Social Embeddedness and Agent Development

Bruce Edmonds
Centre for Policy Modelling,
Manchester Metropolitan University,
Aytoun Building, Aytoun Street, Manchester M1 3GH.
b.edmonds@mmu.ac.uk http://www.cpm.mmu.ac.uk/~bruce
Tel. (0161) 247 6479 Fax. (0161) 247 6802

An Extended Abstract for UKMAS'98
This is a conflation of [7, 9], applied to the issue of agent development.

1 Introduction – Engineering and Social Simulation Goals

Entities that are meaningfully described as artificial 'agents' may be used by humans for many different purposes. Some of these purposes can be grouped by the abstract goals they are designed to fulfil. Two such goals are: *to construct systems that meet certain performance criteria in a reliable way*, which I will call the 'engineering' perspective and *to act as models of social agents so to increase our understanding of them*, which I will call the 'social simulation' perspective. Both of theses goals are valid but will entail some differences in their methods.

The 'engineering' perspective generally means that the performance criteria come first (in any particular loop of the design cycle), then systems are constructed to meet those criteria. This is usually done with an eye to a range of such systems, in which case more general *methods* are developed so that particular systems can be reliably produced as and when they are needed. Using agents as the essential components of such a process was first suggested by Shoham [17], this follows a general trend in software engineering towards the use of increased abstraction [19]. A critical element in engineering systems is that the results must be *reliable*, because people want to use them as a component tool in the execution of their plans. Two of the chief ways in which such systems are made reliable is via *predictability* and *transparency.* That is, the results of ones design decisions must be predictable, so that you can work out the consequences of your actions before you actually use the system and the nature of ones design decisions must be fairly clear, that is there must be a way to guess at design decisions without a full computational prediction of the results for every step.

In contrast to the above, the 'social simulation' perspective one may start with a specification of the agent's mechanisms and structure and then observe the resultant emergence in behaviour and end-result. The researcher often uses the simulation to *explore* the possible behavioural outcomes. The interest in such simulations is often precisely because the resultant behaviour is surprising, in other words that it is *not* transparent. Frequently the results of such simulations are not even predictable. For this reason results and methods of researchers of these two perspectives are sometimes mirror-images of each other (in the sense of being opposite) – social simulators are often aiming to create *exactly* the type of situation that software engineers are trying to prevent.

This paper aims to characterise a feature of such systems of inter-acting agents that distinguishes between the two approaches, namely *social embeddedness*. It is argued that this is an essential feature of societies as we know them and has practical consequences for the agents that inhabit them. For this reason it is suggested that such embeddedness will need to be a feature of many social simulation models. A consequence of social embeddedness is that it may not be practically possible for its component agents to be designed using the 'engineering' perspective (as usually conceived of at present). Of course there are many areas of overlap between these two perspectives in terms of methodologies, tools and ideas and in real-life different perspectives may be taken at different times and for different aspects of a project but I do not have room to consider these here. Neither am I attacking in any way the legitimate development of techniques and methodologies for engineering systems out of agents, but merely pointing out that fully social agents might not sit well in such a project.

2 Social Embeddedness

2.1 Characterising Social Embeddedness

In attempting to elucidate the concept of ‘social embeddedness’, one faces the problem of where to base one’s discussion. In sociology it is almost an assumption that the relevant agents are ultimately embedded in their society – phenomena are described at the social level and their impact on individual behaviour is sometimes considered. Cognitive science has the opposite perspective – the individual’s behaviour and processes are primitive and the social phenomena may emerge as a *result* of such individuals interacting.

This split is now mirrored in the world of artificial agents. In traditional AI it is the individual agent’s mental processes and behaviour that are modelled and this has been extended to considerations of the outcomes when such autonomous agents interact. In Artificial Life and computational organisational theory the system as a whole is the starting point and the parts representing the agents tend to be relatively simple.

I wish to step back from disputes as to the extent to which people (or agents) *are* socially embedded to one of the appropriateness of different types of models of agents. From this view-point, I want to say that an agent is *socially embedded* in a collection of other agents to the extent that it is more *appropriate* to model that agent as part of the total system of agents and their interactions as opposed to modelling it as a single agent that is interacting with an essentially unitary environment. It contrasts modelling agent interaction from an internal perspective (the thought processes, beliefs etc.) with modelling from external vantage (messages, actions, structures etc.). This is illustrated below in figure 1.

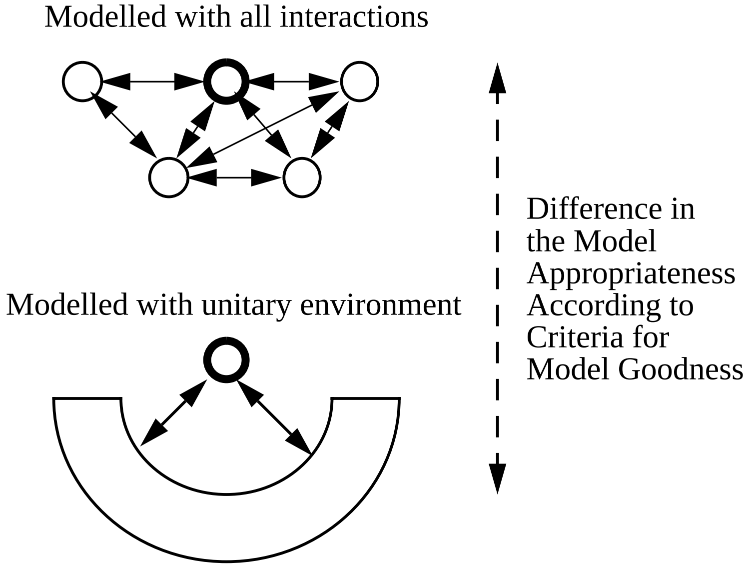

Figure 1. Social embeddedness as the appropriate level of modelling

Notice that criteria for model acceptability can include many things other than just its predictive accuracy, for example: *complexity* [4]. The modelling framework is indispensable; for example, an agent may not be at all embedded from an economic perspective but very embedded from the perspective of kinship relations.

