Artificial Intelligence

Artificial Intelligence 110 (1999) 293–320

www.elsevier.com/locate/artint

Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams

Barry Brian Werger [1]

Ullanta Performance Robotics and Interaction Lab., Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0781, USA

Received 1 February 1998; received in revised form 1 November 1998

Abstract

While terminology and some concepts of behavior-based robotics have become widespread, the central ideas are often lost as researchers try to scale behavior to higher levels of complexity. “Hybrid systems” with model-based strategies that plan in terms of behaviors rather than simple actions have become common for higher-level behavior. We claim that a strict behavior-based approach can scale to higher levels of complexity than many robotics researchers assume, and that the resulting systems are in many cases more efficient and robust than those that rely on “classical AI” deliberative approaches. Our focus is on systems of cooperative autonomous robots in dynamic environments. We will discuss both claims that deliberation and explicit communication are necessary to cooperation and systems that cooperate only through environmental interaction. In this context we introduce three design principles for complex cooperative behavior—*minimalism, statelessness* and *tolerance*—and present a RoboCup soccer system that matches the sophistication of many deliberative soccer systems while exceeding their robustness, through the use of strict behavior-based techniques with no explicit communication. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Multi-robot teams; Scalability of behavior-based systems; Reactive control; Hybrid systems; Mobile robot cooperation; Minimalist robotics

1. Introduction

Since the publication of Brooks’ *Subsumption Architecture* [11] and its later extension to *behavior-based* systems [13,14], there has been an “explosion” of behavior-oriented AI research [41]. Most current autonomous robotic systems are described with some

[1] Email: barry@usc.edu; http: //www-robotics.usc.edu/∼barry/ullanta.

PII: S0004-3702(99)00023-5

reference to “behaviors”, and as being at least partially “reactive”, “subsumption-based”, or “behavior-based”. Yet while the terminology and some basic concepts have become widespread, some of Brooks’ central ideas are often lost as researchers try to scale behavior to higher levels of complexity. “Hybrid systems” with classical deliberative, model-based strategies [29] that plan in terms of behaviors rather than simple actions have become very common for higher-level behavior. This separation of the system into *planning* and *execution* components is a return to the very *functional decomposition* [14] that behavior-based approach hopes to avoid. The high-level deliberative components violate many other principles of behavior-based design [13,49]. As we will discuss, these deviations from a strict behavior-based approach often lead to systems that are far more complicated and prone to failure than they need to be.

We do not claim that there is no place for deliberation, world models, or symbolic computation in robotic systems. We do claim that the strict behavior-based approach can scale to higher levels of complexity than many robotics researchers assume, and that both behavior-oriented development and the resulting systems are in many cases more efficient and robust than those that rely on “classical AI” deliberative approaches.

Our focus in this paper is on systems of cooperative autonomous robots in dynamic environments. We will discuss research that claims that certain types of deliberation and explicit communication are necessary to cooperative behavior [7,8,22,24,32,42,44,45,47], as well as systems that cooperate only through environmental interaction [9,17–19,25,27–29,32,53] but do not function at quite as high a level. In this context we will then present our design principles for complex cooperative behavior—*minimalism, statelessness* and *tolerance*—and present a system that meets or exceeds the design goals of many deliberative systems, through the use of strict behavior-based techniques with no explicit communication.

1.1. A story, revisited

(continuing a story from Brooks’ AIJ paper of 1991 [14]. . .)

> Our 1890’s Artificial Flight researchers, now specialized in different areas, are again miraculously transported onto a modern Boeing 747 in flight. While the vast majority argue the benefits of their own specialty items developed since their first visit (such as reclining mechanisms of the solid steel chairs and design of food-service carts), one, a Dr. Streams, decides that starting out to build such a large complex airplane is futile. Furthermore, he reasons that even identification of key components that must be combined into such a complex airplane is beyond the state of the art and also futile. He sits next to a boy who is playing with a piece of paper, and watches as the boy tosses the folded sheet towards the aisle. It glides gracefully through the air. When the boy asks to be let out to find his “plane”, Dr. Streams stands up with quite an enlightened look on his face.

Back in the 1890s, Dr. Streams’ work with simple paper airplanes shows remarkable success—realizing that it is principally the interaction of the wing surfaces with the air that leads to flight, he develops techniques for manipulating wing shape (flaps, ailerons, etc.) that allow the plane to be controlled. His ideas for basic control of flight become ubiquitous in the literature. Though he and his students continue focusing on the interaction of the plane with the air, and thus move towards airplanes that can carry people, most other researchers become preoccupied with outfitting the paper planes with reclining seats and food-service carts. Though miniature versions of their systems are occasionally at least flying (as they hadn't been before Streams' work), they feel that Streams' approach must not be powerful enough because their specialty items don't integrate very well. This becomes especially apparent in the field of Cooperative Airplane Teams: pilots need to communicate their positions and intentions to each other while in flight, and the only two means of remote communication are the telegraph and smoke signals. The fires for the smoke signals tend to burn the paper planes (not to mention that the signals can't be seen at night); the telegraph wires not only weigh more than the planes but cause planes to pull on each other or get tangled, and crash.

2. Behavior-based design principles

We have developed a wide variety of multi-robot systems that have had to live up to the strict requirements of public performance, competition, and interaction with human beings. These systems have been successful in such diverse domains as laboratory research, robotic theater [50], performance art, robotic sales exhibitions, AAAI mobile robot contests [1,52], and RoboCup robotic soccer competition [48,49]. Much of the success of our ventures can be attributed to three simple design principles which guide our behavior-based systems development: *minimalism*, *statelessness*, and *tolerance*. In this section we present these principles and a brief discussion of our behavior-based approach to system design.

Autonomous robots (especially small, highly mobile ones) must make their way in the real world with severely limited sensing capability, uncertainty of perception, imprecision of action, and relatively low-powered on-board processors (or the communication problems that arise when using off-board processors). While it is often tempting to look to more powerful hardware, more complicated programs, more precise sensors, and more communication in the attempt to generate more sophisticated robotic behavior, we feel that the key to success in such an endeavor lies rather in the philosophy of trying to fully exploit minimal systems [18,49,53]. We believe that control software has not yet come close to fully exploiting the power of even the simplest of mobile robots, and that the proper direction to investigate for great increases in robot capability is efficient use of simple systems rather than the type of hardware/software inflation cycle seen in the desktop computer industry. Moreover, experience has shown that in many cases more precise, "better" robots can be more difficult, rather than easier, to control [40].

2.1. Behavior-based design

As we have mentioned, "behavior-based" has become a very popular term, but one without exact definition. Matarić [29] gives an overview of common conceptions of the behavior-based approach. Brooks [13] describes a set of four key concepts and their key ideas that lead to behavior-based robotics: *situatedness*—the world as its own best model, *embodiment*—the world grounds regress, *intelligence*—intelligence as determined by the dynamics of interaction with the world, and *emergence*—intelligence in the eye of the beholder. Behavior-based systems thus are structured in terms of observable activity that they produce, rather than traditional functional decompositions [14]. The activity producing components, *behaviors*, compete for actuator resources and share perceptions of the world rather than any centralized representation. Behaviors tend to be simple, so that computational "depth" [13]—the computational path from sensor to actuator—is minimized to maintain a high degree of interactivity with the environment. Behavior-based systems are highly parallel so that capability—new behaviors—can be added as increased computational "breadth". Behaviors are "layered" [11] in such a way that capability is incrementally added to a functional system, leading to a design process that goes not from isolated components to a final system which integrates them into something meaningful, but from simple yet complete behavior to more complex complete behavior [14,16,28].

The design of behavior-based systems is often referred to as a "bottom up" process [15, 41,51], but this refers not so much to determination of the structure of the system as to a basis in physical sensing and action, and incremental development of sophistication from simple to complex. The system structure undergoes drastic changes driven by top-down task constraints as well as bottom-up sensorimotor constraints until a set of basis behaviors is determined [27]; it is only with this solid foundation that the design process becomes one mainly of synthesis.

Brooks [13], Matarić [27], Parker [33], Steels [41] and Werger [52] discuss techniques for multi-agent, behavior-based system design. We present in this paper a thorough description of a complete system, the "Spirit of Bolivia" soccer team; in [49], we present a long example of the application of our principles to the design of this team, including the many early drastic transformations it went through that are not presented here.

2.2. Minimalism

Donald et al. [18] introduce *minimalism*, the pursuit of the minimal configuration of resources for performance of a task, as an approach to the design of multi-agent systems. This is theoretically interesting because it can prove that certain resources are inessential to the information structure of the task (Donald [20] discusses this in greater detail). The benefits of minimalism, including easier and faster development and debugging and a more efficient and robust execution system, have led to recent popularity of minimalist systems; Donald et al. [18] give an overview which includes walking and running machines without static stability, dextrous manipulation without sensing, walkers without sensors or actuators, and behavior-based control systems. They also present the concept of *supermodularity*, or relocatable modularity, which partially orders the ability of systems to function when physically embedded in different ways. For multi-robot systems, this functions as a measure of simplicity, ease of reuse, and fault tolerance, and provides for certain performance guarantees. Analyzing the supermodularity of a system allows us some insight into the emergent properties [13,41] of such systems. Donald et al. [19] present the beginnings of a methodology for minimizing parallel manipulation protocols through insights gained from this and other analyses based on their *information invariants* [20] approach to multi-robot system design.

