Open Research Online

The Open University's repository of research publications
and other research outputs

Representing Patterns of autonomous agent dynamics in multi-robot systems

Conference or Workshop Item

How to cite:

Johnson, Jeffrey and Price, Blaine (2003). Representing Patterns of autonomous agent dynamics in multi-robot systems. In: International Journal of Human-Computer Studies, 63(1-2) pp. 228–253.

For guidance on citations see FAQs.

© [not recorded]

Version: [not recorded]

Link(s) to article on publisher's website:
http://mcs.open.ac.uk/bp5/papers/AROB2003/AROB-2003-Johnson.pdf

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data policy on reuse of materials please consult the policies page.

oro.open.ac.uk

Representing Patterns of Autonomous Agent Dynamics in Multi-robot Systems

Jeffrey Johnson and Blaine Price
Department of Design and Innovation & Department of Computing,
The Open University, MK7 6AA, England
j.h.johnson@open.ac.uk b.a.price@open.ac.uk

Abstract

It is proposed that vocabularies for representing complex systems with interacting agents have a natural *lattice hierarchical* structure. We investigate this for the example of simulated robot soccer, using data taken from the RoboCup simulation competition. Lattice hierarchies provide symbolic representations for reasoning about systems at appropriate levels. We note the difference between relational constructs being human-supplied versus systems that abstract their own constructs autonomously. The lattice hierarchical representation underlies both.

1. Introduction

The concept of *autonomous agents* provides an abstraction that covers both synthetic agents such as robots and biological agents, such as plants, animals, and humans. In particular it provides an abstraction that enables us to study human behaviour with a new perspective on planning and managing *socio-technical systems*.

These includes mundane systems that appear to be unpredictable, including urban and regional settlements and their transportation infrastructure, organisations and their management, and less mundane systems such as drug trafficking, organised crime, and terrorism.

There are many ways of defining agents. Ours is based on combinatorial mathematics. The basic ideas are *sets* and *relations* between sets. This includes the special case of many-one relations between sets called *mappings*, or *functions* when the sets are numbers. Agents are *represented by constructive algebraic structures*. In other words, agents are built *bottom-up*.

This is consistent with a widespread view in complexity science. What we propose that is new is how to move up the constructive algebraic hierarchy to build higher level constructs that allowing *symbolic reasoning*.

An essential feature of real systems is that the hierarchy of representation is rarely tree-like. At all level there are connections between things, requiring a *lattice-like* organisation.

Building hierarchical vocabularies takes place in the context of combinatorial explosion. Even small sets have large numbers of subsets. A set with a dozen elements has over four thousand subsets, and there are millions of ways of selecting an 11-player soccer team from a class of thirty students.

It is easy, in theory, to generate *all* combinations of anything. The research challenge addressed by this paper is how can *useful* structures be abstracted from this wealth of latent possibilities. It will be seen that this amounts to the question :

"how can we build a lattice-hierarchical vocabulary to *represent*, *plan*, *design*, and *control* the emergent dynamics of systems of interacting autonomous agents?".

In Section 2 we will develop the mathematical notation under-pinning our approach. But we do not expect mathematics by itself to give answers to this question. We believe that *computation* is another essential ingredient in complexity science.

In Section 3 we will illustrate our theory and ideas using the simple illustrative example of simulated soccer-playing robot agents. Although simple in concept, this system can generate great complexity. Understanding and controlling this system has attracted the attention of some of the world's best and most advanced researchers.

In Section 4 we discuss pattern recognition and the lattice hierarchy, and the possibility of automatic construct abstraction.

Section 5 presents our conclusions, including the lessons to be learned from the study of soccer-playing agents, and how this might generalise to other systems.

