to having to intercept the Refiller.

Figure 7b. is identical to Figure 7a., but the annotations reflect the points of interest for the Refiller. The Refiller starts at the refill station, `1`. When it receives a command for refill, it proceeds directly to the Waiter. `3`, `6`, and `8` show the Refiller as it proceeds to the Waiter. Though it is not visible from the graph, as the commands from the Waiter become more urgent, the Refiller increases its speed. Deviations in the Refiller's path are due to avoiding humans. `2`, `4`, `5`, and `7` show the Refiller reaching the Waiter, and exchanging trays. Point `5` is blank because logging of messages was not active during that time. At `9` the robot is unable to see the tray which is needed to alert the Waiter that it has received its refill. Thus the robot is stuck, waiting to see the tray, creating an unforeseen deadlock situation created by a human not presenting the tray.

From a human-robot interface perspective, analysis showed that Refiller took three times as much time to refill as to go home. This asymmetric effect had also been observed at the AAAI competition, and occurs for several reasons. When the Refiller ceases loitering by the serving station, her change in behavior attracts attention and people come to investigate. As she moves closer to the Waiter, people interacting with the Waiter notice the additional robot and begin interacting with it as well. However, when



a.

b.

Fig. 7. Paths of the Waiter and Refiller with a.) with events of interest of the Waiter noted and the Refiller path in grey and b.) with events of interest of the Refiller noted and the Waiter path in gray.

the exchange is completed, the Refiller is clearly heading away from the more interactive Waiter and attracts little interest from the crowd.

D. Representative 10 Minute Interval

Representative data of both robots' behavioral and emotional states extracted from a 10 minute period is shown in Fig. 8. The two traces are over time (X-axis) and the top of each trace shows the active behavior for the time span. The Y-axis is the emotional state. A line on the plot indicates the time and duration of a particular emotional state. Arrows in the Y direction between traces show the commands issued by the Waiter to the Refiller, via KQML; recall that as a subordinate, the Refiller did not issue commands to the Waiter. Location data messages were also exchanged, but are not shown since they did not impact the emotional or behavioral state of the robot.

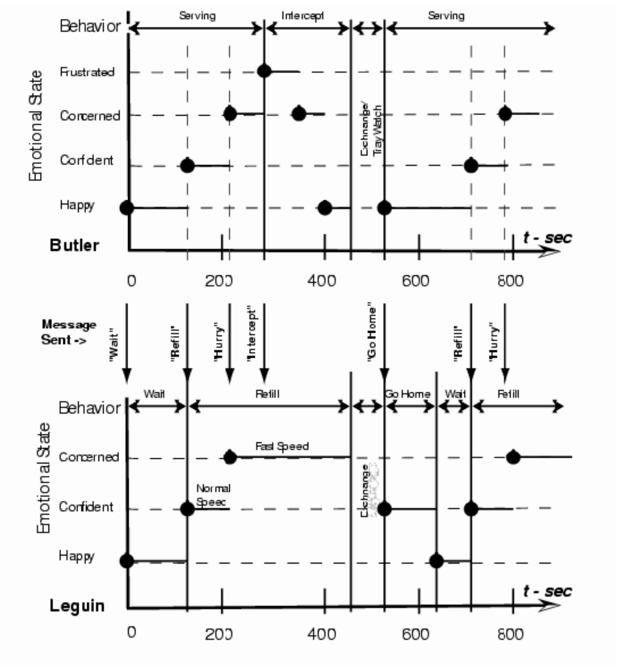


Fig. 8. Representative data run showing emotions and resultant changes in internal behavior and behavior of team member. (Butler is the name of the Waiter, Leguin is the Refiller.)

The traces show that the robots met the research objectives. First, they regulated their subgoals and motivations according to their own current internal emotional states as well as external signals. Second, they socially adapted their actions to the other agents, both human and artificial, depending on the current situational context. The use of emotion produced the desirable emergent societal behavior of avoiding deadlock-like waiting in dependent tasks as well as meeting the design criteria of being compatible with behavior-based architectures and using a distributed control scheme.