Minimalism manifests itself in our design process in numerous ways:

The use of rapid feedback rather than precise computation allows very simple calculations (e.g., scaling or differencing) or comparisons to generate complex patterns of behavior, such as the trajectories described in Section 4.3.1. This goes hand in hand with what is called *situatedness* [13] or *externalization* [20]—the use of the environment as its own best model. A significant amount of our development time is spent trying to externalize as much information as possible so as to make the system more environmentally-driven rather than model-driven. Being so dependent on the environment *reduces* dependence on a specific environment, increasing supermodularity: there is less chance for mismatch between internal representation and "reality". This maintenance of physical grounding includes the avoidance, when possible, of ethereal communications such as radio in favor of *stigmergy* [9] (see Section 5).

Behaviors are homogenized whenever possible. In most cases, it is easier to design and maintain a set of homogeneous robotic control systems than a set of heterogeneous ones. Balch [6] has presented work in investigation of agents' behavioral coding, as well as hardware, to analyze *social entropy*, or amount of heterogeneity in a multi-robot system; we complement this somewhat with the concept of *minimal heterogeneity*, which we define as describing a system in which all agents are homogeneous (both in hardware and software) except for a small number of parameter changes. In systems that are highly reactive to the environment, minimal heterogeneity can have very useful and predictable emergent effects (such as in the *Patrol* behavior described in Section 4.2) which lead to usefully differentiated behavior of members of a team. In dynamic situations, such as soccer play, even strict homogeneity can lead to differentiated and cooperative activity (as discussed in Section 5.1).

Actions are de-coupled and parallelized whenever possible. Separating, for example, rotational control from velocity control—even though it is counter-intuitive on a differentially-steered robot—not only allows greater reliability and re-use of code, but allows more opportunity to design emergent behaviors, such as the obstacle avoidance described in Section 4.2 and the "batting" of stuck balls described in Section 4.3.5. The resulting simpler behaviors are easier to implement and tune, and are more likely to be re-usable.

2.3. Statelessness

We define *statelessness* as a partial ordering on the amount of reactivity of a system; it is a measure of the amount of internal state maintained. A purely reactive system is of course purely stateless, whereas a classical model-based planning system has very low statelessness. The problem with internal state is that it must usually be kept synchronized with the environment, at great expense and with grave consequences of failure; the diametrically opposed problem of reactive systems is that they can guarantee no continuity of behavior, leading to frequent problems of oscillations due to local minima and temporary losses of perception. The interesting degree of statelessness for us is what we define as *perceptual decay*: the persistence of some perceptual data for a brief period of time after it is perceived, analogous to a retinal afterimage. As we will discuss in Section 4, much research has been done on the power added by even a single bit of state to a reactive system; perceptual decay allows us to take great advantage of this without changing our "stateless" code or approach; behaviors are all purely reactive, but some of their inputs fade over time, rather than disappearing suddenly. A slow decay of visual perception of the ball is the only state in our sophisticated soccer players, and is not distinguished from a direct visual perception by any part of the system. Thus we refer to the system as "stateless" (or, to be very precise, "highly stateless") even though it is not strictly reactive. This allows us to ground the entire system in sensing and action, that is, use only the world as its own model, yet address problems of hidden perceptual state and perceptual noise.

An extremely important facet of the stateless approach is the avoidance of sequenced behavior. Systems grow complex and difficult to control when there are conflicts between internal timing and environmental perception. Tambe [44] provides nine examples of types of breakdowns of teamwork in his systems; five of them are due to sequenced behavior. If it is at all possible to use environmental cues instead of sequenced behavior (and we believe that in the vast majority of cases it is), the sequences should be avoided. (Section 4.3.6 demonstrates the type of creative inspiration one can receive through a concerted effort to remove such sequences and other non-stateless control elements.)

A last recommendation for increased statelessness is to exploit physics [13] of the robot and environment. This can be "real" physics—it is often possible to develop or modify a physical device to eliminate the need for difficult computation (such as the raising of the level of the sonars to avoid sensing the ball described in Section 4.1)—or a "classical mechanics" approach to control [24]. Many successes in control of groups of robots have been due to implementation of forces of attraction and repulsion between robots [25,28,41,48]. These "physics-based" approaches are fully distributed and require no communication; they are thus arbitrarily scalable, extremely simple to implement, and robust to environmental changes and robot failures. Our team behavior in RoboCup-97 (see Section 5.1) is based entirely on such a "physics-based" approach.

2.4. Tolerance

We introduce *tolerance* as a bias towards living with uncertainty rather than trying to eliminate it. This is done by attempting to reduce a need for precision before trying to gain precision. Tolerance is addressed by most of the techniques for minimalism and statelessness above, especially the replacement of computation with rapid feedback, but also by careful thought about the minimal information necessary for achievement of a task. For example, the imprecision of dead-reckoning on a carpet led us to realize that effective ball-manipulation behavior can be performed with only visual perception of the ball and a 180-degree (roughly east vs. roughly west) distinction of the robot's heading, which allows for a great deal of uncertainty in the perceived heading (see Section 4.3.7).

2.5. Basis behaviors

Basis behaviors [27] are a set of minimal behaviors that are sufficient to be combined into solutions to a class of tasks. Matarić's [28] research on group behavior showed how various complex, biologically-inspired group behaviors could be composed from a set of general basis behaviors for spatial tasks, through two operators, *summation* of outputs and *switching* of outputs. *Flocking*, for example, is achieved by the summation of *homing*, *dispersion, aggregation,* and *safe-wandering,* while *foraging* results from switching (based on sensory conditions) between *safe-wandering, dispersion, homing,* and *following*.

Development of basis behaviors is somewhat analogous to the selection of representations in symbol-processing systems: the choice of basis behaviors has great influence on the efficiency, and even tractability, of both the development process and the final system. Effort expended in refining basis behavior choices is usually paid back many times over; it is all too easy to reach (and sometimes difficult to detect) a state where a good percentage of a system's code is dedicated to working around earlier implementation choices. We emphasize in Section 2.6 the importance of maintaining the design of a system at the level of physically grounded [15] behavior, always with an eye to perfecting the set of basis behaviors, until final coding, in order to avoid these pitfalls. This facilitates analysis of information flow through the system and allows for the necessary major restructurings with little cost.

2.6. Designing without code

One of the most important factors in minimalist, "stateless", tolerant design is the maintenance of the design at the abstract behavior level (grounded in perception and action) until final coding. We generally spend at least ninety percent of development time without any written code, and are then able to code and debug systems reliably in a matter of hours. The sweeping changes often needed to minimize a system make early coding prohibitive, while the reductions in system complexity make the coding effort far easier at the end of the process. The number of parameters that must be adjusted, the number of behaviors, and the data flow between them is usually drastically reduced between the initial and final designs, which would require re-writing of any existing code. If attention is paid to supermodularity, it is possible to design complex systems of cooperating robots without testing, and have a "first coding" that performs well with minimal tuning. By designing behavior through diagrams such as Fig. 8, information flow through the system can be clearly seen and optimizations become obvious. The system should be rearranged until a good set of *basis behaviors* (see Section 2.5) is determined. Once we have the basis behaviors, design and coding proceeds with a sequential "layered approach" [11].

Our implementation example for later discussion, the "Spirit of Bolivia" RoboCup soccer team [48], displayed very complex and robust behavior at the 1997 competition, including flexible team cooperation, but was designed in a short period during which the team members had no access to robots or computers. Because the abstract design process led to a very small and simple final control system, we were able to code, debug, and tune the system in the twelve hours between arrival at the competition and our first match.

3. Example systems and domains

Throughout the rest of our discussion, we will make extensive reference to an example soccer system, and occasional references to others. We present here some basic information to contextualize these references.

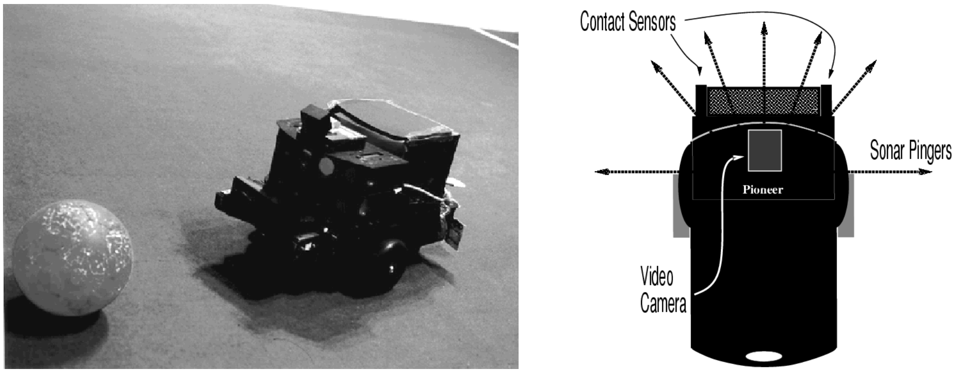

Fig. 1. The Pioneers are differentially steered bases with seven sonar pingers, grippers with contact sensors on the tips, and a Cognachrome vision system.

3.1. Our robots

The robots used in all of our activities are Pioneers sold by ActivMedia, Inc. (see Fig. 1). They feature an $18'' \times 14''$ differentially steered base, five forward- and two side-facing sonars, and 2-DOF grippers with contact and breakbeam sensors. They are equipped with the Fast Track Vision System from Newton Laboratories [36], an on-board processor based on a 16-MHz 68332 microcontroller, which extracts color-blob information from a video source at a frame rate of 60 Hz. It can be trained to recognize three distinct colors at a time, and outputs the size, visual field location, bounding box, and other data for multiple blobs of each color. It also provides various types of line-tracking data. The robots use wide angle (60-degree) non-actuated cameras. We develop our systems with MARS/L from IS Robotics [12], an on-board Motorola 68332 microcontroller programmed in a Common LISP extended for behavior-based control, and often port them to ActivMedia's C-based PAI (Pioneer Application Interface) environment when we borrow robots not equipped with the MARS/L system.

