Using the RoboCup Simulation Environment to Study the Advantage of the Environment Contributing to Cognition[1]

Sanjay Chandrasekharan[1], Babak Esfandiari[2], Neal Arthorne[2]

[1]Cognitive Science Ph.D. Program,
[2]Department of Systems and Computer Engineering,
Carleton University, Ottawa, Canada, K1S 5B6
{schandra, babak}@sce.carleton.ca, narthorn@connect.carleton.ca

Abstract. We report an experiment where the RoboCup simulation environment was used to study the cognitive advantage of signaling structures, which are task-specific structures some agents add to the world to help in decision-making. The RoboCup environment was chosen to test this strategy because of its noisy, dynamic and adversarial nature, which makes it very close to the real-world environment faced by organisms. We used the passing problem in RoboCup as our test problem. It was found that when signaling structures were used for passing, 48 percent of the passes were completed successfully, and the home team retained control of the ball 70 percent of the time.

1 Introduction

Many organisms add structures to the world to reduce their own and others' cognitive complexity. Wood mice (*Apodemus sylvaticus*) distribute small objects, such as leaves or twigs, as points of reference while foraging. They do this even under laboratory conditions, using plastic discs. Such "way-marking" diminish the likelihood of losing interesting locations [1] during foraging. Red foxes (*Vulpes vulpes*) use urine to mark food caches they have emptied. This marking acts as a memory aid and helps them avoid unnecessary search [2]. Ants drop pheromones to trace a path to a food source. Many mammals mark up their territories. The bower bird creates colorful nests to attract mates [3]. Many birds advertise their desirability as mates using some form of external structure, like colorful tails, bibs etc. [4]. Other animals have signals that convey important information about themselves to possible mates and even predators [3].

Such epistemic structures (usually termed signals) form a very important aspect of animal life across biological niches. These structures help the organism hive off a part of the cognitive load to the world. How much cognitive advantage do such structures provide in a noisy, extremely dynamic and adversarial environment? This is the problem we address in this paper. We used the RoboCup simulation environment to study the cognitive advantage provided by this strategy in such environments.

Before we describe our experiment, we will use one case of signaling to analyze the advantage provided by the epistemic structure strategy. Consider the peacock's tail, the paradigmatic instance of an animal signal. The tail's function is to allow female peacocks (peahens) to make a mating judgment, by selecting the most-healthy male [3]. The tail reliably describes the inner state of the peacock, that it is healthy (and therefore has good genes). The signal is reliable because it pays only a peacock with enough resources to produce a flamboyant tail. Sickly males cannot spend resources to produce ornaments. Thus the health of the peacock is directly encoded in the tail; the peacock carries its internal attributes on its tail, so to speak.

[1] Carleton University Cognitive Science Technical Report 2004-04
URL: http://www.carleton.ca/iis/TechReports

To see the cognitive efficiency of this mechanism, imagine the peahen having to make a mating decision without the existence of such a direct and reliable signal. The peahen will need to have a knowledge base of how the internal state, of health, can be inferred from behavioral and other cues. Let’s say “good dancing”, “lengthy chase of prey”, “long flights” (peacocks fly short distances), “tough beak” and “good claws” are cues for the health of a peacock. To arrive at a decision using these cues, first the peahen will need to “know” these cues, and that some combinations of them implies that the male is healthy.

Armed with this knowledge, the female has to sample males for an extended period of time, and go through a lengthy sorting process based on the cues (rank each male on each of these cues: good, bad, okay). Then it has to compare the different results, keeping all of them in memory, to arrive at an optimal mating decision. This is a computationally intensive process.

The tail allows the female peacock to shortcut all this computation, and go directly to the most-healthy male in a lot. The tail provides the peahen a single, chunked, cue, which it can compare with other similar ones to arrive at a decision. The tail is a task-specific structure. It exists just for the peahen to make the mating decision. The other cues (like tough_beak etc.) do not exist for this purpose, they have to be synthesized in a way that helps with the mating task. The tail 'fits' the peahen's task, and provides a standardized way of arriving at a decision, with the least amount of computation. The peacock describes itself using its tail. Reliable self-description, like the peacock’s tail, is one of nature’s ways of avoiding long-winded sorting and inference. In using self-describing metadata structures like XML, we are seeking to emulate nature’s design in the Semantic Web.