The trace shows that the intensity of the emotions was implemented as a constant, even though the performance metrics were linear. See the Refiller from t_{150} to t_{450} when `refill` is active. While CONFIDENT, the Refiller's navigational velocity is normal, then undergoes a step change to maximum when CONCERNED. This constant intensity was due to programming time constraints. A more cognitively plausible implementation is to have the intensity of the emotion change, thereby leading to proportional changes in the velocity. Future work will directly couple emotional intensity with behavioral intensity at the sensory-motor level.

There was no emotion specifically ascribed to the `exchange` behavior. `exchange` was essentially a suspend mode for the robots so there was no performance metric to serve as a input to the ESG. This could be interpreted as the robot being ambivalent when there is no metric for progress.

The robots begin after a reset with the Waiter HAPPY and issuing a "wait" command to synchronize the Refiller. The Refiller activates `wait` and begins circulating near the refill station (home). The Waiter begins serving and after aproximately t_{150}, the rate of treat consumption becomes sufficiently high. She issues a "refill" command and is CONFIDENT that her command will be achieved. Leguin receives the "refill" command which causes her to change behavior from `wait` to `refill` at a normal speed.

At about 220 seconds into the run, conditions relating to the rate of consumption of treats, or the rate of progress of Refiller, cause the Waiter to change to CONCERNED, triggering another "hurry" message to the Refiller. While the Waiter's behavior is still `serve`, the Refiller's emotional state is now CONCERNED. As a result, her speed increases to her maximum safe navigational speed. Approximately one minute later, the Waiter senses an increase in treat consumption, or insufficient progress by the Refiller, causing an emotional state change to FRUSTRATED. The Waiter at this point has calculated that the Refiller will not reach her before she runs out of treats. The ESG dictates she should abandon her current `serve` behavior and move to `intercept` the assistant to expedite the refill. The Waiter calculates the relative bearing to the Refiller, then attempts to intercept at the maximum safe speed. She sends a message to the Refiller to this effect, so the Refiller, if not already CONCERNED and moving fast, attempts to speed up.

During this intercept behavior, it is interesting to see the Waiter's emotional state moving from FRUSTRATED to CONCERNED and eventually HAPPY as she closes with the Refiller. The reduction of negative emotion is due to the TTE is more closely approaching the TTR; the relationship between TTE and TTR directly influence the emotional state as per Table III. These emotional state changes do not dissuade her from intercepting the Refiller.

At about 450 seconds into the run, the robots have effectively intercepted one another, and switch states to a stationary exchange behavior. During this time, a handler makes the physical exchange of treats from the Refiller's tray to the Waiter's tray. Also, while this process takes place, `face-finding` is de-instantiated and `tray-watch` monitor is activated. When the tray was recognized, at about 500 seconds, the Waiter's emotional state became HAPPY and she sent a message to the Refiller to "go home." This elicited the behavior of returning to the serving station, and the emotional state of CONFIDENT. In this example, the Refiller returns to the serving station to be restocked at about 600 seconds. While the Refiller has been returning home, the Waiter has been happily serving. Shortly after being restocked, the Refiller receives another message from the Waiter requesting a refill, and the process starts again.

VI. Discussion

As noted in Sec. I, the motivation for this work is to investigate cognitive models. We believe emotions are an interesting biomimetic source of insight into appropriate design principles. The use of emotions for control raises the issues of whether it can shown to be better than traditional control methods, what the advantages are in general of using emotions, how do behaviors and emotions emerge in such a system, what aspects of emotions are not currently captured, and scalability. Each of these issues is discussed below.

A. Emotions versus Traditional Control

This article does not claim that using a formal model of emotion to control of a robot is the only or best implementation. The motivation for this work has been to explicitly explore emotions as one possible method of controlling multi-robot systems. The simple implementation of the ESG raises the question of whether it have been easier just to engineer the solution. We believe the answer is "no." An emotional model provides a number of features beyond a simple state machine with timeouts, such as Kube's system [31]. While this implementation of emotions is quite similar to Kube's sytem, the larger issues are not as simple as a timeout. Behaviors change based not only on time, but also on resource expenditure, and task progress. The correct action for each robot to take when they do not know why the other is doing what it is doing is problematic. It is expected that the responses will increase with broader domains, making it harder to engineer a system with consistent design principles.