2. Mathematical Preliminaries

Let S be a *system*. S has a set of *primitives*, $X = \{ x_1, \dots , x_n \}$. A *relation*, r, on X, is defined by a proposition p_r: $(x_{r0}, x_{r2}, \dots x_{rm}) \rightarrow \mathbf{T}$, where $\mathbf{T}$ is a *truth set*, and $\{ x_{r0}, x_{r2}, \dots x_{rm} \} \subseteq X$. For simplicity in the first instance, let $\mathbf{T} = \{\text{True}, \text{False}\}$. This can be extended to include probabilistic logic and fuzzy logic. We will say that the elements $x_{r1}, x_{r2}, \dots x_{rm}$ are *r-related* if and only if p_r: $(x_{r1}, x_{r2}, \dots x_{rm}) =$ True. When this holds we will say the system contains the object $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$, which will be called a *simplex*. By an abuse of language, we will write r:$\{ x_{r0}, x_{r2}, \dots x_{rm} \} \rightarrow \langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$, and say the relation r *maps* the set $\{ x_{r0}, x_{r2}, \dots x_{rm} \}$ to the simplex $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$.

If the primitives are said to exist at *Level N*, then the simplex $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$ will be said to exist at *Level N+1*.

Sometimes the object $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$ has a *symbolic name*. Let *Names* be the set of symbolic names. Then there is a *naming mapping*, n: $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle \rightarrow$ name. The mappings r and n can be combined in a way that maps the set directly to the name of a structured object, nr: $\{ x_{r0}, x_{r2}, \dots x_{rm} \} \rightarrow$ name. This notation is useful in drawing simplified diagrams, but it also allows that the structure $\langle x_{r0}, x_{r2}, \dots x_{rm} ; r \rangle$ can always be constructed, given the set and relation.

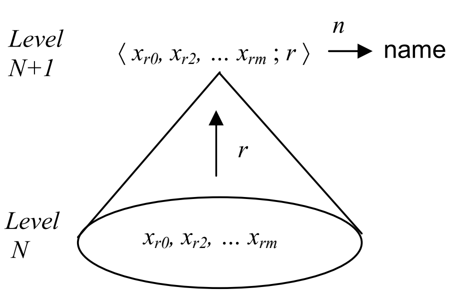


Figure 1. A set of part relationally mapped to a named simplex

The simplex, as structure, may have *emergent properties* nor possessed by its component elements. These properties may be structural. For example, an aeroplane can fly, when none of its parts can, and a choir can sing a harmony that individuals cannot. Emergent properties may also be numerical. For example, a car can have acceleration, while its parts cannot.

The generality of hierarchical structure is that parts are assembled into wholes, and wholes are assembled into higher level wholes. It is also possible for parts to be shared between higher level wholes, as illustrated in Figure 2. In this case the multi-level structure is called a *lattice hierarchy* (Figure 2).