The peacock example (and others above) shows that the reduction of others’ cognitive complexity using external informational structures is very common, and it can be considered one of the building blocks of nature. Signaling exists at all levels of nature, from bacteria to plants, crickets, gazelles and humans. Note that the signal provides cognitive congeniality to the receiver, and not to the sender. The sender, for instance the peacock, gains because he has an interest in being selected for mating. This is very similar to the case of semantic mark-up, where the cognitive congeniality (less processing load) is for the reader, the encoder does the marking up because she stands to gain in some way.

Even though signaling is a basic structure of cognition, it has received very little attention from agent design methodologies. Many researchers have considered the role of stigmergy in changing environment structure. Stigmergy is a coordination mechanism where the action of one individual in a colony triggers the next action by others [5]. It is a form of indirect communication, and has been a favored mechanism for situated AI because it avoids the creation of explicit representations.

In the following section we develop a framework to understand how epistemic structures like signals fit in with agent-environment relationships in current agent design.

2 Agent Design Taxonomy

We categorize agent design into four design methodologies or frameworks, after [6]. To illustrate these four frameworks, we will use the problem of providing disabled people access to buildings. There are four general approaches to solve this problem.

Approach 1. This involves building an all-powerful, James Bond-style vehicle that can function in all environments. It can run, jump, fly, climb spiral stairs, raise itself to high shelves, detect curbs etc. This design does not incorporate detailed environment structure into the vehicle, it is built to overcome the limitations of all environments.

Approach 2. This involves studying the vehicle's environment carefully and using that information to build the vehicle. For instance, the vehicle will take into account the existence of curbs (and them being short), stairs being non-spiral and having rails, level of elevator buttons etc. So it will have the capacity to raise itself to short curbs, climb short flight of straight stairs by making use of the rails etc. Note that the environment is not changed here.

Approach 3. This involves changing the environment. For instance, building ramps and special doors so that a simple vehicle can have maximum access. This is the most elegant solution, and the most widely used one. Here structure is added to the environment, so that it contributes to the agent's action. Our analysis will focus on this approach.

Approach 4. The fourth one is similar to the first one, but here the environment is all-powerful instead of the vehicle. The environment becomes "smart", and the building detects all physically handicapped people, and glides a ramp down to them, or lifts them up etc. This solution is an extreme case of approach III, we will ignore it in the following analysis.

The first approach is similar to the centralized AI methodology, which ignores the structure provided by specific environments during design. The environment is something to overcome, it is not considered a resource. This methodology tries to load every possible environment on to the agent, as centrally stored representations. The agent tries to map the encountered world on to this internal template structure.

The second approach is similar to the situated AI model promoted by Rodney Brooks [7]. This methodology recognizes the role of the environment as a resource, and analyses and exploits the detailed structure that exists in the environment while building the agent. Notice that the environment is not changed here. This is a passive design approach, where the environment is considered a given.

In the third approach, the designer actively intervenes in the environment and gives structure to it, so that the agent can function better. This is *Active Design*, or agent-environment co-design. The idea is to split the intelligence load — part to the agent, part to the world. This is agent design guided by the principle of distributing cognition, where part of the computation is hived off to the world. Kirsh [8] terms this kind of "using the world to compute" active redesign.

This design principle underlies many techniques to minimize complexity. At the physical level, the Active Design principle can be found in the building of roads for wheeled vehicles. Without roads, the vehicles will have a hard time, or all vehicles will need to have tank wheels. With roads, the movement is a lot easier for average vehicles. This principle is also at work in the "intelligent use of space" where people organize objects around them in a way that helps them execute their functions [9]. Kitchens and personal libraries (which use locations as tags for identifying content) are instances of such use of space in cognition.

