

Complexity Science and Representation in Robot Soccer

Jeffrey Johnson¹ and Blaine A. Price²

¹Design and Innovation Department and
The Open University, Milton Keynes, MK7 6AA, UK
j.h.johnson@open.ac.uk

²Computing Department
The Open University, Milton Keynes, MK7 6AA, UK
b.a.price@open.ac.uk

Abstract. Complexity science is characterised by computational irreducibility, chaotic dynamics, combinatorial explosion, co-evolution, and multilevel lattice hierarchical structure. One of its main predictive tools is computer-generated distributions of possible future system states. This assumes that the system can be represented inside computers. Robot soccer provides an excellent laboratory subject for complexity science, and we seek a lattice hierarchical vocabulary to provide coherent symbolic representations for reasoning about robot soccer systems at appropriate levels. There is a difference between constructs being human-supplied and being abstracted autonomously. The former are implicitly lattice-hierarchically structured. We argue that making the lattice hierarchy explicit is necessary for autonomous systems to abstract their own constructs. The ideas are illustrated using data taken from the RoboCup simulation competition.

1 Introduction

Robot soccer is an excellent laboratory subject for the emerging new science of complexity characterised by computational irreducibility, chaotic dynamics, combinatorial explosion, co-evolution, and multilevel lattice hierarchical structure. The earlier benchmark problem of computer chess also has many of these properties, with the exception of not being chaotic. Start a game of chess in a given position, compute each of a sequence of moves, and the result is always the same.

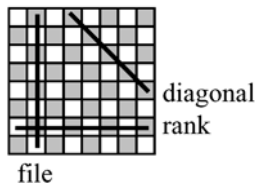


Fig. 1. Structured space in robot chess

Chess players use *structure* in order to play the game. Some of these structures are so fundamental that they have been given names such as the ranks, files, and diago-

nals illustrated in Figure 1. These names reflect spatial properties such as contiguity, and the functional properties of the pieces, e.g. rooks move on ranks and files, bishops move on diagonals, while kings and queens do both.

The *space* of the chessboard is divided into micro-units at the level of a player. These micro-units are aggregated into structures (e.g. ranks, files, diagonals) reflecting the modes of movement of the players. Thus the space of chessboard is hierarchically structured, with a well-defined set of areas, the squares, at the lowest level. Soccer and robot soccer do not have such an obvious lowest level of aggregation. At higher levels the space is structured by the goal area, the halves of the pitch, the penalty spots, the centre spot, and the centre circle, etc. All of these areas are defined because they play rolls in the rules.

In robot soccer the positions of the players and ball are assumed to be on an x - y grid. Even when the positions are represented by floating point numbers, this grid is finite. In this respect, the soccer pitch is like an enormous chessboard. The chessboard has 8 x 8 squares, while in robot soccer there are typically 1680 x 1088 pixels.

In chess, the players use a hierarchical *representation* that includes the ranks, files, diagonals and other more ambiguous areas such the right, left and centre of the board. Even though the number of squares and the number of these constructs is relatively small, chess is characterised by combinatorial explosion as chess players attempt to predict future system states. The 1.8 million pixels of the robot soccer pitch present an even more formidable combinatorial explosion in the way that the pixels can be grouped to form coherent and *relevant* areas of the pitch.

Our research is based on the premise that complex systems have hierarchically structured vocabulary reflecting the relational structure at micro- and macro-levels. By their nature, complex systems have to be investigated using computers, and this means that their representation must be explicit. In the case of robot soccer, this means there must be vocabulary for representing relationships between dynamically forming parts of the pitch, and dynamically forming relationships between players, opponents, and the ball.



N – kNight, K – King, R – Rook

Fig. 2. The Knight-Fork

Just as in chess there are ‘interesting’ structural relationships with names such as the ‘knight fork’, there are ‘interesting’ structural relationships in robot soccer. In Figure 2 the knight, N , checks the opponents king, K , and threatens the more valuable rook, R . This structure is so dangerous in chess that it has its own name, the ‘knight fork’. When players reason about chess, the ‘knight fork’ is an entity in its own right, with emergent properties not possessed by the individual pieces.

Figure 2 illustrates a similar well-known structure in soccer. In this case, defending player 3 threatens to take the ball from player-1. If player-1 feigns a pass to player-2, then player-3 must move to intercept that pass. In so-doing, player-2 moves out of position, and player-1 can slip past. This structure has its own vocabulary, e.g. player-2 ‘draws out of position’ player-3, allowing player-1 to pass.