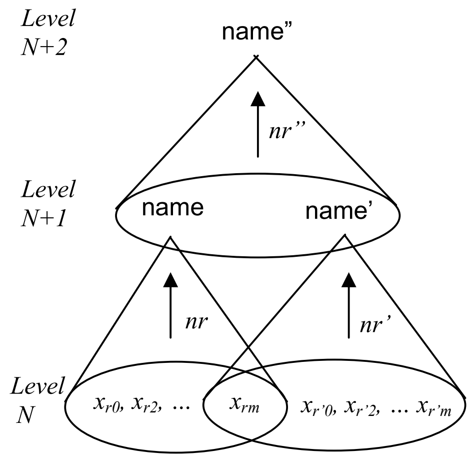


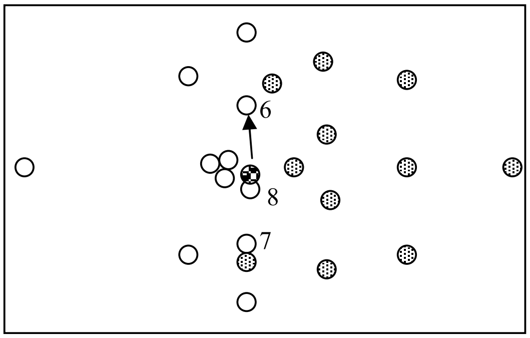
Figure 2 A multilevel lattice hierarchy

3. An example: soccer-playing robot agents

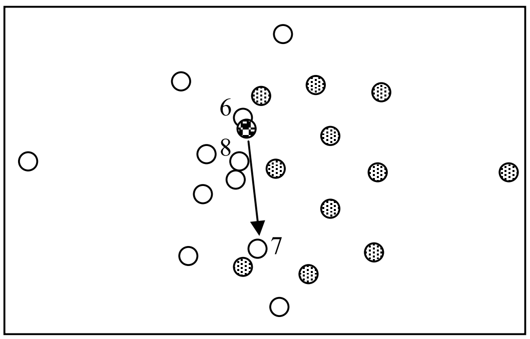
Robot soccer has taken over from computer chess as a benchmark problem in AI, Artificial Life, and Robotics. The International RoboCup Federation has encapsulated the challenge as “having a team of humanoid robots beat the human world champion soccer players by 2050”. This challenge unpacks into many engineering challenges related to materials, electronics, sensing, bio-engineering, power and control. It also unpacks into the challenge of devising tactics and strategy for soccer-playing robot agents.

One of the RoboCup competitions involves a soccer simulation in which teams can control the actions of simulated players, subject to their imperfect local perception of the pitch, other players, and the ball. The great contribution made by RoboCup is that all participants must share their research findings, and in recording the many simulation games that have been played.

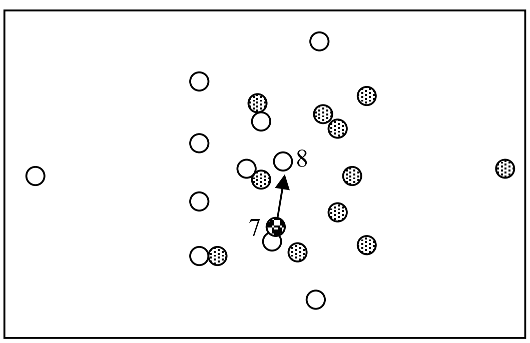
This is the database used for the research outlined in this paper. In the simplest case, the record for a single game is six thousand sets of twenty-three x-y co-ordinates, giving the positions of the twenty-two players and the ball. Each set corresponds to one tenth of a second, with games running for two halves of five minutes each.



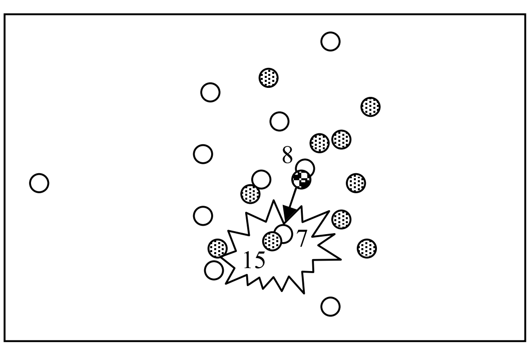
(a) Player 8 passes the ball to player 6



(b) Player 6 passes the ball to player 7



(c) Player 7 passes the ball to player 8



(d) Player 8 passes the ball to player 7, but it is won by the nearby player 15

Figure 3 Ball-passing relations

There is a lot of other information available, but this simple subset is sufficient for our purposes here. It is easy to write a computer program to display these data, and replay them as animations.

Following our methodology, there is a well-defined set of agents, namely twenty-two players and the ball. These can be distinguished as belonging to three subsets, the *red team* (agents 0-10), the *blue team* (agents11-21), and the ball (agent 22).

The methodology then suggests that we seek 'interesting' relationships between the agents to form higher level constructs that will be useful in reasoning about the tactics and strategy of the game.

What makes an 'interesting' relationship in this system? Presumably a pass from one member of the team to another is interesting. Figure 3 shows a sequence of passes, from robot 8 to 6, from 6 to 7, from 7 to 8, and from 8 back to 7, who then loses the ball to the opposition robot 15.