A good example of Active Design at the information level is bar coding. Without bar coding, the checkout machine in the supermarket would have to resort to a phenomenal amount of querying and object-recognition routines to identify a product. With bar coding, it becomes a simple affair. The Semantic Web enterprise is another instance of Active Design at the information level. The effort is to create structure in an information environment (the Web) so that software and human agents can function effectively in it. The Active Design principle is also at work in the Physical Markup Language effort, which tries to develop a common standard to store information in low-cost Radio-frequency Identification (RFID) tags. These tags can be embedded in products, quite like meta-tags in web pages. Such tagged objects can be easily recognized by agents fitted with RFID readers (for instance, robots working in a recycling plant).

The Active Design approach is applied at the social level as well, especially in instances involving trust. Humans actively create structure in the environment to help others make trust decisions. Formal structure created for trust includes credit ratings, identities, uniforms, badges, degrees, etc. These structures serve as reliable signals for people to make trust decisions. Less reliable, and more informal, structure we create include standardized ways of dressing, talking etc.

3 Using RoboCup to Study Epistemic Structure

We have modeled the efficiency of the epistemic structure strategy using genetic algorithms and Q-learning [10], but in both cases we used a largely static environment with almost no noise or adversaries. The results show that the advantage of using the epistemic structure (ES) strategy is quite significant, the agents spend 58% of their time generating such structures. But such static environments do not provide a sense of the cognitive advantage provided by the ES strategy, because most organisms live in noisy, dynamic and adversarial environments,

RoboCup provides an interesting dynamic and adversarial environment to study the efficiency of the epistemic structure strategy. However, being a game, there is not much scope to add task-specific structure to the environment. The only structure that can be added to the RoboCup environment is 'yells', or signals from teammates. We chose to use this structure, and studied the passing problem (i.e. how an agent in control of the ball can decide whom to pass the ball) to test the efficiency of this structure. For our study, we developed RoboCup teams that used three different approaches to passing. The teams were based on the UvA TriLearn 2002 source code [23].

3.1 Team 1: Centralized Passing

This team (A1) uses approach 1 in our agent design taxonomy. A1 does not depend on task-specific information from other agents. In A1, when an agent has possession of the ball (i.e., the ball is within a kickable margin), it calculates the pass suitability (passability) for each teammate, and passes the ball to the teammate with the highest passability. If no teammate has passability above a fixed threshold value, the agent will dribble the ball toward the opponent goal.

The goalie in this team is based on the original UvA algorithm, except for one modification: in a goal kick or free kick, the goalie will use A1 to calculate the best receiver for a pass and kick the ball to that teammate. This differs from the UvA standard behavior of the goalie kicking the ball straight down the field. The A1 passing algorithm is described below:

```
A1: Centralized Passing
   Input(s):  None.
   Output(s): Best pass receiver

// set the minimum passability
   Pb <- passabilityThreshold

// initialize best pass receiver to none
   receiver <- none

   for each Teammate except goalie
        Pt <- calculatePassability( agent, Teammate ) // see P1
        if ( Pt > Pb ) then
               Pb <- Pt
               receiver <- Teammate
        end if
   end for

return receiver
```

The following section describes P1, the algorithm that computes the suitability of an agent to receive a pass, or what we term passability.

```
P1 Calculate Passability

   Input(s):   source - the agent who has possession of the ball
   target - the target player whose passability is to be calculated
   Output(s):  passability - a real number indicating pass suitability of target
   player

   posSource <- global position of source
   posTarget <- global position of target

   // draw a line between source and target
   Line L <- Line::makeLineFromTwoPoints( posSource, posTarget )
   sumOfDistances <- 0.0;

   // for each opponent, add their distance to the line to the
   // sum of distances
   for each Opponent
```

```
        oppDistToLine <- L.getDistanceWithPoint( position of Opponent )
        // only add opponents that are close to the line
        if ( oppDistToLine < 15.0 ) then
               sumOfDistances += oppDistToLine
        end if
   end for

   passability <- sumOfDistances

   // modify passability to favour forward passing
   if ( angle to opponent goal -
        angle to posTarget < 50 ) then
        passability *= 1.3
   else
        passability *= 0.4
   end if

   // modify for congestion
   if ( target is congested ) then
        passability *= 0.5
   end if
   if ( source is congested ) then
        passability *= 0.5
   end if
   // modify to prevent long passes
   if ( distance to target > 20.0 ) then
        passability *= 0.5
end if
```