One source of resistance may be vagueness associated with emotions. The emotion-theoretic literature is divided on the number of emotions and their labels, which may give a robot designer pause. However, one reason that this is not a serious disadvantage is that even emotional theorists are less interested in labeling than in the results of emotions; the exact number and names of emotional states become less important for more fuzzy implementations.

B. Advantages of Emotions

The primary performance claim made in this article is that the use of a formal model of emotions breaks cyclic dependency problems without centralized planning if robots have alternatives (e.g., intercepting or heading directly to the refill station) and minimal communication. The increase in performance from being able to reliably use robots for interdependent tasks is another advantage. The use of a Refiller reduces the time "off-task" that would be spent by the Waiter traveling between its desired location and its refill station. Emotions caused the Refiller to dynamically increased its speed which leads to faster refills.

Another possible performance benefit is that the robots internal emotions led it to actions which favorably modified the environment. It was observed that the Refiller's increase in speed caused people to be more wary of it and get out of its way, magnifying the increase in performance from just an increase in velocity. While this claim is impossible to rigorously prove since the robots interact with large numbers of people, it is notable. It also suggests that people respond appropriately to robots which act "naturalistically" even if they are unaware of the robots emotional state. This would tend to support the motivation for the application of emotions to agents.

In addition to improved performance, the use of emotions has practical implementation advantages. First, the coding of emotions was simple. The script code for each robot was on the order of 45 lines of code, with the ESG portion consisting of less than 20 of those lines of code. In constrast, the modular behaviors comprised several thousand lines of code. Second, emotions under the multilevel process theory map directly onto structures in hybrid deliberative/reactive architectures. This means emotions can be added to these systems without any re-conceptualization of components.

Another advantage of the emotional implementation is that it is local to each robot and based on task progress. A robot does not have to understand a team member's emotional state (if any) as per [14].

C. Emergent Behaviors and Emotions

This article illustrates how the desired types of cooperation between distributed robots were made possible using emotions, as well as provided reinforcement for the multilevel process theory of emotions.

In terms of emergent behaviors, it should be noted that the robots do not need to interpret or understand each other's emotions or understand other's task. The TTR data is derived from location data; receiving accurate measurements provides a convenience for optimizing the response, but it is not required. The TTR could be estimated by the Waiter without any communication with the Refiller. This is simple, modular, and in the spirit of distributed robotics.

The various theories of emotion do not require an agent to clearly express its internal emotional state to other agents, and emotions can be invoked by the missing presence of another agent. Social expression of emotions emerge at the sensory-motor level in the multilevel process theory of emotions and so are theoretically supported by our implementation. Expressiveness and social facilitation with humans was outside of the scope of this work and will be addressed in future work.

While our system does not require one robot to understand what another robot is doing and why, there are situations where this might be useful. Such knowledge might help the assisting robot to choose among multiple requesting robots (e.g., Robot X needs the resource more than Robot Y) or methods (e.g., Robot X is operating in a covert mode, so I must be cautious and minimize communications). The existing KQML structure supports the attachment of this type of information at a later date; this is one of the reasons for the use of a general agent communication language rather than an ad hoc protocol.

One of the interesting aspects of the work reported in this article is that it simulated the sensory-motor level of emotions. The external inputs (e.g., requests) or the sensory-motor stimuli (e.g., seeing a tray full/empty) gave rise to simple reflex-like reactions involving the motor system only (e.g., increasing moving speed, stopping, etc.). Note that in some regards communication can be treated as the equivalent of direct perception stimuli.

This work also simulated the schematic level of emotions by combining the sensory-motor processes (which led to the arousal of the motor expressive behavior) with perception (perceiving the speed at which the treats disappear, perceiving the Refiller's lack of progress, etc.) to lead to slightly more complex states such as CONFIDENT and FEELING HAPPY. These states were in turn each associated with behaviors or action tendencies (see Table I) which are the most appropriate action to take given the current situation: current internal state, and current external environment.