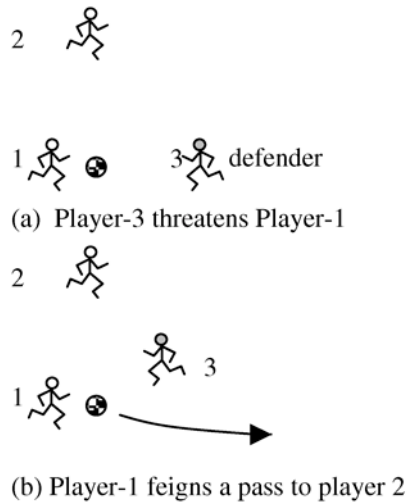


Fig. 3. Player 3 is drawn out of position by the relationship between players 1 and 2

The richer the vocabulary of these structures, the greater will be the advantage possessed by teams using that vocabulary. There are of course astronomic numbers of possible configurations of players and the ball on a soccer pitch. How can the ‘interesting’ structures be found? One answer to this is analyse many soccer games, and observe which configurations occur at ‘interesting’ times. These include the scoring of goals, but also include events such as the ball being lost, or even large areas of space opening up.

Early work in the analysis of RoboCup agents concentrated on the offline analysis of statistics gleaned from game logs, such as the work of Takahashi and Naruse [1] who measured statistics for teams at RoboCup 1997 such as number of goals, assists, kicks, own goals, and so on. Takahashi [2] continued this work and found no relationship between scoring and collaboration between agents when looking at the basic statistics as above. He did find, however, that collaborative actions, such as the number of 1-2 passes (player A passes to team-mate B in order to avoid defending player D, then B passes back to A once A has passed D), correlated highly with team ranking.

Tanaka-Ishii and colleagues [3] did a detailed offline statistical analysis of teams from 1997 with teams from 1998 with respect to 32 evaluation features, such as number of pass chains, average distance covered by one play, average pass length, and so on. They also compared the robustness of teams by replaying games with a reduced squad and found that some teams performed better with fewer players. They conclude that teams that perform poorly may not be the worst teams, but merely teams that have been let down badly by one aspect of their play. They argue for a collaborative modular team which can take the best performing parts of each team and also point towards the benefits of the online coach, which was introduced the following year.

Raines et al. [4] developed a system called ISAAC for post-hoc offline analysis of the events leading up to *key events*, such as *shots on goal* in the case of the RoboCup soccer simulation. ISAAC analyses the situations when the defence of the goal suc-

ceeds or fails with respect to a number of variables, such as the distance of the closest defender, the angle of the closest defender with respect to the goal, and the angle of the attacker from the centre of the field, the angle of the shot on goal and the force of the kick. The user is able to do a perturbation analysis to determine which changes in a rule will increase the goal success rate (e.g. changing the angle at goal, increasing the force of the kick). This enables analysing teams to seek improvements.

A similar off line approach is described by Wünstel et al. [5] who analysed player movement with respect to the ball to determine what kinds of movements a given player tends to make, although without reference to the context the player is in.

The online coach of Visser et al. [6] compared the formation patterns of opponent players from past games with a set of pre-defined formation patterns in an attempt to predict opponent formations. This allowed the coach to direct its own players to deal with the anticipated behaviour.

The online coach of Riley and Veloso [7] used pre-defined movement models and compared them with the actual movement of the players to predict future behaviour and advise its players accordingly.

Recent work in the analysis of agents in RoboCup has centred on predicting opponent behaviour. Kaminka et al. [8] used a system to identify and learn sequences of coordinated agent behaviour over one or more games for a given team. This was a post-hoc offline method which analysed logs of games after they were played. They ran experiments to show that the system was able to pick sequences that were characteristic of the team rather than arbitrary. Visser and Weland [9] developed a system that works on live games as opposed to a post-hoc analysis. Their system looks at the behaviour of the opponent goalkeeper as it leaves the goal as well as the passing behaviour of opponent players in order to find rules which characterise these agents. It updates these rules every 1000 cycles with the intention of making it available to the on-line coach to take advantage of the data.

Our work differs from that discussed above in our search for a coherent vocabulary through algebraic structures. This reflects our motivation in complexity science, and the desire to discover a methodology for representing complex systems in general, using robot soccer as a well defined, well researched, and replicable laboratory subject.

2 Complexity Science

Complexity science is characterised by computational irreducibility, chaotic dynamics, combinatorial explosion, co-evolution, and multi-level lattice hierarchical structure. Each of these suggests that predicting the future behaviour of complex systems will require relatively high levels of computation:

- computational irreducibility means that the computational load on making predictions is relatively high.
- deterministic chaos means that a high level of computation will be required for making useful predictions. A single point sample in the space of future possibilities has almost no useful information, and many future states have to be computed to gain information on the distributions of possible future system states.
- co-evolving systems tend to be both chaotic and computationally irreducible.

