In Proceedings of the Sixth International Conference on Cognitive Modeling, 58-63.
Mahwah, NJ: Lawrence Earlbaum.

Why Soccer Players Yell: Using RoboCup to Model the Advantage of Signaling

Sanjay Chandrasekharan (schandra@sce.carleton.ca)
Babak Esfandiari(babak@sce.carleton.ca)
Institute of Cognitive Science,
Carleton University, Ottawa, Canada, K1S 5B6

Neal Arthorne(narthorn@connect.carleton.ca)
Department of Systems and Computer Engineering,
Carleton University, Ottawa, Canada, K1S 5B6

Abstract

We report an experiment where the RoboCup simulation environment was used to study the cognitive advantage provided by signals, which we view as *task-specific* structures generated in the environment to improve decision-making. We used the passing problem in RoboCup as our test problem and soccer-players' 'yells' of their 'passability' as the task-specific signals. We found that yells improved accuracy– agents using the yells to decide the best player performed much better than agents computing the best pass themselves. The accuracy advantage derives from the task-specific nature of the yell, and such task-specific structures (signals) are used by organisms across species. From this, we reason that player yells are an instantiation of an implicit, evolved, and adaptive strategy, rather than an explicitly reasoned-out process.

Many organisms generate stable structures in the world to reduce cognitive complexity, for themselves, for others, or both. 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 during foraging (Stopka & MacDonald, 2003). 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 (Henry, 1977, reported in Stopka & MacDonald, 2003). The male bower bird builds colorful bowers (nest-like structures), which are used by females to make mating decisions (Zahavi & Zahavi, 1997). Many birds advertise their desirability as mates using some form of external structure, like colorful tails, bibs etc (Bradbury & Vehrencamp, 1998). Other animals have signals that convey important information about themselves to possible mates and even predators (Zahavi & Zahavi, 1997).

Such *epistemic structures* (Chandrasekharan & Stewart, 2004), usually termed signals, form a very important aspect of animal life across biological niches. These structures allow the organisms to hive off part of their cognitive load to the world. How much cognitive advantage do such structures provide in noisy, dynamic and adversarial environments? Where do the advantages come from? What are its components? These are the problems we address in this paper. We used the RoboCup simulation environment to study the cognitive advantage provided by epistemic structure strategy.

Signaling is generally studied as communication (considered a good thing), and most approaches do not focus on the computational advantages provided by signaling. To understand the computational advantage provided by signaling, 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 (Zahavi & Zahavi, 1997). The tail reliably describes the inner state of the peacock, that it is healthy (and therefore has good genes).

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 imply 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 *perceptually* (i.e. without computation) 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 are *task-neutral* structures, *which have to be synthesized by the pea-hen into a task-specific structure*, to help with the mating decision. The tail, being a task-specific structure, allows the pea-hen to short-cut this synthesizing process. 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 its system state using its tail. Such self-description is one of nature's ways of avoiding long-winded sorting and inference.

The peacock example (and others above) shows that the reduction of others' cognitive complexity using task-specific external informational structures (what we term epistemic structures) is very common, and it can be considered one of the building blocks of nature. Note that while we develop the notion of epistemic structure using stable (or quasi-permanent) structures added to the world like markers and tails, task-specificity is a property common to all signals, including transient environment structures (like vocal signals for mating, warning etc.). In our view, transient signals are an adaptation of the basic epistemic structure theme (of adding task-specific structures to the world), to suit the highly dynamic or adversarial nature of such decision-making environments. In other words, stability is not the crucial property for being an epistemic structure/signal, task-specificity is. To understand the advantage provided by signaling, we have to understand the efficiency provided by task-specific structures.

A Taxonomy of Agent-Environment Relations

Even though signaling is a basic structure of cognition, it has received very little attention *as a cognitive strategy*. In the following section we develop a framework to understand how signaling, or the epistemic structure strategy (where the environment is changed in a way that it contributes task-specific structures for decision-making), fits in with other agent-environment relationships. We categorize agent-world relations into four strategies. To illustrate these strategies, we use the design problem of providing disabled people access to buildings. There are four general strategies to solve this problem.

Strategy 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.

Strategy 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.

Strategy 3: This involves adding structure to 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, the world is "doped", so that it contributes to the agent's task. Our analysis will focus on this approach.

Strategy 4: This strategy is similar to the first, 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 strategy III, we will ignore it in the following analysis.