In our terms, each pass can be represented as a relationship between two robots, for example, ⟨ robot-8, robot-6; r_{pass} ⟩, which can be simplified to ⟨ 8, 6 ⟩ . The following table can be constructed from the data file for this game.

clock tick	robot acquiring ball	pass
42	8	
54	6	⟨ 8, 6 ⟩
66	7	⟨ 6, 7 ⟩
74	8	⟨ 7, 8 ⟩
81	7	⟨ 8, 7 ⟩
82	15	

Table 1. Possession and ball passing

Apart from passes between two robots being 'interesting', sequences of passes are also potentially interesting. A *passing sequence* can have one of three outcomes: an opposing player wins the ball, the ball goes out of play and the opposition win control of the ball, or a goal is scored. Each of these can be considered to be 'interesting' *events*.

In previous AROB papers (see references) it has been argued that structural events like these can be important in devising tactics and strategies for team robot behaviour. In particular, sequences of passing events characterise human football, as defenders are 'drawn out of position' by combinations of attacking players. Thus in terms of representation, we have individual players at Level N, pairs of player related by passing at Level N+1, and sequences of passing pairs at level N+2, as shown in Figure 4.

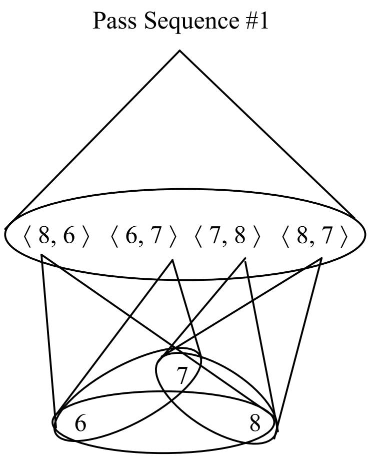


Figure 4 A lattice hierarchy of ball-passing

The game quoted in Figure 3 and Table 1 above was the 2000 RoboCup Simulation Final, won 1-0 by Portugal (the 'red team'). A list of the numbers of passing events for the first half of this game is given below.

Run Sequence # passes	Red Team	Blue Team
7	1	0
5	1	1
4	2	0
3	3	3
2	5	5
1	11	8

Table 2. Ball-Passing Sequence Events

From the table it can be seen that the red team dominates the game in terms of having the most and the longest passing sequences. Certainly when one watches this game, the red team seems to dominate. As in human football, the ability to pass accurately is very important in robot soccer. It is also important that the ball is passed to a player who is in a 'good position' to do something with it.

The leads to another canonical relationship between the soccer robots, the 'closest to' relation. At any tick of the clock it is possible to compute the distances between the robots, and for each find that which on the other side is closest.

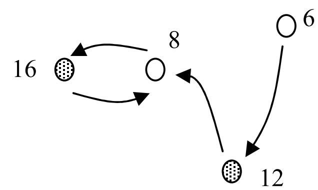
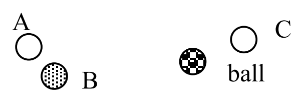


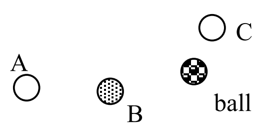
Figure 5 The closest opposing robot relation

This relationship is generally not symmetric. For example above, robots 8 and 16 are mutually closest to each other, but robot 12 is closest to robot 8, while robot 6 is closest to robot 12.

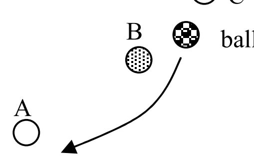
In human soccer, defenders are sometimes given the task of *marking* another player. This means that they have to stay close to that player. This usually results in a symmetric 'closest to' relationship between the two player, that can be very frustrating for the attacking player. Unless the defender can be 'shaken off', by interaction with other team members. In other words, a player forced in to the structure ⟨ me, marking opponent ⟩ might seek to form a structure with ⟨ ball, team-mate⟩ in order to break the me-marker relationships, as illustrated in Figure 6.