The Pioneer has a low level controller that runs the Pioneer Server Operating System from ROM. PSOS provides sensor readings periodically, and accepts commands which set rotational velocity, translational velocity, and individual wheel speeds, as well as other motor- and sensor-control functions.

Both of the Pioneer environments we work with—MARS/L and PAI—provide easy access to both sensor readings and motor control. MARS/L for Pioneer provides standard behaviors that can be connected appropriately, and PAI provides functions. In the figures illustrating behaviors, and in discussion, it should be assumed that these behaviors or functions are responsible for maintenance of assigned velocities (ultimately through PID control).

3.2. RoboCup robotic soccer

Our primary example system is the "Spirit of Bolivia", our entry in the RoboCup-97 world robotic soccer championship in Nagoya, Japan. We discuss only the team's four “fielders” only; our goalie, whose behavior was much simpler though very effective and guided by the same principles, is described in [48,49].

The domain of robot soccer has been characterized by friendly and hostile agents, inter-agent cooperation, real-time interaction, and a dynamic, uncertain environment [35]. A notion of sportsmanship (and the rules of many competitions) add the characteristic of non-damaging interaction between players. We define a “comprehensive” set of team behaviors as one that addresses all of these characteristics; a minimally comprehensive set of behaviors will cause members of a team to, at the very least, progress towards the goal and obstruct progress of opponents by interacting constructively with teammates (and safely with all agents) in a physical real-time soccer competition.

RoboCup [23] has become a popular domain for multi-agent systems research. It is a soccer competition that has leagues for small and mid-size physical robots and for simulated players. The environment is very dynamic and the task is fairly complex (and will grow in complexity as teams progress to more sophisticated competitive behavior).

The essential difference between the small and mid-sized robot leagues is the use by the small league of a global vision system: the players have access to an overhead view of the entire field. The mid-size robot league and simulator league have agents with only local perception; that of the simulator league is of course much more reliable. The simulator league and small-size league tend to use powerful workstation-class processors and have access to fairly reliable communication; the mid-size league is split between powerful off-board processors with communication problems [43,54] and less-powerful on-board processors. As a result of this, cooperation has proven rarer in the mid-size league than in the other leagues; we compare “The Spirit of Bolivia” to teams in other leagues that have the potential for more sophisticated teamwork, since it was the only mid-size team to display planned-for, interactive team cooperation in Nagoya (the co-champion Osaka team [43] has done very interesting work on learning of cooperative behavior [43,46], but due to time constraints did not integrate it into their 1997 competition team [5]; the co-champion USC team assigned roles to team members that did not change in response to field situation or teammates’ activities [38]; the Uttori team has done extensive work on flexible cooperation which was not reflected in the team’s performance at RoboCup due to communication failure; and the RMIT team concentrated on development of novel hardware, and as a result seemed to have an underdeveloped software control system at the competition [5]). Our team behavior was far from flawless (as discussed in Section 5.1), but it did produce coordinated and constructive team interaction on various occasions.

The “Spirit of Bolivia” was outstanding in the middle-size league in several other ways as well: it was the only team that displayed effective obstacle avoidance (the Osaka team implemented avoidance, but due to vision problems it was only useful when competing against the RMIT team [43], the USC team did not display or describe collision avoidance, and the other two teams were too beset by technical difficulties to demonstrate any); it demonstrated three types of effective ball manipulation [31] (as described in Section 4.3) rather than just the single “pushing” technique used by other teams, and the robots maintained an awareness of the ball that allowed them to interact with it in very smooth trajectories, with minimal time spent searching, even if it left their field of view (due to the perceptual decay described in Section 4.3.3). These traits combined to allow the “Spirit of Bolivia” an extra functionality: the robots could be let loose outside of the field to play with conference attendees, and be trusted to stay “on top of the ball” while interacting quite safely with the humans.

It is very difficult to discuss the performance of RoboCup teams objectively; the only clear measure for comparison is how well they compete (and even then, an evolving set of rules and variations of their application complicate matters [30]). We present here competition results from the middle-size league as demonstration that our approach was able, at least, to produce a highly competitive team.

The competition was not a true round-robin, and our team did not have a chance to play an *official* game against either of the teams that were named co-champions. Our two official games were scoreless ties. When one of the co-champion teams was unable to continue on to a tie-breaking game, an exhibition game was played between our third-place “Spirit of Bolivia” and the other co-champion. Our team won this game [43]. As the co-champions were unable to win against each other in various games, by winning this unofficial match our team was the only one to have defeated either of the co-champions. “The Spirit of Bolivia” was also the only team against which no goals had been scored during the entire competition.

3.3. Formations

Various examples of team mobility in formations will be presented for comparison with the flexible formations of our soccer team. These are carried out in simulation [7,21,33,34] and on real robots, both in laboratories and outdoors [7]. These systems move in more constrained configurations than those that result from flocking [28] and similar “physics-based” types of group coordination, but we are able to make comparisons to some our soccer team behaviors which combine flocking with ball-tracking and perceptual decay.

3.4. Service robotics

Our service robotics systems (hors-d’oeuvres servers [52] and sales exhibitions) must interact very safely with humans while exhibiting both high mobility and task-achieving behaviors in very crowded and dynamic environments. These systems will also be compared to formation systems such as [7] and other communication- and model-dependent systems such as those described in [44].

4. Individual competence in robotic soccer

Individual players of robotic soccer teams must display certain skills in ball-searching, ball-approaching, ball-manipulation, and obstacle avoidance [4,35,44,47]. The common mid-size league approach to ball-manipulation at RoboCup-97 seemed to be rotation in place to search for the ball, fairly linear motion towards the ball, when it was seen, and an occasional last-minute detour past the ball when it was visually apparent that the robot was heading directly towards its own goal, followed by a new search (rotation in place) for the ball. Even though teams were attempting to maintain a model of the locations of objects in the environment [39], there was in general no recovery when the robots lost visual contact with the ball.

In the small-size league, Veloso et al. [47] describe generation of complex trajectories for ball handling, using global positions of objects, prediction of ball motion, and setting of intermediate targets both for navigating around the ball and to line up for an advance.

The "RoboCup Challenge" description [4] claims that a pan-tilt active vision system (and the accompanying complicated control and spatial memory organization for lost objects) or omnidirectional lens is necessary to keep track of the ball and other field items. It also states that a classical model-plan-execution cycle is not feasible due to the dynamic nature of the competition, and a lookup-table mapping perception to action, even if precise knowledge of the environment needed to build such a table could somehow be gained, would be sufficient for very simple behavior but difficult to scale. Prediction of ball speed and direction is said to be the key to catching passes and intercepting, as well as goal-keeping.

Our team members follow trajectories very similar to those generated by the global knowledge and precise calculation of the small league teams such as [47] (see Fig. 4). They manage to chase down, intercept, dribble, and smoothly circle the ball in order to line up properly for an advance, while effectively avoiding collisions, without any calculation of ball motion or modeling of world state—that is, in a completely "stateless" manner.

The bottom level of our robots' behavioral structure is *Patrol*. This is a small extension of Brooks' *avoidance* or Matarić's *safe-wandering*. Since our design de-couples control of motion into rotational and translational components, *Patrol* itself does not generate interesting behavior. The addition of *Safety*, which complements all other behaviors, makes it a complete "creature" [14].

4.1. Safety—de-coupled motor control

The translational component of all motion control, *Safety*, receives input from the five forward-facing sonars and outputs a velocity proportional to the distance to the closest sonar-perceived object. By itself, this causes the robot to maintain the maximum safe speed to avoid hitting anything. The only physical modification we made to the stock Pioneer robots for RoboCup competition was the raising of the sonar pingers to a level (about three inches higher than normal) at which they would not perceive the ball, but would perceive other robots and walls. This is a minimizing exploitation of physics: though seemingly obvious in hindsight, there was a period before choosing this modification during which we didn't consider changing the morphology of the robots, and planned a sonar-visual interaction to distinguish the ball from other objects. This would have been far less reliable, and would have ruined the simplicity of our de-coupled velocity control. The *Safety* behavior runs in parallel with all other behavior and has no competition for velocity control.

4.2. Patrol—bottom level behavior

The *Patrol* behavior basically rotates either clockwise or counterclockwise. As implemented for RoboCup-97, it held a fixed rotational velocity, which caused it to rotate in a circle in the absence of obstacles, and a somewhat distorted circle in the presence of obstacles, due to the interaction with *Safety*. When approaching a wall, for example, the

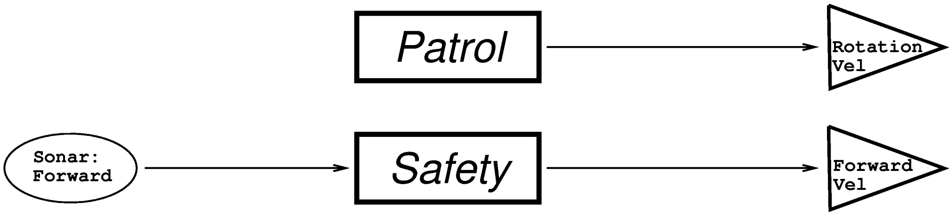

Fig. 2. The lowest level behaviors of our soccer players. *Patrol* (which outputs a fixed rotational velocity) and *Safety* (which outputs a forward velocity proportional to the distance to the closest sonar-detected obstacle) combine to produce a robust, obstacle-avoiding navigation behavior.

speed of the robot decreases with the distance to the wall, while the rotation continues at its normal rate, until the robot is again headed in a direction without obstacles. An interesting emergent property of this simple *Patrol* behavior results from the application of minimal heterogeneity—difference in direction of rotation. If two or more robots in an enclosed space (such as the RoboCup field) *Patrol* in with different directions of rotation, they tend to patrol separate territories due to the *Safety*/*Patrol* interaction.