- combinatorial explosion, of its nature, implies high levels of computation for the search techniques used to explore large spaces of possibilities.
- complex systems are usually multi-level with micro- and macro-subsystems. The vocabulary to represent these systems should be coherent with respect to hierarchical aggregation, so that higher and lower level data are consistent.

This means that the representation of the system must be absolutely explicit to support computation. Paradoxically, many complex social systems are currently administered by people, with nearly all the computation and much of the data being in their heads. Such intuitive human processing supports many predictions in business and social administration. It even characterises human soccer. By comparison, the representation for robot soccer is explicit – it has to be because the system is autonomous and implemented on machines. For this reason we are interested in the simulation competition of RoboCup, since we believe that this will give new insights applicable more generally in complexity science and its applications.

3 The Lattice Hierarchy and Representation

Systems are characterised by wholes assembled from parts. This is illustrated in Figure 4, in which a set of three blocks is assembled to form a structure that we will call an *arch*. The arch is clearly more than the sum of its parts since it has emergent features, such as the possibility of walking through it on the path between α and β .

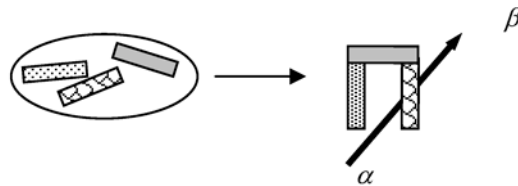


Fig. 4. The arch is a structure built from a set of parts, and has emergent properties not possessed by its parts

In Figure 4, the arrow labelled R indicates that the set is mapped to the whole by the relation R between the blocks. If the blocks are labelled a , b , and c , then the set is represented in the usual way by the notation $\{ a, b, c \}$. We will denote the R -structured set of blocks as $\langle a, b, c; R \rangle$. Then, $\{ a, b, c \} \neq \langle a, b, c; R \rangle$.

In Figure 5, the set of blocks is represented by an *Euler ellipse*, a variant of the Euler circle used to represent the set properties of intersection, subset, and union. The *hierarchical cone construction* then has the Euler ellipse as base and the name of the structure as its apex.

When analysing any system there is the problem of building a coherent representation between the highest level construct, 'the system', and the lowest level atoms such as the players or the pixels in simulated robot soccer. Generally there is a pre-existing vocabulary in vernacular language made up of terms that are more or less well defined. For example, in soccer the terms 'goal area' and 'Red's half' can be defined precisely, while 'the left wing' and 'the goal mouth' may be less well defined.

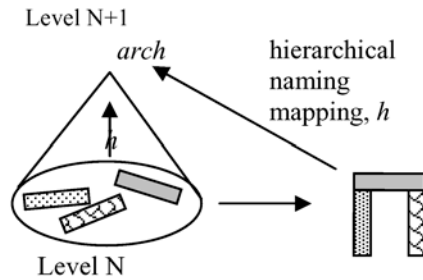


Fig. 5. The hierarchical cone construction

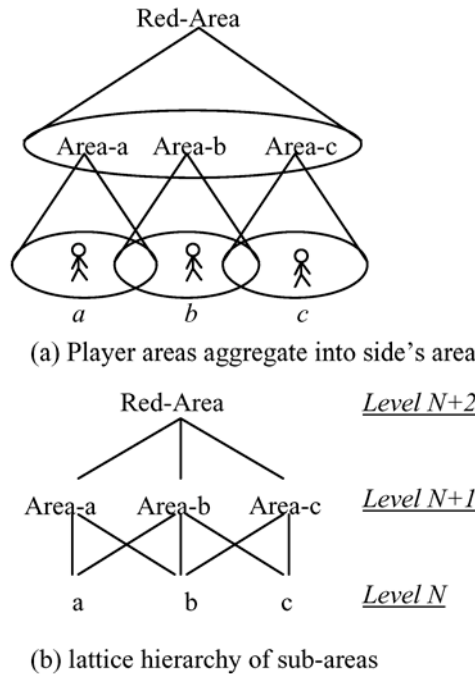


Fig. 6. A lattice hierarchy

Terms in the vocabulary may exist at many levels and the ‘set’ containing them will be called the *hierarchical soup*. As with the computer analysis of other complex systems, robot soccer has *The Intermediate Word Problem* of lifting a coherent hierarchically structured vocabulary out of the soup.