The first strategy is similar to the centralized AI one, which ignores the structure provided by specific environments. The environment is something to be overcome, it is not considered a resource. This strategy 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 strategy is similar to the situated AI model promoted by Rodney Brooks (1991). This strategy recognizes the role of the environment as a resource, and analyses and exploits the detailed structure that exists in the environment to help the agent. Notice the environment remains unchanged, it is considered a given.

The third strategy is similar to one aspect of distributed cognition, where task-specific structures are generated in the environment, allowing the agent to hive off part of the computation to the world. Kirsh (1996) terms this kind of "using the world to compute" active redesign. This strategy underlies many design techniques to minimize complexity. At the physical level, the strategy 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 (Kirsh, 1995). Kitchens and personal libraries (which use locations as tags for identifying content) are instances of such use of space in cognition.

Another application of task-specific structures 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. The effort is to generate task-specific structure in an information environment (the Web) so that software and human agents can function effectively in it. This 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, like meta-tags in web pages. Such tagged objects can be easily recognized by agents fitted with RFID readers (for instance, robots in a recycling plant).

The epistemic structure strategy is applied at the social level as well, especially in instances involving trust. Humans add structures to 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 informal, structure we create include standardized ways of dressing, talking etc.

Using RoboCup to Study Epistemic Structure

Given this broad range of current and potential applications of this strategy, it is important to understand how the epistemic structure strategy works, how it evolved, how efficient it is compared to other strategies, in what conditions it works, and where it breaks down.

We have modeled organisms learning to add task-specific structures to the world, across generations and within their lifetime, using genetic algorithms and Q-learning (Chandrasekharan & Stewart, 2004). In both cases, we used an environment with no adversaries and almost no noise. The results show that the advantage of using the epistemic structure (ES) strategy is quite significant, agents spend 58% of their time generating such structures. But since most organisms live in noisy, dynamic, adversarial environments, static environments do not provide a sense of the comparative advantage of the ES strategy over others.

The RoboCup simulation environment, which simulates a soccer game, provides an interesting dynamic and adversarial environment to study the efficiency of the epistemic structure strategy. Briefly, the simulation environment consists of a standard central server and two teams of decision-making agents (usually 11 to a team) that connect to the server. Researchers around the world develop agent teams to study multi-agent systems, while the server is maintained by the RoboCup administration as a standard test-bed for such systems.

In a game, the server sends the agents field information (like current coordinates, coordinates of opponents or teammates seen, coordinates of the ball etc.). The agents use this information to update their world model, and analyze this information to send action commands back to the server (like kick, dribble, turn, turn_neck etc.), which are then 'executed' by the server, thereby changing the state of the agents and what they can perceive. This process also changes the configuration of the field in a very dynamic fashion, akin to a soccer game. The agents can communicate with each other using a narrow channel broadcast by the server (each agent can hear utmost 2 messages in a cycle), but every message by every agent is heard by every other agent, an approximation to player yells in a real soccer field.

Being a game environment, Robocup does not provide much scope to add *stable* task-specific structure to the environment. However, it does allow transient task-specific structure to be added, these are 'yells', or signals from teammates. We used this structure to study the advantage provided by task-specific structures, using the passing problem (i.e. how an agent in control of the ball can decide whom to pass the ball) as our test decision-making problem. For our study, we developed three RoboCup teams (11 agents each) that used three different approaches to passing. The teams were based on the publicly available UvA TriLearn 2002 team (Kok, 2002), which implements some basic low-level skills (like dribbling, kicking etc.).