The implementation of *Patrol* we have developed since RoboCup-97 adds two parameters for additional exploitation of minimal heterogeneity. When a player perceives its own goal within a certain range (specified by one parameter), the rotational velocity is increased; the same happens within a range of the opponent's goal, specified by the second parameter. The effect of this is specification of territories to be patrolled; when the first parameter is large and the second is small, the player will *Patrol* in an offensive position on the opponent's side of the field; in the reverse case, the player will remain near home. Range to the goal is determined simply by vertical position of the top of the goal in visual coordinates.

The basis behavior *Safety* by itself moves straight forward until something gets in its way; *Patrol* by itself rotates in place. From their parallel execution emerges a robust behavior of navigating around the environment and avoiding obstacles. Fig. 2 diagrams this current behavior system.

4.3. Ball manipulation

Our players manipulate the ball in three ways: dribbling, kicking, and batting; always towards the opponent's goal. Dribbling entails moving the ball through fairly continuous contact with the front of the robot, kicking is propulsion of the ball away from the robot at high speed (by swinging the rear around rapidly), and batting (with the side of the gripper) is used when the ball is resting against the wall or "held" by an opponent. The robots follow smooth, complex trajectories to properly align with the ball for forward progress.

4.3.1. Ball-manipulation basis behaviors

The basic intuition behind our ball manipulation is that the robot should aim to interpolate itself between the ball and its own goal, and move towards the opponent's goal. This behavior suffices for both offense and defense. It is, however, too complicated to be considered a basis behavior and implemented monolithically. We decompose this behavior

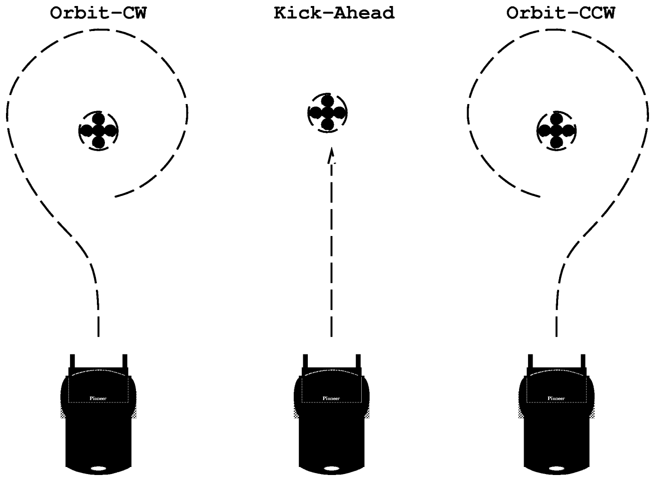

Fig. 3. The three basis behaviors sufficient to generate ball-manipulation and trajectories for soccer play.

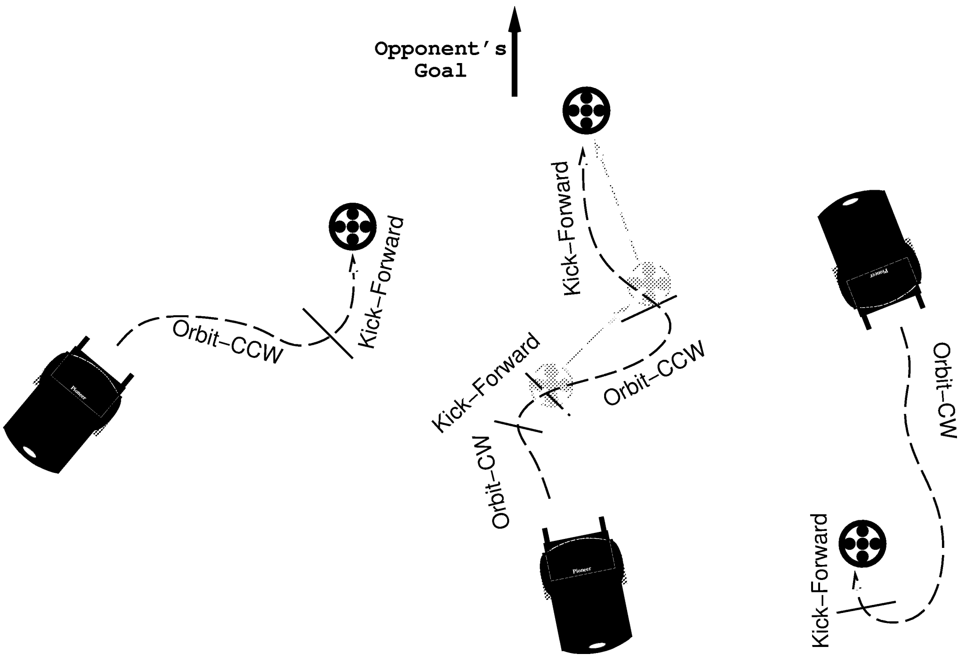

Fig. 4. Some typical trajectories generated by combination of the ball-manipulation basis behaviors.

into three components that we determined to be minimal yet sufficient for generation of all ball manipulation and trajectory generation. These three basis behaviors, *Orbit-CW*, *Orbit-CCW*, and *Kick-Ahead*, are illustrated in Fig. 3. *Orbit-CW* and *Orbit-CCW* approach the ball directly from a distance, but fall into an "orbit" around the ball when close; they differ only in direction (clockwise or counter-clockwise) of this orbit. *Kick-Ahead* basically pushes the ball forward with small corrections when necessary to keep the ball centered in the field of view, and in some situations (described in Section 7) performs the high-speed kick. Rapid switching between these behaviors (at a maximum rate of 10 Hz) based on environmental factors (described in the next section, and in Table 1) leads to trajectories such as those illustrated in Fig. 4, which are generated quickly enough to handle a moving ball.

4.3.2. Behavior selection

This switching between basis behaviors is handled by the *Orienter* behavior. It passes the visual position of the ball to one of the basis behaviors, selected as follows:

Given **B** is a bounding-box of the ball, **OG** of the opponent's goal, and **MG** of the robot's goal, and that "north" is the direction towards the opponent's goal,

If robot sees **OG**	
If **B** overlaps **OG**,	*Kick-Ahead*
Else if **B** is left of **OG**,	*Orbit-CW*
Else if **B** is right of **OG**,	*Orbit-CCW*
Else if robot sees **MG**	
If **B** is left of **MG**,	*Orbit-CCW*
Else if **B** is right of **MG**,	*Orbit-CW*
Else if robot facing north,	*Kick-Ahead*
Else if robot facing east,	*Orbit-CCW*
Else if robot facing west,	*Orbit-CW*

This behavior selection can be performed as lookup in a table such as Table 1. Whenever one of the basis behaviors receives the ball-position information, it in turn outputs a rotational velocity. If the ball is not seen, there is no output from the ball manipulation behaviors at all, and the default *Patrol* rotational velocity is used.

Table 1
A reactive policy for the *Orienter* behavior

Ball left of goal	Ball right of goal	Ball left of my goal	Ball right of my goal	Heading	Behavior
0	1	?	?	?	*Orbit-CCW*
1	0	?	?	?	*Orbit-CW*
1	1	?	?	?	*Kick-Ahead*
0	0	0	1	?	*Orbit-CW*
0	0	1	?	?	*Orbit-CCW*
0	0	0	0	West	*Orbit-CW*
0	0	0	0	East	*Orbit-CCW*
0	0	0	0	North	*Kick-Ahead*

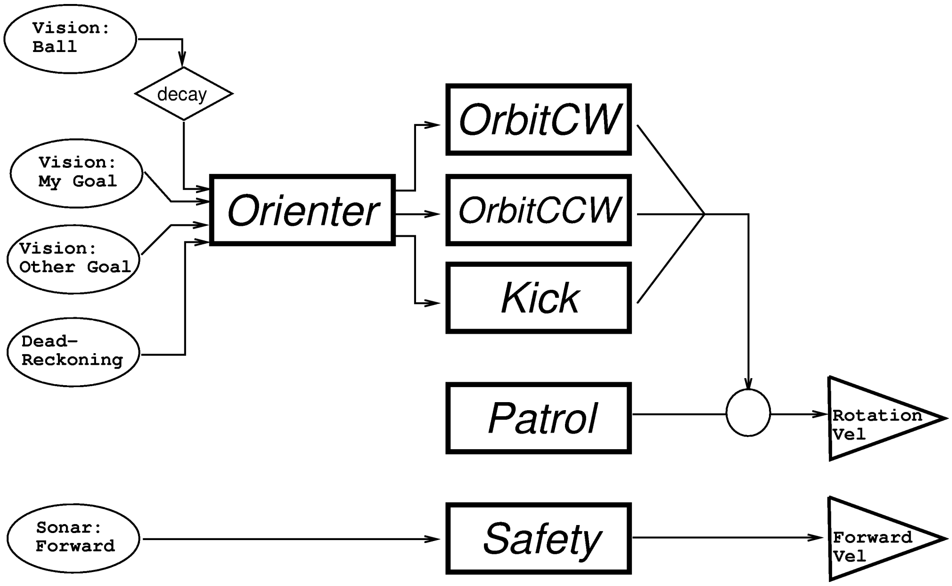

Fig. 5. Addition of the ball manipulation layer. Information flowing into the top of a circle, when available, overrides information flowing horizontally through the circle.