This algorithm modifies the original UvA player algorithm and is used by all agents.

3.2 Team 2: Passing with Yells

This team (A2) is an implementation of the Active Design approach. Here every agent calculates its own passability using P1. This calculation is done for every cycle a teammate has control of the ball. The fastest player in a set who can reach the ball is determined to have control of the ball. Once the passability value is calculated, each player uses the 'say' command to communicate this number to teammates.

When updating the world model, every agent uses incoming aural messages from teammates to track the best passability at a given time. If a message arrives announcing a higher passability, then the sender of the message becomes the new best pass receiver. Every five cycles, the best passability is reset to the minimum threshold, to ensure that old information is not used to make the passing decision.

As in centralized passing, the goalie uses A1 to calculate the pass receiver, but unlike its teammates, the goalie uses the centralized approach with no input from teammates. This ensures that the goalie always passes to someone.

3.3 Team 3: Passing with Filtered Yells

This team (A3) is also an implementation of Active Design, but it has some properties of the Brooksian approach, in the sense that it takes into consideration the limitation of the communication channel, which is a significant property of the environment. A3 also uses P1, and in the same way as the A2 algorithm. However, instead of agents saying their passability every cycle, here agents listen to others' yells and compare their passability with the ones they hear. That is, they compare their passability with the current best value, and announce their passability only if it is better. This lowers the load on the communication channel, by allowing only the best messages through. Once again, the goalie uses the centralized approach to passing.

3.4 Algorithm Analysis

A1 uses P1 t times every cycle, where

$$0 \le t \le t_{\max} \text{ and } t_{\max} = (\textit{number of players on a team}) - 2$$

An agent will not pass to itself or a goalie. P1 itself is similarly bounded by a factor of the maximum number of opponents, therefore the complexity of A1 is at most proportional to the square of the number of players on one team. The complexity can be less because players with unknown positions (not known with a certain confidence level) will not be included in the passability calculation.

Agents in A2 and A3 use P1 once per cycle to calculate individual passabilities. Their complexity is therefore equal to the complexity of P1. However, this is mitigated by the fact that the player must wait for a constant number of cycles before determining the best pass.

4 Experiment

Each modified UvA team was pitted against the original UvA team to test the passing algorithms. Each team played 10 games. Logs of individual agents' decision-making were collected and analyzed to extract the successful and unsuccessful passes, and the passability values. Note that even though A1 uses centralized decision-making to pass, the other agents in A1 calculate their own passabilities and store these values. In effect, all agents in all the three conditions calculate their passabilities using P1 when a team mate has the ball. In A2 and A3, this information was 'yelled', and the passing agent's decision to pass was based *entirely* on this information. In A1, there was no yelling by individual agents, they just stored their passability values. The passing agent here used the P1 algorithm in a centralized manner, to calculate the passability for everyone else.

5 Results

5.1 Pass Completion

Pass completion is a measure of the ability of a player to pick the correct pass recipient. Although pass completion primarily depends on the effectiveness of the passability function P1, it can also show the relative effectiveness of the three algorithms with respect to each other. We analyzed the log files of games played with the three passing algorithms, and checked who next kicked or caught the ball after a player made a pass. If it was the intended recipient, the pass was completed, otherwise the pass failed.

Table 1 shows the results of running the three modified UvA teams against the original UvA team, and testing over ten games for each team. The centralized approach achieved the best results with A2 and A3 achieving a similar percentage of pass completion.