The term hierarchy is often misunderstood to mean a tree-like structure. More commonly, hierarchies have a ‘lattice’ structure, since things may aggregate into more than one structure at higher levels. Figure 6 provides a simple example in which three players define a subset of the pitch according to the pixels they are closest to. For example, the pixels closest to player *a* are shown by a Euler ellipse in Figure 6(a). Some pixels will be equidistant to some players, and so belong to both their areas, as shown by the intersecting ellipses for the players *a* and *b*, and the players *b* and *c*.

These areas have been given the names Area-a, Area-b, and Area-c. The union of these areas, together with that for all the other Red team members, makes up the part of the pitch controlled by the Red team, called the Red-Area. This simple hierarchy has three levels.

When constructing hierarchical vocabularies, often there is no obvious bottom level. Also it is sometimes necessary to define new structures between existing levels, thereby creating new levels. For this reason, levels are usually given the denotation *Level N+k*, emphasising the relative nature of the levels (Fig. 6(b)).

Figure 6(b) shows a graph of the ‘part-of’ relation implicit in the assembly structure in Figure 6(a), illustrating the notation of lattice. A *lattice hierarchy* is defined to be a class of objects with a *part-of* relation. This is *anti-symmetric*, so that *x part-of y* implies *not y-part-of x*. This is a *quasi-order* on the class, and in graph theory its graph is called a *lattice*. The quasi order is weaker than a partial order, since in a partial order $x \leq y$, or $y \leq x$, or both (equality). For example, there is no line between Area-a and Area-c in Figure 6(b) in the part-of relation between Levels N and N+1.

4 Structure in Robot Soccer

We have experimented with a number of RoboCup games, investigating ‘interesting’ relational structure.

One kind of structure concerns passes between players. A pass, $\langle \text{player-1, player-2}; R_{\text{pass}} \rangle$ is a pair of players and a relation between them. A set of passes between player of the same side is clearly an interesting structure in soccer. Here, a pass is structure between a pairs of players, and the pass-sequence is a structure on the set of passes.

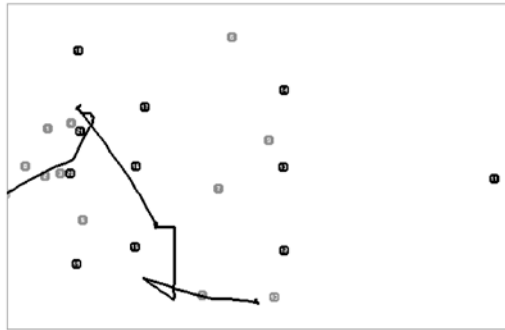


Fig. 7. A set of passes as a structure

Figure 7 shows a particularly long sequence of passes, which results in a goal being scored. The path forms as a consequence of the movements of the players, both on and off the ball, and the relationships that this creates between them.

Figure 8(a) shows a ‘nearest opponent’ relationship between the players. As can be seen, the graph has five components, reflecting the interactions between the players.

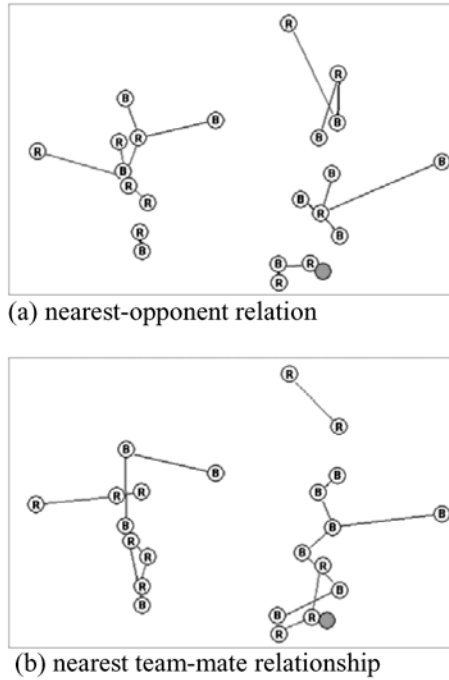


Fig. 8. Relational structure in robot soccer

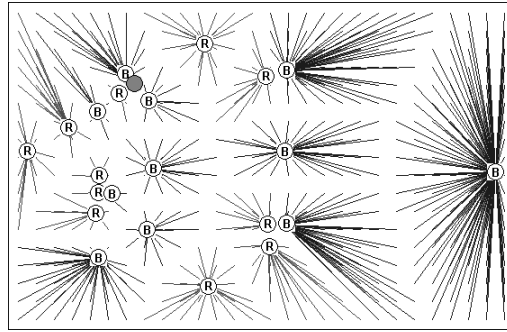
Figure 8(b) shows a ‘nearest team-mate’ relationship. Inspection of the graph shows that Blue’s structure is most highly connected, with two components. By comparison, Red’s structure has four components. In some sense, Blue’s structure hangs together better than Red’s structure.