(a) A is closely marked by B



(b) B is drawn to C and the ball



(c) A has shaken off B

Figure 6 Shaking off a close marking opponent

4. Pattern Recognition and the Lattice Hierarchy

The previous section illustrated the formation of structures representing constructs such as 'passing' or 'being closest. To find examples of these in a data set it is necessary to *recognise* the defining *pattern*. This means:

(i) appropriate sets of candidate elements have to be recognised for the relation r
(ii) the relationship r has to be tested between the elements of the candidate subsets

For example, in recognising 'closest to' structures, it is necessary to generate all the pairs (A,B) where A belongs to one team and B belongs to the other. Then it is necessary to test to see if A is closest to B. Although this appears to be a binary relation, it is more than this. To decide the closest opponent to A requires that *all* the opponent pairs are formed and the emergent property of distance(A,B) be calculated. The particular $\langle$ A, B; r_{closest} $\rangle$ is therefore recognised by inspecting the whole set of pairs, (A,B). Thus $\langle$ A, B; r_{closest} $\rangle$ is higher in the lattice hierarchy than the pair $\langle$A, B$\rangle$; where A is a red robot and B is blue.

In terms of implementation, (i) usually involves forming lists, where each element can be tested independently of the others. It just has to be tested to see if it has the required properties. On the other hand, (ii) can be much more demanding, since it may require that relationships between all the elements be tested simultaneously. This may involve complicated functions to build constructs and test them.

One of the great objectives in building intelligent agents is to have agents that can structure the universe for themselves, by abstracting their own constructs. Although there are combinatorially many subsets, generating testing any given subset is not always a computationally demanding task, especially when notions of sampling are used. By contrast, the possibility of generating and testing 'useful' relationships is much more onerous.

In the first instance we are analysing the RoboCup data using human-inspired constructs such as the pass-sequences and closest-opponent as discussed above. However, the data lend themselves to more open-ended experiments in automatic construct generation. In our terms, means generating the lattice hierarchy and generating a *vocabulary* to name it structures the various emergent levels. It also means generating automatic pattern recognisers to test elements and relationships.

Conclusions

The lattice hierarchy is, arguably, one of the simplest structures available for representing complex systems with hierarchical vocabularies. To use this symbolic representation it is necessary to have pattern recognisers between the levels, able to determine higher level structures. Hierarchical aggregation through relatively simple substructures is likely to be computationally the most tractable. Ultimately, synthetic systems must abstract their own constructs and vocabularies. We believe that in all cases, the lattice hierarchy will implicitly or explicitly be used.

These ideas generalise to other complex system, including the social systems mentioned in Section 1. Although robot soccer, even in its simulated form, is difficult to control successfully, it is much simpler and better behaved than many human systems. For this reason we think it is an appropriate laboratory subject for understanding better the nature and use of lattice hierarchies.

References

RoboCup Federation, (www.robocup.org) Proc. RoboCup Conferences, 1998, 1999, 2000, 2001, 2002.

Johnson, J.H., Sugisaka, M., 'Desirable-Event subgoal theory in the design of swarm robot control systems, AROB-7, Japan, Jan 2002

Johnson, 'Visual communication in swarms of intelligent robot agents', Artificial *Life and Robotics*. 2001

Johnson, J.H., Sugisaka, M., 'Complexity Science for the design of swarm robot control systems', *Proc. IECON-2000*, IEEE Conf. on Industrial Electronics & Control, 2000

Johnson, J.H., Sugisaka, M. 'Culture and communication in the design of swarm robot control systems', Int. Conf. Advanced Systems Engineering, Coventry University, 2000.

Johnson, J.H., 'Robot football, artificial life, and complexity', *Artificial Life and Robotics*, **3**, 230-235, 1999