Table 1: Number of passes completed

Team	Total Passes	Passes Completed	Percentage
A1	2416	1110	45.94%
A2	2384	804	33.72%
A3	3455	1324	38.32%

5.2 Correct Passing

To understand the effectiveness of the three algorithms in deciding the right player, we determined the number of 'correct' passes, which is defined as passes where the agent in possession of the ball passes to the best player (the agent with the highest passability value). This view assumes that an individual player’s judgment of its passability is the ‘correct’ value. The validity of this view is tested in the next section.

For A1, this analysis provides a sense of how often the centralized algorithm agreed with the individuals’ assessment of their own passabilities. For A2 and A3, which depend entirely on the communication channel to decide on passing, a correct pass indicates that the message from the most suitable player got through to the passing agent. As the communication channel bandwidth is low, the agent may not always know the best passability in the team, and therefore will make an incorrect pass. Thus, for A2 and A3, the ratio of correct passes to incorrect passes reveals the effectiveness of the communication channel, irrespective of the performance of the passing algorithms.

The results (in Table 2) show that the centralized algorithm makes the ‘correct’ judgment around 39 percent of the time. A3, the filtered yell model, is more effective than A2 in allowing agents to know the teammate with the highest passability in any given cycle. Overall, the A3 strategy is better in choosing the player with the highest passability.

Table 2: Number of passes where the player with the highest passability was chosen

Team	Total Passes	Best player	Percentage
A1	2416	942	38.99%
A2	2384	803	33.68%
A3	3455	1454	42.08%

5.3 Correct and Completed Passes

The above tables give us our preliminary data. Given the narrow communication channel, there is no direct way of gauging the effectiveness of the epistemic structure strategy, because only a third of the signals get through to the decision-making agent. To understand the effectiveness of the ES strategy, we need to create a ‘what if’ scenario, i.e. what if all the messages get through? To find out this, we have to first find out the effectiveness of knowing the best player, that is what percentage of time did knowing the best player result in a completed pass? To get this value, we extracted the intersection of the two tables above, i.e. the completed passes when the best player was chosen. This is given in Table 3.

Table 3: Number of passes completed when the best player was chosen

Team	Passes completed	Best player chosen	Percentage
A1	510	942	54%
A2	344	803	42.8%
A3	668	1454	45.9%

Averaging for the three teams, this means nearly 48% of the time, when the best player (the player with the highest passability, according to his own estimate) is chosen, the pass is completed. This is quite impressive for a simple passing algorithm. This is the maximum advantage provided by the epistemic structure strategy.

What about the mirror of this, how many times was the pass completed when the player chosen was not the best player?

Table 4: Passes completed when the best player was not chosen

Team	Passes completed	Best player not chosen	Percentage
A1	512	1474	34.7%
A2	400	1581	25.3%
A3	563	2001	28.1%

This means knowing the passability value can provide around 20% improvement in passing for A1 and 18% for A2 and A3. Therefore the accuracy of individuals' calculation of their own passability is indeed better. This justifies the assumption we made about correctness above. Signaling of this task-specific structure can result in around 19% improvement in success rate overall.

At a higher level, this illustrates why signaling is a preferred strategy in making mating decisions – because an agent's judgment about his own system state is always better than another agent's judgment of the same state. Of course, the strategy is complicated by the problem of reliability of signals (mimicry) and 'eavesdropping', where other agents 'listen in' and use the signal to further their own interests.

Note that in the above table, the players receiving the pass are not randomly chosen. The passability algorithm is still being used, either its output is not the best player (for A1), or the best player's message does not reach the passing agent (A2 and A3). This means the passing agent may be passing to the next-best player, for instance. So the 19% improvement is a very conservative estimate. If we compare with a team that has no passability calculation at all (individuals' yells), then the epistemic structure strategy can provide a full 48% improvement, if a fair number of the yells get through.

The above analysis shows the improvement in performance for individual players. What about the team performance? To understand this, we extracted the number of times the home team got the pass when the best player was chosen.