The generation of trajectories by combination of basis behaviors can best be understood by tracing the process on paper. The ball is always approached from a direction that minimizes the chance of its being accidentally knocked towards the robot's own goal—this is the importance of the east-west distinction. The robot will push the ball forward with the *Kick-Ahead* behavior, which makes minor heading adjustments to center the ball in the visual field, until its alignment to the ball and goal change enough to mandate a temporary switch to an *Orbit*.

Fig. 5 shows the behavioral structure of the ball-manipulation system.

4.3.3. Basis behavior implementation issues—hidden state

Given the sensing limitations of the non-actuated cameras on our robots, the implementation of the *Orbit* behaviors is not straightforward. As the robot only sees in the direction in which it is headed, it is unavoidable that the ball must be out of the robot's field of view for significant portions of the process of orbiting the ball. Thus the control system must deal with the issue of *hidden state*, important parts of the task environment that are not perceivable. Purely reactive systems are not well-suited to such non-Markovian environments [26].

Bowling et al. [10] claim that an accurate memory model is needed to overcome problems of hidden state in nondeterministic environments, giving examples from a simulated soccer system with limited perception. They advocate use of a probabilistic model that maintains "reasonable estimates" of locations of objects relative to the agent. Estimation of objects' motion within the environment and effects of the agent's motion must be calculated. They compare this favorably to the "simplest model of memory that provides enough functionality to be usable by the client", one that updates a memory of directions to unseen objects by the amount of agent rotation. Arkin and Balch [2] also make use of a world model that must be updated to keep track of other agents in the environment as a robot loses perception of them. The RoboCup Challenge [4] states that an actuated camera or extremely wide-angle lens (such as an omnidirectional one) is necessary to keep track of the ball and targets, and that modeling of ball speed and direction is key to catching and intercepting the ball.

On a more minimalist note, Littman [26] examines the improvement of performance brought to reactive systems by even a single bit of memory. In many cases, this minimal state addition can lead to optimal performance in non-Markovian environments.

We employ a "stateless" approach that has power similar to Littman's explicit addition of internal state, which we call *perceptual decay*. Quite simply, our robots retain an "afterimage" of the most recent perception of an object for a fixed period of time after such perception is lost. This afterimage is not distinguished from a direct perception by the behaviors within the system. When these behaviors are properly designed, the afterimage will lead to recovery of direct perception. Perceptual decay has become ubiquitous in all of our robotic systems that operate in dynamic environments.

In our soccer robots, the application of a few seconds of decay to visual perception of the ball not only effectively and "statelessly" solves all of our problems of hidden state, but allows us to exploit (as discussed below and in Section 5.1.1) the loss of perception of the ball in powerful ways. Our implementation of the ball-manipulation basis behaviors makes this clear.

4.3.4. Table implementation of ball-manipulation basis behaviors

Armed with this perceptual decay of the ball's position, we implement the three ball-manipulation basis behaviors as simple lookup tables that map the ball's position in the visual field to robot rotation. We divide the visual field into a 5×5 grid and assign a rotational velocity to each of the twenty-five resulting areas. Fig. 6 gives an example of such a table.

These tables are fairly robust; our initial guessing of the values was functional, though not optimal. Hand tuning was simple, but deciding the values of the entries in the table is a good problem for learning which we intend to explore.

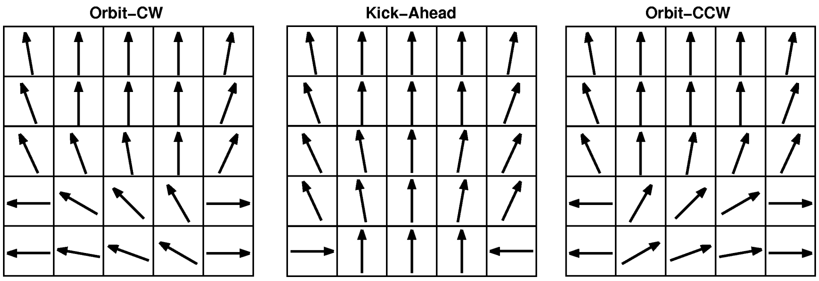

Fig. 6. Ball-Manipulation basis behaviors. Each of these behaviors is implemented as a table that assigns rotational velocities to segments of the visual field (represented here by arrows). The rapid acceleration away from the ball indicated in the lower corners of *Kick-Ahead* produce the "rear-end kick".

Careful examination of the table will reveal that "recovery" behavior is built into the edges of the visual field. When the ball leaves the field of view, it leaves an afterimage in one of the edge squares, and the rotation indicated by that entry in the table is performed until either the period of decay expires or visual perception of the ball is re-acquired. We place in all the edges rotational velocities that have a high likelihood of steering the robot back towards the ball.

Making use of this built-in recovery, the *Orbit* behaviors actually cause the robot to "tack" around the ball. When approaching from a distance, the ball tends to stay within the table. When the robot gets close to the ball and must circumnavigate it, it rotates away from the ball until it is fairly tangential and loses direct perception, then rotates back rapidly (tracing this process through Fig. 6 will help in understanding this process). This process repeats, as the robot moves forward at the speed determined by *Safety*, until vision or dead-reckoning indicate a change in behavior. The tacking occurs rapidly enough that a smooth trajectory around the ball is generated.

This tacking behavior insures that the robot thoroughly explores the space of headings while traveling around the ball, making it highly likely that at some point it will either see both the ball and the opponent's goal at the same time, or see the ball while dead-reckoning indicates a heading of roughly "north"—which, as indicated in Section 4.3.2, are the triggers for the *Kick-Ahead* behavior.

4.3.5. Emergent properties of ball-manipulation basis behaviors

Two major functionalities emerge from the interaction of the ball-manipulation behaviors and the de-coupled velocity control of *Safety*: detours and batting.

When a robot or other obstacle comes between a player and the ball, the player makes a detour around the obstruction without losing awareness of the ball. As the robot's velocity slows due to *Safety*'s perception of the obstacle, the tacking behavior becomes more pronounced and the robot tends to shift towards one side of the obstacle until it again has a clear path toward the ball. Since rotation is still doing what is necessary to tack around the ball, the robot rotates back towards the ball frequently enough that the perceptually-persistent "afterimage" of the ball is never lost. This same combination of perceptual decay and de-coupled velocity control is what allowed our hors-d'oeuvres serving robots [1] to navigate as a cohesive group through dense crowds with many obstructions.

When the ball is stuck against a wall (a common occurrence at RoboCup-97) or being advanced by an opponent, the robot still tries to tack around it. Again, when *Safety* detects the wall or robot behind the ball, it slows the velocity to an eventual standstill. In this case, the back and forth motion of the tacking behavior becomes extreme, and generally leads to the side of the robot's gripper "batting" the ball out of its stuck position. Since the *Orbit* behaviors cause the robot to interpolate between the ball and its own goal, this batting almost always causes the ball to move towards the opponent's goal.

4.3.6. Kicking—style without sequences

Consideration of the breakdown of the visual field in the behavior tables will reveal that there are two squares—the lower left and right corners—that can be quite problematical for the *Kick-Ahead* behavior. If the ball is perceived in one of these areas, it is likely to be just to the side of the robot (see Fig. 7). Attempting to rotate towards the ball would

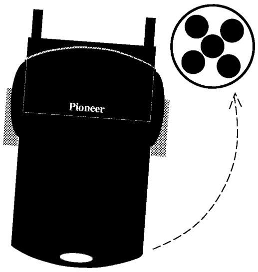

Fig. 7. The proper configuration for a rear-end kick.

likely knock it towards the side of the field, away from its path to the goal. It seems that the robot would either have to pass the ball and come around for a new approach, or back up far enough to re-align without danger of pushing it in the wrong direction. The former is unappealing because of the time it would take to re-acquire the ball and the chance that it might not be re-acquired before being stolen; the latter involves the type of sequenced behavior which we strive to avoid. Our solution to this problem exploits the physics of the Pioneers—their lack of rotational symmetry, which in some cases can make control more difficult, here turned out to be an asset. When the ball enters one of the problematic areas of the visual field, the robot rotates very rapidly *away* from the ball, and kicks the ball with its rear end as it comes around. Though the configuration necessary for this to occur successfully is fairly rare (once or twice a game), the accuracy and speed of a successful kick led to its being hailed as the most stylish move yet seen in RoboCup competition [31]. When the kick is not successful, the fast rotation causes the robot to regain visual contact with the ball quickly. This drastically different style of ball manipulation is implemented by merely placing two appropriate rotational velocities in the *Kick-Ahead* table.

4.3.7. Tolerance of uncertainty

The ball-manipulation is very robust to uncertainty because of redundancy and the reactivity of a feedback loop. It is not possible to rely on the vision system for localization from most areas of the field; the goals (the only perceivable fixed objects) are not visible from many angles, and are difficult to perceive as the robot gets farther away. Our other localization modality, however, is also very unreliable: the dead-reckoning system, on the soft RoboCup surface, drifts very quickly. We are, however, able to get acceptable performance by using the two redundantly: by using the simple north/west/east heading distinction, we reduce our need for heading accuracy to about $+/-60$ degrees. It is likely, however, that even within the timespan of a match the dead-reckoning will drift enough that even this distinction leads to error. To correct for this, we recalibrate the dead-reckoning system, using a rough approximation based on visual data, whenever the robot does perceive the goal clearly. This calibration has an error rate of approximately $+/-45$ degrees, but this falls well within our tolerances for effective advances. The heading error, though large, is within our tolerance, and is prevented from growing much larger. This insures that the vast majority of the time, using the dead-reckoning information will at least cause the robot to move the ball towards the opponent's goal, which makes it likely that the robot will reach some point where it can perceive the goal visually, and use the more accurate visual alignment.