D. Aspects of Emotions Not Currently Captured

Emotions found in the human system are the most complex ones among the animal kingdom. They serve a wide variety of functions which we list here non-exhaustively: organization of memory, learning, perception biases, categorization, self-regulatory function, motivation and performance, decision-making, and communication, maintenance of social norms between society of agents, and more.

At this time, only the sensory-motor and schematic levels of the multilevel process theory have been implemented. The conceptual level is planned for future work. It should be noted that this initial implementation leaves open the issue of how personality impacts the choice and expression of behaviors; this, like expressiveness, was deemed beyond the scope of this effort. It is conjectured that personality is not local to a script; instead it is a property of the robot as a whole.

Emotional responses were partitioned and labeled ad hoc and would profit from further inquiry. The implementation used only a rudimentary sensory-motor linkage which is the subject of future work. It largely ignored human-robot interactions and the expression of emotions for implicit communication, which will also be addressed in future work.

E. Scalability

An interesting question is whether the desired emergent societal behavior will scale if more robots or refill stations are added. In the case where the additional Waiters outnumber the Refillers, there is the possibility that more requests for the shared resource can be generated than can be handled. The issue of pre-emption, how to decide when to re-task a Refiller heading for Waiter A to now service Waiter B, becomes central. The case of more Refillers than Waiters is less severe, where multiple Refillers may attempt to service a single Waiter resulting in wasted duplication of effort. However, it can be imagined that in military or humanitarian resupply, the duplication might be desirable to ensure that at least one Refiller accomplished its task even in the presence of dangerous terrain.

The increase in the number of robots and multiple instances of resources suggests a reconsideration of Lin and Hsu's object prioritization [26] as a method of breaking dependencies (Sec. II-D). However, emotions may be more useful for a solution which minimizes communications; for example, we envision a system where the Waiter could simply broadcast its data and commands without knowing who was available. The Refillers could then use receive those messages and resulting emotions would lead to servicing by nearer robots that can be pre-empted (emotions associated with servicing the request outweigh the emotions associated with task progress on the current task).

Another aspect of scalability is how useful emotions will be when the robots have multiple tasks and roles to fulfill. Each task contributes an influence to the overall perceived emotional state of the robot, following the emergent property of reactive behaviors. This is expected to be easier to implement and be more sensitive to situations where the robot is performing a few tasks sub-optimally than traditional planning and control methods which require explicit modeling of the relative performance contributions of each task.

VII. Summary and Conclusions

This article has shown how emotions can guide emergent cooperation in teams of heterogenous multi-robot systems working on interdependent tasks. Agent resource resupply is an example of one such cooperative task. The use of a secondary, Refiller assistant maximizes the time-on-task of the primary agent, the Waiter. However, cooperation introduces a dependency between the agents, where the failure of the assistant to refill the Waiter can stop task progress by the primary. The emotional mechanism promotes the benefits of cooperation without centralized control or deliberation. As a result, emotions are well-suited for control of distributed, behavior-based systems where centralized, deliberative methods are often too computationally expensive or too immature to implement.

Although emotions play many important roles in human performance, this article has focused on their role in monitoring and maintaining progress toward specific goals. The implementation reported in this article is appropriate for behavior-based and hybrid deliberative/reactive robots. It implements a partial translation of the multilevel process theory of emotions [6]. The process levels correspond to the levels found in most hybrid deliberative/reactive architectures (*sensory-motor* or reactive behavior, *schematic* or assemblages of behaviors, and *conceptual* or deliberative).

Emotions provide the ongoing monitoring function; from that monitoring, emotional states are generated which suggest through their action tendency the most likely appropriate behavior for the given situation (see Table II). The results reported Sec. V-D illustrate how task progress was made possible or improved by one of two emotional responses. For minor negative emotion inducing situations, the robot *dynamically adapted already active behaviors*; for example, the Refiller increased its sensory-motor level of response (e.g., velocity) as its emotions became more negative (Happy to Concerned), For escalating situations, the robot *instantiated new behaviors to affect alternative ways of achieving a mission*; for example, where the Waiter experienced a schematic-level behavioral change from `serve` to `intercept`.