Related to these structures, there are relations between parts of the pitch and the players. The parts of the pitch are squares, where sets of pixels make up the squares, and the set of squares cover the whole pitch.

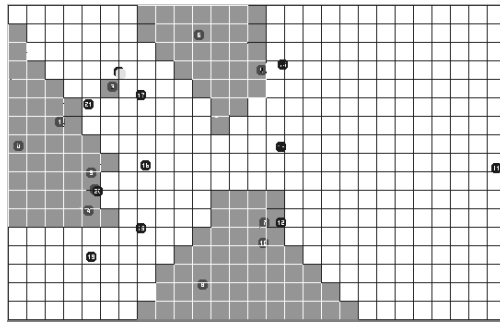
Figure 9(a) shows a relation between the players and the pitch squares. Figure 9(b) shows how the blue team, (B), dominates the game by owning almost all the pitch. This resonates with positional chess, when the players are not seeking tactical material advantage, but seeking to control the board. Here it can be seen that the blue team owns almost all the pitch. This whole game showed a similar dynamic pattern, with the blue area rapidly growing after the kick-off. Not surprisingly the blue team won by many goals.

5 Construct Formation

Each of the names or words in a lattice hierarchy vocabulary represents a structure. In most of the programs that people craft, the constructs exist *a priori* in the human mind. One of the goals for intelligent systems is to have them abstract their own constructs from their interaction with their environment, to create their own vocabulary.



(a) the relation between players and the pitch



(b) the pitch structured by the teams

Fig. 9. Spatial relational structure

The lattice hierarchical structure is potentially a *meta-representation* for this process. With this architecture, autonomous systems can investigate structured sets and keep information on those that are ‘interesting’. We would argue that the relational structure underlying the lattice hierarchy will characterise any vocabulary, and therefore the lattice hierarchy will be fundamental in automatic construct abstraction.

6 Conclusions

In this paper we have defined lattice hierarchies as fundamental structures in complex system. For us, the intermediate word problem is fundamental in robot soccer. We have investigated relational structure in simulated robot soccer games, and shown how it fits into lattice hierarchies. The ultimate goal of the research is to have the lattice hierarchical vocabulary emerge automatically as the robots interact with their environment. This would make a significant contribution to complexity science and its application in other areas.

References

1. Takahashi, T. and T. Naruse, From Play Recognition to Good Plays Detection: Reviewing RoboCup 97 Teams from Logfile, in RoboCup-98: Robot Soccer World Cup II. 1998, Springer-Verlag. p. 187-192.

2. Takahashi, T., LogMonitor: From Player's Action Analysis to Collaboration Analysis and Advice on Formation, in RoboCup-99: Robot Soccer World Cup III, M. Veloso, E. Pagello, and H. Kitano, Editors. 2000, Springer-Verlag. p. 103-113.
3. Tanaka-Ishii, K., et al., A Statistical Perspective on the RoboCup Simulator League: Progress and Prospects, in RoboCup-99: Robot Soccer World Cup III, M. Veloso, E. Pagello, and H. Kitano, Editors. 2000, Springer-Verlag. p. 114-127.
4. Raines, T., M. Tambe, and S. Marsella, Automated Assistants to Aid Humans in Understanding Team Behaviours, in RoboCup-99: Robot Soccer World Cup III, M. Veloso, E. Pagello, and H. Kitano, Editors. 2000, Springer-Verlag. p. 85-102.
5. Wünnstel, M., et al., Behavior classification with self-organizing maps, in RoboCup 2000, Robot Soccer World Cup IV, P. Stone, T. Balch, and G. Kraetschmar, Editors. 2001, Springer-Verlag: Berlin. p. 108-118.
6. Visser, U., et al., Recognizing Formations in Opponent Teams, in RoboCup-2000: Robot Soccer World Cup IV, P. Stone, T. Balch, and G. Kraetschmar, Editors. 2001, Springer-Verlag.
7. Riley, P. and M. Veloso, Recognizing probabilistic opponent movement models, in RoboCup-01, Robot Soccer World Cup V. 2002, Springer-Verlag.
8. Kaminka, G., et al. Learning the Sequential Coordinated Behavior of Teams from Observations. in RoboCup 2002. 2002. Japan.
9. Visser, U. and H.-G. Weland. Using Online Learning to Analyze the Opponents Behavior. in RoboCup 2002. 2002. Japan.