The perceptual decay of the ball position adds to the robot's ability to tolerate visual uncertainty. Loss of visual contact with the ball for short periods of time, due to occlusion, vision system failure, lighting changes, or any other factor, are smoothed over. The robot acts as if the ball is still visible in whatever part of the visual field it most recently appeared in; this in practice is a fairly reliable estimate of current ball position for the short time it takes to regain visual contact. The "recovery behavior" built into the edges of the basis behavior tables make extremely effective use of perceptual decay to increase robustness.

De-coupling of behaviors, in simplifying the system, reduces many risks of uncertainty. An extremely simple behavior such as *Safety* is likely to be fairly foolproof; but as individual behaviors grow more complex it becomes difficult to foresee the effects of all possible inputs and prepare the system for all contingencies.

Finally, the "stateless" reactive nature of the system also contributes to robustness. Since the robot never has any commitment to a plan of action, its mistakes are usually corrected as soon as the sensors can bring in new information.

4.3.8. Sophistication of ball-manipulation behavior

We believe that our ball-manipulation behavior operates on a level comparable to teams such as [47] which use global vision systems for a clear view of all objects in the environment. They describe trajectories that bring robots around the ball and into a position to advance, and which detour around other agents, based on calculation of motion of objects in the environment and setting of intermediate targets for segmented approaches. Our robots also navigate around the ball as appropriate in order to line up for an advance, and make detours around opponents, but do so with no such calculation, internal representation, or global information. The sophistication of having three effective ball-handling techniques is not only unmatched by other systems, but achieved without any added control system complexity. Furthermore, rather than being unable to function in an inaccessible (hidden-state) environment as many predict a strictly behavior-based system would be, our players exploit loss of perception to generate the tacking behavior which leads to both the thorough exploration of heading space necessary to trigger behavior changes, and the useful emergent behaviors of extracting stuck balls by batting and detouring around obstacles. If, as the principle of emergence dictates, intelligence is in the eye of the beholder [13], the purposeful, smooth, dynamically-adjusted trajectories generated by the high reactivity of this approach are similar in intelligence to many of the precisely-calculated, segmented trajectories generated by deliberative systems.

5. Team cooperation

Particularly in the area of coordination of teams of robots there is intense debate over deliberative, behavior-based, and hybrid control strategies. Many researchers state that various types of deliberation, models of other agents, and explicit communication are necessary for cooperative behavior [7,8,22,24,32,42,44,45,47]; others advocate ethologically-inspired systems which cooperate only through environmental interaction [9,16,17,25,27, 28,32,53], and still others [18,19,21,32,49] pursue other methodologies for minimal control of groups of robots. In [3,16,24,27,33], general discussions of the tradeoffs between local and global control in multi-robot systems are presented.

Arkin [3] proposes three dimensions of group organization for the study of multi-agent system efficiency: communication, organization, and task type and complexity. Brooks [16] names a different set of key issues: emergence (especially the ability to predict global behaviors of locally interacting agents and effects of small changes to individuals); individuality (robustness of systems of interchangeable agents); cooperation (the ability to achieve more by cooperating than by acting individually); interference (the ability to avoid it when desirable); density dependence (how group efficiency changes as group size changes); and communication (the need to develop a clear understanding of the relation between communication increase and performance increase). Donald [19,20] has developed the beginnings of a formal method for comparisons of tradeoffs between sensing, computation, and communication in multi-robot systems, and a methodology for systematically reducing resource requirements. We will discuss these issues further below, but for convenience summarize all of the arguments: communication and shared models allow the use of well-known and understood traditional AI methods, but tend to be brittle when faced with real-world uncertainty; while ethologically inspired models are very robust but depend on emergent behavior, which is difficult to analyze, and do not have a clear path of scalability to tasks more complex than insect-like behavior.

Tambe [44,45] advocates general models of teamwork, rather than common domain-specific plans, to address the uncertainties of complex, dynamic environments. He claims that preplanned, domain specific coordination is not only inefficient in terms of reuse but prone to drastic failure—it is impossible to plan for all contingencies. Through joint intentions [22], knowledge of their own and others' commitments and responsibilities, and a "deep" or causal model of teamwork, which lets them reason about coordination and communication, agents can select from among reactive plans for roles in group efforts.

Stone and Veloso [42] counter that such negotiation in real-world, real-time domains is infeasible. They take issue with Tambe's [44] assertions about the inflexible nature of pre-determined team actions, claiming that the problem lies with the rigid roles that are commonly used in plans. They introduce PTS (periodic team synchronization) domains, such as soccer, where teams have the occasional ability to pause and communicate reliably, and *formations* which have homogeneous subgroups that can interchange roles dynamically. *Locker room agreements*, made during the periodic synchronizations, are used to determine protocols for changing of roles or formations.

Both Tambe [44] and Stone and Veloso [42] seem to be suffering from the "horizon effect" described some years ago by Brooks [13]: they are getting somewhat better performance from their traditional systems by including reactive components for the present, but are really only pushing the limitations of their deliberative systems a little further into the future. They both concede the difficulty of failure detection and recovery within such hybrid systems. Tambe [44] lists nine "illustrative examples" of breakdown of teamwork in planning systems; five are due to interrupted sequences of behavior, two are due to communication failure, and two are due to failure to spread symbolic information properly around the system. Although his work addresses these failures, it does so by making the failing mechanisms more reliable rather than seeking to replace them with others that are not fundamentally error-prone.

It is often assumed that team cooperation entails some type of explicit communication between team members and a shared world model. “Ethereal” communication (such as by radio), however, adds a great deal of complexity; often, in practice, with little benefit. In many cases, when a symbolic message is received, a mapping must be performed between the perceived environment and the information of the message. This mapping is subject to both error of perception of the sender and error of the perception of the receiver. The decision of how to target communication can be very difficult: either one specific robot is the target of the message, in which case the decision of *which* to send to can be very difficult (requiring mapping of symbolic representations to perceived objects in the environment), or the message can be broadcast to all robots, which will respond *en masse*, only to have to sort out later, through perception, what each should do. It is far easier, when possible, to skip straight to the perception. This is one of the reasons why traditional planning for teams of robots can be so brittle in dynamic, uncertain environments: if one robot fails, or if some part of an internal representation gets out of synch with the physical world, reorganization of the physically embodied system through symbolic means requires prohibitive analysis and model verification.

Kraus [24] begins to look at systems that take a “classical mechanics” approach to team coordination. Robots can be given behaviors that have properties familiar to particle physics. The complexity of such systems is low, and they can be analyzed using theory from physics. This is somewhat analogous to ethologically inspired *stigmergic* systems, which interact through their effects on the environment in a manner similar to the group formation of efficient pheromone trails by ants [53]; ant-like collective sorting [9]; adaptive territorial and task division [17,37], and flocking and foraging behavior [28]. These systems are robust to failure because they do not assign specific roles to specific robots, and do not specify how tasks are to be accomplished. No unit is essential, and error in many cases can be “more creative than inefficient” [17]: interesting behaviors tend to emerge from unexpected events, rather than system failures.

Below we describe the cooperative behaviors we implemented for the “Spirit of Bolivia”. Though based on an extremely simple modification of the system we have already presented, we believe that our team behavior embodies many of the ideals that researchers in both stigmergic and deliberative camps strive for, while suffering few of the pitfalls they seek to avoid.

5.1. RoboCup team behavior

The only modification of our behavioral system to achieve team cooperation was the addition of a simplified form of *Dispersion* [28]: when the robots sense something close to their sides through sonar, they move away from it a little bit. When merged in with our other behaviors (see Fig. 8), this leads to very interesting emergent effects.

5.1.1. Offensive formation

In an offensive situation, robots are trying to align with the ball and move it toward the opponent’s goal. The robots do not collide head-on in pursuit of the ball, due to the *Safety*

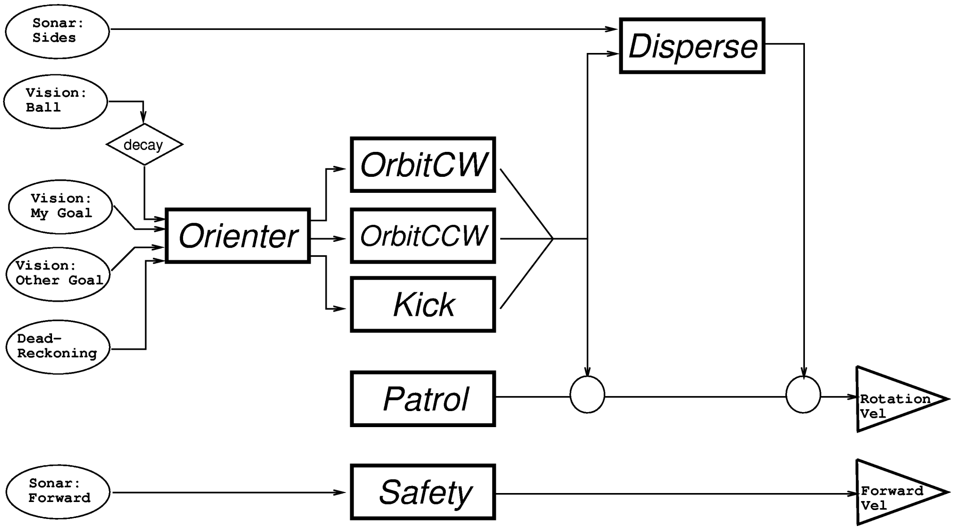

Fig. 8. The final behavioral system from RoboCup-97.