While other researchers have considered emotions for human-robot interactions, and intra-robot control, this work appears to be the first to use emotions for emergent intra- and inter-robot coordination specifically for multiple robots working on an interdependent tasks. The simplicity of the implementation on two fully autonomous robots interacting with humans in unstructured environments compared with the power of the results offers support that this multilevel theory is a useful model of emotions. We note that the work reported in this paper is preliminary and basic cognitive and robotics issues, such as expressiveness for social facilitation and the addition of the conceptual level, remain. These issues are the subject of current and future work and are being explored both through simulation and empirical data collection.

Acknowledgments

Portions of the work described in this paper were supported by ONR/NRL (N00173-99P1543), DARPA (F04701-99-C-0317) and a travel grant from the AAAI. The authors would like to thank the staff at MOSI, Brian Minten, Kevin Murphy, and Mark Micire for their help in data collection and the anonymous reviewers for their suggestions.

References

[1] M. J. Mataric, M. Nilsson, and K. Simsarian, “Cooperative multi-robot box-pushing,” in *IROS-95*, Pittsburgh, PA, 1995, pp. 556–561.

[2] L.E.Parker, *Alliance: An architecture for fault tolerant multi-robot cooperation*, Technical Memorandum 12920, Oak Ridge National Laboratory, 1995.

[3] L.E.Parker, “Adaptive heterogeneous multi-robot teams,” *Neurocomputing*, vol. 28, pp. 75–92, 1999.

[4] R. Zajonc, “Feeling and thinking: Preferences ned no inferences,” *American Psychologist*, , no. 35, pp. 151–175, 1980.

[5] B. Zajonc and H. Markus, “Affect and cognition: The hard interface,” *Cognitive Science*, , no. 16, pp. 539–581, 1984.

[6] H. Leventhal and K. Scherer, “The relationship of emotion to cognition: A functional approach to a semantic controversy,” *Cognition and Emotion*, vol. 1, no. 1, pp. 3–28, 1987.

[7] R. R. Murphy, *Introduction to AI Robotics*, MIT Press, 2000.

[8] L. Parker, “Alliance: An architecture for fault tolerant multi-robot cooperation,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[9] M. Schneider-Fontan and M. Mataric, “Territorial multi-robot task division,” *IEEE Transactions on Robotics and Automation*, vol. 14(5), oct 1998.

[10] R. A. Brooks, “A robust layered control system for a mobile robot,” *IEEE Journal of Robotics and Automation*, vol. RA-1, no. 1, pp. 1–10, 1986.

[11] M. El-Nasr, T. Ioerger, and J. Yen, “Peteei: A pet with evolving emotinal intelligence,” in *Autonomous Agents 99*, 1999, pp. 9–15.

[12] J. Velasquez, ,” *Cathexis, A Computational Model for the Generation of Emotions and their Influence in the Behavior of Autonomous Agents, Master Thesis, MIT*, 1996.

[13] J. Velasquez, “From affect programs to higher cognitive emotions: An emotion-based control approach,” in *EBAA 99*, 1999.

[14] C. Beazeal and J. Velasquez, “Toward teaching a robot “infant” using emotive communication acts,” in *Socially Situated Intelligence Workshop, SAB*, 1998.

[15] C. Breazeal and B. Scassellati, “Infant-like social interactions between a robot and a human,” *Adaptive Behavior*, vol. Winter, 2000.

[16] F. Michaud, P. Prijanian, J. Audet, and D. Letourneau, “Artificial emotions and social robotics,” in *Proceedings of the Distributed Autonomous Robotic Systems*, 2000.

[17] F. Michaud, P. Prijanian, J. Audet, D. Letourneau, L. Lussier, C. Theberge-Turmel, and S. Caron, “Experiences with an autonomous robot attending aaai,” *IEEE Intelligent Systems*, vol. September-October, 2001.