Table 5: Home team got the ball when the best player was chosen

Team	Home team got ball	Best player chosen	Percentage
A1	653	942	69.3%
A2	560	803	69.7%
A3	976	1454	67.1%

When the best player was chosen, the ball stayed with the home team nearly 70% of the time. This is probably a result of the passability algorithm taking into account the number of opponent players in the pass trajectory, and the movement of agents towards the pass trajectory. Either way, knowing the passability calculations by individuals provide an overall advantage for the team.

The individual and team performance in retaining control of the ball (when individual passabilities are known) shows that the epistemic structure strategy is quite effective. The environment contributing to cognition provides significant advantages in improving accuracy.

6 Limitations and Future Work

This experiment looked at a single aspect of decision-making based on the ES strategy, the advantage in accuracy. In the next set of experiments, we are planning to investigate a second aspect of this strategy, the advantage in processing load, by measuring the time taken for calculations. The efficiency of a cognitive strategy is a combination of both accuracy and processing efficiency. We are interested in both, and also in the way they interact.

One of the major limitations of the study is the indirect way of assessing the effectiveness of the ES strategy. This is a direct result of the narrow communication channel. If the server parameters had allowed us to manipulate the hear_max value beyond 2 messages per cycle, we would've been able to judge the effectiveness of the strategy better. This would've also provided a way to better understand the relationship between channel-width and signal effectiveness in a dynamic environment. The freedom to change parameters, and a more user-friendly way of doing this, could lead to the RoboCup environment being used more widely by disciplines like cognitive science.

In this study, the opponent team was the same one for all the games. Even though this could be considered as providing a standardization for the results reported here, it is desirable to test a cognitive strategy in a variety of situations. Similarly, tests need to be done to determine the optimal number of waiting cycles used by a player in A2 and A3 before deciding on whom to pass. A further limitation is that the opponent team was not designed to intercept the passability messages, or to manipulate them. So the adversarial nature of the environment was limited to pass interception. In future work, we plan to use different teams against our teams.

We also plan to investigate how unreliable messages affect decision-making based on the ES strategy. This is the equivalent of mimicry in biological systems. Another interesting study would be to examine how centralized decision-making can be combined with ES-based strategies, and in what conditions such combinations are effective. Varying the noise parameters for different combinations of strategies may provide insight into how the structure of the environment can lead to different decision-making strategies.

From a purely RoboCup perspective, a serious limitation of our design is the variable performance in scoring goals. Even though the ES strategy provides significant advantage in maintaining control of the ball at the individual and team levels, this control does not translate into more goals. This is because our implementation focused on just the passing decision, resulting in teams obsessed with passing. They never consider shooting the ball as an option. That being said, we observed wide variability in the scoring of goals in A1 and A3. A2, being limited by the communication channel, mostly lost. We are working on redesigning the algorithms to improve the performance in scoring.

References

1. Stopka, P. and Macdonald, D.W. Way-marking Behaviour: An Aid to Spatial Navigation in the Wood Mouse (Apodemus Sylvaticus). BMC Ecology,(2003) Published online, http://www.biomedcentral.com/1472-6785/3/3

2. Henry, J.D. The Use of Urine Marking in the Scavenging Behaviour of the Red Fox (Vulpes Vulpes). Behaviour, 62:82-105 (1977)