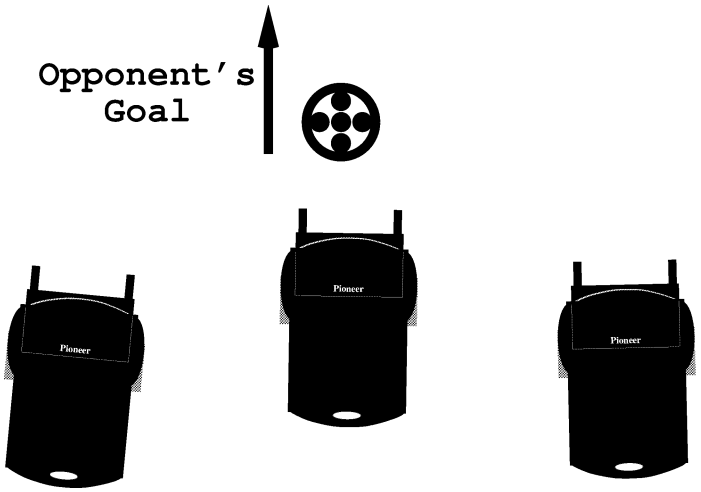

Fig. 9. Offensive group formation.

behavior, and maintain a set distance from the robots on their sides due to *Disperse*. Their tendency, however, due to the ball-manipulation behaviors, is to approach the ball from directly behind. One robot will by serendipity be the first to be so aligned and push the ball forward. Any other robots close to the ball will continue trying to get directly behind it, but will be caused to veer away by *Disperse*'s reaction to the physical presence of the first robot. The robots on the sides stay slightly behind, because they slow down as they draw abreast of the ball and turn more sharply towards it (thereby causing *Disperse* to perceive the pushing robot). As the lead robot pushes the ball forward, the other robots will continue to follow the ball, remaining roughly parallel to and slightly behind the ball-pushing robot due to the competition between *Disperse* and the ball-manipulation behaviors. The result is motion across the field in the reliable formation shown in Fig. 9. There is no "decision" to enter such a formation; it follows naturally from the robots' "attraction" to the ball and "repulsion" from each other, in situations where the ball is moving forward.

This formation provides an effective "fumble protection" that is very important in the robotic soccer domain. A robot will often accidentally knock the ball off course while dribbling it forward; this formation provides backup and recovery. In our team it is not uncommon for possession of the ball to transfer between the robots of an advancing group without loss of possession by the team. The formation also provides for a very quick defense (discussed below) if the ball is stolen.

The size of the offensive formation is limited as well by the interaction between the *perceptual decay* of the ball position and the *Disperse* behavior. As the group grows larger, peripheral robots tend to lose sight of the ball for longer amounts of time. This provides for a "drop out" of group members; once they lose perception of the ball for more than the decay period, they leave the formation by reverting to their *Patrol* behavior. When the robots are "dressed" for RoboCup, the stable group size is three.

5.1.2. Defensive group formation

In a defensive situation the ball is not advancing, so that the same attractive and repulsive forces cause the robots to fall into a semi-circular arrangement around the ball rather than the V-formation of the advance (see Fig. 10). This formation very effectively prevents the opponent from continuing to move the ball up the field, and places players in a good position to gain possession of the ball. The robots engage in the emergent "batting" behavior as they get close to each other and the opponent; in a pass-like maneuver the robot

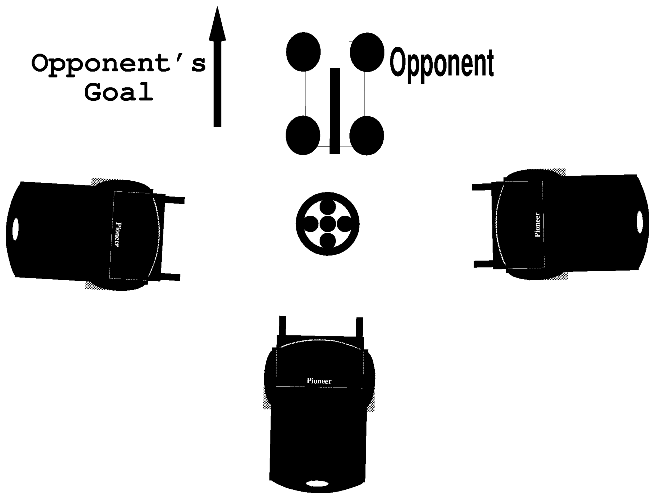

Fig. 10. Defensive group formation.

directly opposite the opponent can bat the ball sideways to a teammate, which will quite smoothly take possession as it follows the motion of the ball away from the center of the formation.

5.1.3. Transition between formations

Given a *Patrol* behavior as we have discussed in Section 4.2 which assigns a rough territory to each player, we get fairly sophisticated formation behavior. The robots patrol their territories when not close to the ball, dynamically enter population-controlled formations when they are useful for advancing the ball, and drop into defensive formations when they need to block an opponent advance and turn it around. The offensive or defensive response is triggered solely by the *behavior* of other agents and the ball, with no need for symbolic distinction of teammates from opponents.

5.2. Comparisons

Balch and Arkin [7,8] have done extensive work with formations of robots. These formations consist of simple geometric patterns of four (or in some cases two) robots. Their approach requires global knowledge of the direction in which the group is traveling and a predetermined spatial relation to at least one specific robot in the group. Balch and Arkin [7] discuss performance problems linked to radio communication as a demonstration of the utility of a passive (environmental) communication approach, and in [8] they mention that problems of robot failure have not been addressed, and that possible solution paths are automatic reconfiguration and fault-tolerant communications. We believe that the sophistication of our offensive formation is not far from that of Balch and Arkin's work. Our formations, however, are not dependent on specific relations between specific robots, and thus exhibit some of the tolerance to robot failure that they seek. Our soccer system also performs something akin to automatic reconfiguration in the case of such failure; new robots in the environment are incorporated as necessary to maintain the stable group size determined by the *density dependence* [16] that arises from the interaction dynamics of the system (as discussed in Section 9).

Parker's [33] work on formations compares systems with varying amounts of global knowledge. Performance of the strictly local system was relatively poor, but we attribute this at least in part to the lack of a true minimalist approach. The robots were aligning themselves relative to the *heading* of the their peers, rather than their mere relative location. This caused formations to break during sharp turns as robots tried to stay, for example, on the left side of some other robot, rather than trying to keep the other robot to their left side. Again, the formations of four robots required specific spatial relations between specific robots, and were thus brittle to robot failures or behavioral fluctuations.

Parker also discusses the suitability of behavioral analysis as a robust approximation to global knowledge. This is realized to some extent as well in our formation behaviors—it is the behavior of the ball and other robots, rather than any symbolic distinction that causes transitions from patrolling to offensive to defensive behavior. Her assertion that global goals (known at design time) allow more local control is another factor that allows minimization of systems such as the "Spirit of Bolivia".

We hope that we have shown that, contrary to what is claimed by Tambe [45], it is not necessarily true that “an agent must be provided “deep” or causal models of its domains of operation”, such as a general model of teamwork, for effective cooperation. In the simulated soccer domain, Tambe says that failure detection and recovery requires advanced spatial reasoning and agent tracking/plan recognition skill, and because of this has not been implemented, leaving the team susceptible to breakdown [45]. The need for agents to share joint intentions—to know the intentions of their teammates—makes the chance of such failures high in uncertain domains. The flexibility brought to pre-determined team actions by Stone and Veloso’s *formations* [42] is a step towards embracing the minimalist principle of homogeneity, but still requires that at some point specific roles be assigned to specific robots, and that agents negotiate in order to change roles or formations. Challenges they list for their *locker-room agreements* include determination of when to switch roles or formations, smooth transitions of roles and formations, and how to make sure that all agents use the same formation. Though its transitions are simpler, our current system makes some headway towards addressing these challenges. Robots reliably enter into and switch roles smoothly and in appropriate situations, and flow in and out of formations (in the colloquial sense) in appropriate situations. As we prepare for RoboCup-98 by incorporating into our system the ability to recognize teammates and opponents on the field, we expect to scale our formations and strategies to the point where our minimalist “stateless” strategies will have the power to generate team behaviors as complicated as those generated by deliberative systems, with a greater level of robustness, as they have done for the ball manipulation behaviors.

6. Future work

In the future, we would like to include passing in our repertoire of ball-manipulation behaviors. This will require reliable visual distinction of teammates and opponents on the field, and a kicking device capable of propelling the ball faster than the robot can move, with more general applicability than the current “rear-end kick”. An improved *Patrol* behavior described in Section 4.2, will allow team members to locate themselves in advantageous positions to receive passes. One additional behavior, similar to *Orienter*, will identify a potential pass receivers; another will activate the kicking device at appropriate moments (based on input from *Orienter* and the receiver-identifier). With these capabilities we believe we will have a number of new formations, good localization around the field, and effective passing behavior.

We also plan to use learning to train elements of the system such as the values of the ball-manipulation basis-behavior tables, the identification of pass-receivers, and the parameters for localization through the *Patrol* behavior as described in Section 4.2. We would also like to develop an on-line learning system that is able to dynamically improve the robot’s ability to distinguish field objects: given a number of situations where we are fairly sure we see an object (such as we have defined for the goal, for use in recalibrating the compass (Section 4.3.7)), the robot can track the progression of perceptions that approach these situations over time.

7. Conclusion

The behavior-oriented AI "revolution" has barely begun, and it can be a hard climb from the bottom up. We hope that we have demonstrated that, at least in the dynamic arena of robotic soccer, minimalist behavior-based systems have the potential to scale to levels of behavior that rival "classical AI" and hybrid systems not only in robustness but in sophistication of behavior.