[18] R. Pfiefer, *Cognitive Perspectives on Emotion and Motivation*, chapter Artificial Intelligence Models of Emotion, pp. 287–320, Dordrecht, Netherlands: Kluwer Academic Publishers, 1988.

[19] F. Paulhan, *The Laws of Feeling*, London: Kegan-Paul, 1887, English Trans. 1930.

[20] H. Simon, "Motivational and emotional controls of cognition," *Psychology Review*, vol. 1, pp. 29–39, 1967.

[21] A. Sloman and M. Croucher, "Why robots will have emotions," in *Proceedings of the Seventh IJCAI, Vancouver, B.C.* 1981, pp. 197–202, Morgan-Kaufmann.

[22] A. Swagerman, "The artificial concern realization system acres: A computer model that has emotions and can name them," Unpublished manuscript, University of Amsterdam, Holland, 1986.

[23] A. Ortony, G. Clore, and A. Collins, *The Cognitive Structure of Emotions*, Cambridge University Press, 1988.

[24] M. Mataric, "Issues and approaches in the design of collective autonomous agents.," in *Robotics and Autonomous Systems*, December 1995, pp. 321–331.

[25] D. Goldberg and M. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," in *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, Menlo Park, July 27–31 1997, pp. 637–642, AAAI Press.

[26] F. Lin and J. Hsu, "Cooperation and deadlock-handling for an object-sorting task in a multi-agent robotic system," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2580–85, 1995.

[27] V. Hartonas-Garmhausen, E. M. Clarke, and S. Campos, "Deadlock prevention in flexible manufacturing systems using symbolic model checking," in *ICRA '96*, 1996, vol. 1, pp. 527–532.

[28] S. Qutub, R. Alami, and F. Ingrand, "How to solve deadlock situations within the plan-merging paradigm for multi-robot cooperation," in *IROS '97*, 1997.

[29] P. Svestka and M. Overmars, "Coordinated path planning for multiple robots," in *Robotics and Autonomous System*, 1998, pp. 125–152.

[30] T. Fukuda, IH. shihara, and HN. iraoka, "Deadlock resolution in distributed autonomous robotic system with hand-to-hand motion," in *IEEE ICRA*, 1996, vol. 3, pp. 2049–2054.

[31] C. Kube and H. Zhang, "Stagnation recovery behaviours for collective robotics," in *1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 1994, vol. 3.

[32] F. Hara, K. Sasajima, and T. Sawada, "Morpho-functional characteristics of ssa-based two-mobile-robots system carrying a long bar on wavy road," in *IEEE Intelligent Robots and Systems*, 1998, vol. 3, pp. 1411–1418.

[33] R.C. Arkin, "Homeostatic control for a mobile robot: Dynamic replanning in hazardous environments," *Journal of Robotic Systems*, vol. 9, pp. 197–214, 1992.

[34] C. Lisetti, "Motives for intelligent agents: Computational scripts for emotion concepts," in *Sixth Scandinavian Conference on Artificial Intelligence*, 1997.

[35] R.C. Schank and C.K. Riesbeck, *Inside Computer Understanding: Five Programs Plus Miniatures*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

[36] S.P. Engelson and D.V. McDermott, "Passive robot map building with exploration scripts," in *Technical Report: YALEU/DCS/TR-898, Yale University, Dept. of Computer Science*, 1992.

[37] W. Lim and J. Eilbert, "Plan-behavior interaction in autonomous navigation," in *SPIE Mobile Robots V*, 1990, pp. 464–475.

[38] R.R. Murphy, "Use of scripts for coordinating perception and action," in *IROS-96*, 1996, pp. 156–161.

[39] P. Rowe and A. Stentz, "Parameterized scripts for motion planning," in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems*, 1997, pp. 1119–24.

[40] R. C. Arkin, *Behavior-Based Robotics*, MIT Press, 1998.

[41] R. Murphy and A. Mali, "Lessons learned in integrating sensing into autonomous mobile robot architectures," *Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Hardware Agents*, vol. 9, no. 2, pp. 191–220, 1997.

[42] N. H. Frijda, *The Emotions*, New York: Cambridge University Press, 1986, 8.