Team 1: Centralized Passing

This team (A1) uses strategy 1 in our agent-world taxonomy. A1 does all computations centrally, and 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
```

The passability values used above are based on test games, where different pass situations and passabilities were tested. The algorithm modifies the original UvA player's decision-making algorithm and is used by all our agents.

Team 2: Passing with Yells

This team (A2) is an implementation of the epistemic structure 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 signal this value to teammates.

When updating the world model, every agent uses these 'yells' 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.

Team 3: Passing with Filtered Yells

This team (A3) is also an implementation of the signaling strategy, but it has some properties of the Brooksian approach, because 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 yelling 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.

Experiment

To test the efficiency provided by the task-specific structure strategy, our three teams were pitted against the original UvA team. 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.

Results

Since we are interested in understanding the performance of the strategies in making the passing decision, we analyze only the completion of passes, and not the goals scored, which is affected by many factors other than passing.

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 strategies, 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 our three 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	Completed passes	Total Passes	Percentage
A1	1110	2416	45.94%
A2	804	2384	33.72%
A3	1324	3455	38.32%

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 signaled structures 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, even though the signaling channel is feeble.

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

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

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, i.e. 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 -- 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. We consider this quite impressive for a simple passing algorithm. This is the maximum advantage provided when an agent uses task-specific structures for decision-making in the Robocup environment.

What about the mirror of this, how many times was the pass completed when the player chosen was *not* the best player? Table 4 below provides these values.

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%

Comparing with the earlier table, this means knowing the passability value (task-specific structure) can provide around 20% improvement in passing for A1 and 18% for A2 and A3. This means individuals' calculation of their own passability is more accurate than others' calculation of the same value. This justifies the assumption we made about 'correctness' above. Once computed, signaling of this task-specific structure to a decision-making agent can result in around 19% improvement in success rate overall, in a dynamic and adversarial environment like robocup.

At the strategy level, this illustrates why signaling is a preferred strategy in making mating decisions – *because an agent revealing his/her own system state is always more accurate than another agent's judgment of the same state*. We hypothesize that this is also the reason why soccer players yell. Anecdotal evidence (from soccer-players) indicates that yells are used heavily by players while making passing decisions, suggesting an implicit understanding of this accuracy advantage. Given that it is the task-specificity of the yell that is contributing to its accuracy and use, and this task-specificity is shared by signals across species, it is also reasonable to assume that yelling is not an explicitly thought-out process, but an instantiation of an evolved, general, adaptive strategy. Even novice and beginner players yell. In tasks or functions involving cooperation (like mating, or hunting in a group), adding task-specific structures about your system (status, intention, perspective) to the world has adaptive value.

On the receiver side, a decision-maker would be basing her decision on more accurate information when she is using such structures. Of course, this is complicated by the problem of judgement, reliability (mimicry in organisms) and 'eavesdropping', where other agents 'listen in' and use the signal to further their own interests. It is interesting to note that this complexity has not lead to organisms dropping the signaling strategy altogether. Instead, they have evolved ways to counter mimicry and eavesdropping. This means the adaptive advantage provided by signaling is quite high.

The above analysis shows the improvement in performance for individual players. What about 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, or the movement of the agents towards the pass trajectory. Either way, 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.

Limitations and Future Work

The current experiment investigated one aspect of decision-making based on the ES strategy, the advantage in accuracy, i.e. identifying the best player to pass. In the next set of experiments, we are planning to investigate a second aspect of the task-specific structure strategy, the advantage in processing load, by measuring the time taken for passability calculations by agents with different perspectives. 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 epistemic structure strategy. This is a direct result of the narrow communication channel. If the server parameters had allowed us to manipulate the number of messages the agents can hear to beyond 2 messages per cycle, it would have been possible to judge the effectiveness of the strategy better. The ability to vary the size of the communication channel would have also provided a way to better understand the relationship between channel-width and signal effectiveness in a dynamic environment. Such 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 and ethology.

In this study, the opponent team was the same in 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.

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.

References

Alcock, J. (1998). *Animal Behavior: An evolutionary approach*, Massachusetts, Sinauer Associates.

Bradbury, J.W. & Vehrencamp, S.L (1998). *Principles of Animal Communication*. Sunderland, Mass: Sinauer Associates

Brooks, R. A. (1991) Intelligence without Representation, *Artificial Intelligence*, 47:139-160

Chandrasekharan, S. and Stewart, T. (2004). Reactive Agents Learn to Add Epistemic Structures to the World. Carleton University Cognitive Science Technical Report 2004-01. http://www.carleton.ca/iis/TechReports/

Henry, J.D. (1977). The use of urine marking in the scavenging behaviour of the red fox (Vulpes vulpes). *Behaviour*, 62:82-105.

Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18, 513-549.

Kirsh, D.(1995) The Intelligent Use of Space. *Artificial Intelligence*. 73: 31-68

Kirsh, D. (1996). Adapting the environment instead of oneself. *Adaptive Behavior*, Vol 4, No. 3/4, 415-452.

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

Noda, I. (1995). Soccer Server: A Simulator for RoboCup, JSAI AI-Symposium 95: Special Session on RoboCup

Stopka, P. & Macdonald, D. W. (2003) Way-marking behaviour: an aid to spatial navigation in the wood mouse (Apodemus sylvaticus). *BMC Ecology*, published online, http://www.biomedcentral.com/1472-6785/3/3

Zahavi, A., & Zahavi, A. (1997). The Handicap Principle: A missing piece of Darwin's puzzle. Oxford: Oxford University Press