3. Zahavi, A. and Zahavi, A. The Handicap Principle: A Missing Piece of Darwin's puzzle. Oxford University Press, Oxford (1997)
4. Bradbury, J.W. & Vehrencamp, S.L. Principles of Animal Communication. Sunderland, Mass: Sinauer Associates. (1998)
5. Susi, T. & Ziemke, T. Social Cognition, Artifacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artifact-mediated Collaborative Activity. Cognitive Systems Research, 2(4), 273-290 (2001)
6. Bryson, J.J. Cross-paradigm Analysis of Autonomous Agent Architecture. Journal of Experimental and Theoretical Artificial Intelligence, 12(2):165--190, (2000)
7. Brooks, R. A. Intelligence without Representation, Artificial Intelligence, 47:139-160. (1991)
8. Kirsh, D. Adapting the Environment Instead of Oneself. Adaptive Behavior, Vol 4, No. 3/4, 415-452. (1996)
9. Kirsh, D. The Intelligent Use of Space. Artificial Intelligence. 73: 31-68 (1995)
10. Chandrasekharan, S. and Stewart, T. Reactive Agents Learn to Add Epistemic Structures to the World. Carleton University Cognitive Science Technical Report 2004-01. (2004) http://www.carleton.ca/iis/TechReports/
11. Kitano, H,, Asada, M., Kuniyoshi, Y., Noda, I. and Osawa, E.: RoboCup: The Robot World Cup Initiative, Proceedings of IJCAI-95 Worksop on Entertainment and AI/Alife, 19-24, (1995)
12. Weiss, G. (ed.) Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, Mass. (1999)
13. The RoboCup Federation, http://www.RoboCup.org
14. Noda, I., Soccer Server: A Simulator for RoboCup, JSAI AI-Symposium 95: Special Session on RoboCup (1995)
15. Stone, P., Riley, P., and Veloso,M. The CMUnited-99 Champion Simulator Team. In: RoboCup-99: Robot Soccer World Cup III, Veloso, M., Pagello E., and Kitano, H. (eds.), Lecture Notes in Computer Science Vol. 1856, Springer Verlag, Berlin (2000)
16. Reis, L.P., and Lau, N. FC Portugal Team Description: RoboCup 2000 Simulation League Champion In: RoboCup-2000: Robot Soccer World Cup IV, Stone,P., Balch, T., and Kraetzschmar, G., (eds.), Lecture Notes in Computer Science Vol. 2019, 29-40, Springer Verlag, Berlin (2001)
17. Lau, N. and Reis, L.P. FC Portugal 2001 Team Description: Flexible Teamwork and Configurable Strategy. In: RoboCup-2001: Robot Soccer World Cup V, Birk, A., Coradeshi,S., Tadokoro, S., (eds.), Lecture Notes in Computer Science Vol. 2377, 515-518, Springer Verlag, Berlin (2002)
18. Reis,L.P., Lau, N., Oliveira, E. C. Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents. In: Balancing Reactivity and Social Deliberation in Multi-Agent Systems, Markus Hannebauer, Jan Wendler, Enrico Pagello editors, LNCS 2103, 175-197, Springer Verlag, (2001)
19. Stone, P. and McAllester, D. An Architecture for Action Selection in Robotic Soccer, Proceedings of the Fifth International Conference on Autonomous Agents (2001)
20. Jinyi, Y., Jiang, C., Yunpeng, C., and Shi, L. Architecture of TsinghuAeolus. In: RoboCup-2001: Robot Soccer World Cup V, Birk, A., Coradeshi,S., Tadokoro, S., (eds.), Lecture Notes in Computer Science Vol. 2377, 515-518, Springer Verlag, Berlin (2002)

21. Stone, P. and Veloso, M. Beating a Defender in Robotic Soccer: Memory-based Learning of a Continuous Fnction. In: Touretzky, D.S., Mozer, M. C., and Hasselmo, M. E. (eds.), Advances in Neural Information Processing Systems 8, pages 896–902, MIT Press, Cambridge, Mass, (1996).

22. Matsubara, H., Noda,I., Hiraki, K. Learning of Cooperative actions in Multiagent Systems: A Case Study of Pass Play in Soccer. In Adaptation, Co-evolution and Learning in Multiagent Systems: Papers from the 1996 AAAI Spring Symposium, 63–67, AAAI Technical Report SS9601, AAAI Press, Menlo Park, CA, (1996).

23. Kok, J.R. UvA TriLearn 2002 Source Code, University of Amsterdam (UvA), Faculty of Science Intelligent Autonomous Systems Group, (2002) http://carol.wins.uva.nl/~jellekok/RoboCup/2002/index_en.html