We have presented the design principles of *minimalism, statelessness,* and *tolerance*, and described their application to the design of a RoboCup robotic soccer team. This system displays an outstanding level of sophistication: effective obstacle avoidance in a dynamic environment, generation of smooth, effective trajectories, three separate methods of ball handling, and dynamic configuration into appropriate population-limited offensive and defensive formations. This is all done in a strict behavior-based manner with no internal models of the environment, calculation of trajectories, or explicit communication, by a minimal control system that occupies fewer than four pages of LISP code and was developed in less than a week. We have compared this team to others which have been developed over much longer periods of time, and which have access to global views of the environment, more powerful processing and communication, and come to the conclusion that these capabilities do not make a great difference to the level of behavioral sophistication achievable. Furthermore, we have shown that certain problems of brittleness that plague these more "traditional" systems do not apply to minimalist behavior-based systems. Finally, we have outlined goals for the scaling up of our soccer system to more complex team behavior in the future.

We hope that others find our principles as fruitful as we have, or are at least inspired to ask themselves occasionally, "can I do this with only perception?", "do I really need all this complexity?", and "can I avoid this need for precision?"

References

[1] R. Arkin, The 1997 AAAI mobile robot competition and exhibition, AI Magazine 19 (1998) 3.

[2] R. Arkin, T. Balch, Cooperative multiagent robotic systems, in: D. Kortenkamp, R.P. Bonasso, R. Murphy (Eds.), AI-based Mobile Robots: Case Studies of Successful Robot Systems, MIT Press, Cambridge, MA, in press.

[3] R. Arkin, J.D. Hobbs, Dimensions of communication and social organization in multi-agent robotic systems, in: Proc. 2nd International Conference on Simulation of Adaptive Behavior, 1992.

[4] M. Asada, P. Stone, H. Kitano, B. Werger, Y. Kuniyoshi, A. Drogoul, D. Duhaut, M. Veloso, H. Asama, S. Suzuki, The RoboCup physical agent challenge: Phase I, Applied Artificial Intelligence 12 (1998).

[5] M. Asada, S. Suzuki, Y. Takahashi, E. Uchibe, M. Nakamura, C. Mishima, H. Ishizuka, T. Kato, M. Asada, Vision-based Robot Learning Towards RoboCup: Osaka University "Trackies", in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.

[6] T. Balch, Social entropy: A new metric for learning multi-robot teams, in: Proc. 10th International FLAIRS Conference, 1997.

[7] T. Balch, R. Arkin, Behavior-based formation control for multi-robot teams, Submitted for publication, 1997.

[8] T. Balch, R. Arkin, Motor schema-based formation control for multi-agent robot teams, in: Proc. 1995 International Conference on Multiagent Systems, 1995.

[9] R. Beckers, O. Holland, J. Deneubourg, From local actions to global tasks: Stigmergy and collective robotics, in: Proc. 4th International Workshop on the Synthesis and Simulation of Living Systems, 1994.

[10] M. Bowling, P. Stone, M. Veloso, Predictive memory for an inaccessible environment, in: Proc. International Conference on Intelligent Robots and Systems, 1996.
[11] R. Brooks, A robust layered control system for a mobile robot, IEEE J. Robotics and Automation 2 (1986).
[12] R. Brooks, C. Rosenberg, L—A common lisp for embedded systems, in: Proc. Lisp Users and Vendors Conference, 1995.
[13] R. Brooks, Intelligence without reason. Computers and Thought, in: Proc. IJCAI-91, Sydney, Australia, 1991.
[14] R. Brooks, Intelligence without representation, Artificial Intelligence 47 (1991) 139–159.
[15] R. Brooks, Elephants don't play chess, Robotics and Autonomous Systems 6 (1990) 3.
[16] R. Brooks, Challenges for complete creature architectures, in: Proc. First International Conference on Simulation of Adaptive Behavior, 1990.
[17] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C Detrain, L. Chretien, The dynamics of collective sorting: Robot-like ants and ant-like robots, in: Proc. First International Conference on Simulation of Adaptive Behavior, 1990.
[18] B. Donald, J. Jennings, D. Rus, Minimalism + distribution = supermodularity, J. Experiment. Theoret. Artificial Intelligence (JETAI) 9 (1997) 2–3.
[19] B. Donald, J. Jennings, D. Rus, Information invariants for distributed manipulation, Internat. J. Robotics Research 16 (5) (1995).
[20] B. Donald, On information invariants in robotics, Artificial Intelligence 72 (1995) 217–304.
[21] G. Dudek, M. Jenkin, E. Milios, D. Wilkes, Experiments in sensing and communication for robot convoy navigation, in: Proc. IEEE International Conference on Intelligent Robots and Systems, 1995.
[22] N. Jennings, Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, Artificial Intelligence 75 (1995) 195–240.
[23] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, RoboCup: The Robot World Cup Initiative, in: Proc. IJCAI-95, Montreal, Quebec, 1995.
[24] S. Kraus, Negotiation and cooperation in multi-agent environments, Artificial Intelligence 94 (1997) 79–97.
[25] C.R. Kube, H. Zhang, Collective robotic intelligence, in: Proc. 2nd International Conference on Simulation of Adaptive Behavior, 1992.
[26] M. Littman, Memoryless policies: Theoretical limitations and practical results, in: Proc. 3rd International Conference on Simulation of Adaptive Behavior, 1994.
[27] M. Matarić, Issues and approaches in the design of collective autonomous agents, Robotics and Autonomous Systems 16 (1995) 321.
[28] M. Matarić, Designing and understanding adaptive group behavior, Adaptive Behavior 4 (1995) 1.
[29] M. Matarić, Behaviour-based control: Examples from navigation, learning, and group behavior, J. Experiment. Theoret. Artificial Intelligence 9 (1997).
[30] I. Noda, S. Suzuki, H. Matsubara, M. Asada, H. Kitano, Overview of RoboCup-97, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.
[31] D. Normile, RoboCup soccer match is a challenge for silicon rookies, Science 277 (1997).
[32] S. Onn, M. Tennenholtz, Determination of social laws for multi-agent mobilization, Artificial Intelligence 95 (1997) 155–167.
[33] L. Parker, Designing control laws for cooperative agent teams, in: Proc. IEEE International Conference on Robotics and Automation, Atlanta, GA, 1993.
[34] L. Parker, Adaptive action selection for cooperative agent teams, in: Proc. 2nd International Conference on Simulation of Adaptive Behavior, 1992.
[35] M. Sahota, A. Mackworth, Can situated robots play soccer?, in: Proc. 10th Biennial Conf. of the Canadian Society for Computational Studies of Intelligence, Bauff, Alberta, 1994.
[36] R. Sargent, B. Bailey, C. Witty, A. Wright, Dynamic object capture using fast vision tracking, AI Magazine 18 (1) (1997).
[37] M. Schneider Fontan, M. Matarić, A study of territoriality: the role of critical mass in adaptive task division, in: Proc. 4th International Conference on Simulation of Adaptive Behavior, 1996.
[38] W.-M. Shen, J. Adibi, R. Adobbati, B. Cho, A. Erdem, H. Moradi, B. Salemi, S. Tejada, Toward integrated soccer robots, AI Magazine 19 (1998) 3.
[39] W. Shen, J. Adibi, R. Addobati, B. Cho, A. Erdem, H. Moradi, B. Salemi, S. Tejada, Autonomous soccer robots, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.

[40] T. Smithers, On why better robots make it harder, in: Proc. 3rd International Conference on Simulation of Adaptive Behavior, 1994.

[41] L. Steels, The artificial life roots of artificial intelligence, Artificial Life 1 (1–2) (1994).

[42] P. Stone, M. Veloso, Task decomposition and dynamic role assignment for real time strategic teamwork, in: 3rd International Conference on Multi-Agent Systems (ICMAS-98), Submitted.

[43] S. Suzuki, Y. Takahashi, E. Uchibe, M. Nakamura, C. Mishima, H. Ishizuka, T. Kato, Trackies: RoboCup-97 middle-size league world champions, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.

[44] M. Tambe, Towards flexible teamwork, J. Artificial Intelligence Research 7 (1997).

[45] M. Tambe, Implementing agent teams in dynamic multi-agent environments, Applied Artificial Intelligence (1997).

[46] E. Uchibe, M. Asada, K. Hosoda, State space construction for behavior acquisition in multi agent environments with vision and action, in: Proc. International Conference on Computer Vision, 1998.

[47] M. Veloso, P. Stone, K. Han, S. Achim, The CMUnited-97 small robot team, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.

[48] B. Werger, The Spirit of Bolivia: Minimal control for comprehensive team behavior, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.

[49] B. Werger, Principles of minimal control for comprehensive team behavior, in: Proc. IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998.

[50] B. Werger, Scripted behavior for theatrical robot actors. Ullanta performance, Robotics Technical Report No. 98-01, 1998.

[51] B. Werger, Multiple agents from the bottom up, in: Proc. AAAI-97, Providence, RI, 1997.

[52] B. Werger, Profile of a winner: Brandeis University and Ullanta Performance Robotics' 'Robotic Love Triangle', AI Magazine 19 (3) (1998).

[53] B. Werger, M. Matarić, Robotic food chains: Externalization of state and program for minimal-agent foraging, in: Proc. 4th International Conference on Simulation of Adaptive Behavior, 1996.

[54] E. Yakota, K. Okazi, A. Matsumoto, K. Kawabata, H. Kaetsu, H. Asama, Omni-directional autonomous robots cooperating for team play, in: H. Kitano (Ed.), RoboCup-97: The First Robot World Cup Soccer Games and Conferences, Springer, Berlin, 1998.