Let us take a three examples to make this a little clearer.

- Consider first an economic model of interacting agents where each of these agents individually has a negligible effect on its environment, which would mean that a model of the whole system could be easily transformed into one of a single agent interacting with an economic environment. Here one would say that each agent was not socially embedded since there is little need to model the system as a whole in order to successfully capture the agent's behaviour.
- Next, consider an intermediate case: an agent which interacts with a community via a negotiation process with just a few of the other agents. Here a model which just considers an agent, its beliefs and its interaction with these few other agents will usually provide a sufficient explanation for all that occurs but there may still be some situations in which interactions and causal flows within the whole community will become significant and result in surprising local outcomes, but these may be controlled for using good software design practice. Here one could meaningfully attribute a moderate level of social embeddedness.
- Finally, consider the phenomena of fashion. Something is fashionable only if considered so by a sizable section of a population. Although there is some grounding of the selection of particular styles in the climate, economic mood etc. this is tenuous. Fashion is largely a self-producing social phenomena; the *particular content* of fashion is contingent – it has little immediate connection with an individual's needs, otherwise fashions would not vary so greatly or change so rapidly. A model of how fashions change which took the form of an individual interacting with a *unitary* social environment would not capture much of the dynamics. Here we have a high level of embedding – fashion is an essentially social phenomena, so it is appropriate to model it at this level.

At first sight this seems a strange way to proceed; why not define social embeddedness as a property of the system, so that the appropriate modelling choices fall out as a natural result? I am using artificial agents to model real social agents (humans, animals, organisations etc.), and so it is not enough that the outcomes of the model are verified and the structure validated (as in [15]) because I also wish to characterise the emergent process in a *meaningful* way – for it is these *processes* that are of primary interest. When observing or modelling social interaction this meaning is grounded in the modelling language, modelling goals and criteria for model acceptability (this is especially so for artificial societies). The validation and verification of models can not be dispensed with, since they allow one to decide which are the candidate models, but most of the *meaning* comes from the modelling framework.

2.2 Possible Effects of Social Embeddedness on Behaviour

If one had a situation where the agents were highly embedded in their society, what noticeable effects might there be? The efficacy of being socially embedded from the point of view of the embedded agent comes from the fact that if the most appropriate model is one that takes in far more than just its interactions with its social environment, then that agent will not have access to that model – it can not explicitly model the society it inhabits. In general, this may mean that:

- it will be more productive for the agent to cope by constructing behaviours that will allow it to exploit the environment rather than attempting to model its environment explicitly;
- it is worth frequently sampling and interactively testing its social environment to stand in stead of complete internal models of that environment (e.g. gossip);
- agents specialise to inhabit a particular social niche, where some sub-set of the total behaviour is easier exploit;
- at a higher level, there may be a development of social structures and institutions to ‘filter out’ some of the external complexity (Luhman, as summarised in [2]);

To summarise, the effect of being socially embedded might be that the agents are forced to develop set of heuristics that are specific to their particular society, rather than ones which are predictable from a more general top-down analysis.

2.3 Checking for Social Embeddedness

Given the presence of social embeddedness can have practical consequences on the modelled social behaviour, then it can be checked for. This is particularly so for a model of artificial agents, because the data is fully available, but it also means it is necessary to specify the modelling framework and selection criteria first.

Let us suppose that our criteria for model goodness are complexity and explanatory power. By explanatory power, I mean the extent of the phenomena that the model describes. Thus there is a familiar trade-off between explanatory power and complexity in *our* modelling of our simulation [14]. If two descriptions of the simulation are functionally the same, the social embeddedness comes out as a difference between the complexity of the models at the agent and social levels.

In the model below we will use a rough measure of the social embeddedness based on where most of the computation takes place that determines an agent’s communication and action. This will be indicated by the proportion of nodes which preform an external reference to the individual actions of other agents to those nodes that preform internal calculations (logical, arithmetic, statistical etc.).

3 A Example: a Model of Co-evolving Social Agents

3.1 The Set-up

The model is based upon Brian Arthur’s ‘El Farol Bar’ model [1], but extended in several respects, principally by introducing learning and communication. There is a fixed population of agents (in this case 10). Each week each agent has to decide whether or not to go to El Farol’s Bar on thursday night. Generally, it is advantageous for an agent to go unless it is too crowded, which it is if 67% or more of all the agents go. Before making their decision agents have a chance to communicate with each other.

3.1.1 The environment

Each agent gets the most utility for going when less than 7 of the other agents go (0.7), they get a fixed utility (0.5) if they do not go and the lowest utility for going when it is crowded (0.4).

In this way there is no fixed reward for any particular action because the utility gained from going depends on whether too many other agents also go. In this way there is no fixed goal for the agent's learning, but it is relative to the other agent's behaviour. Thus it is in each agent's interest to discoordinate their action with a majority of the others. It is impossible for all agents to gain the maximum utility, there is always some conflict to provide a potential for continual dynamics.

3.1.2 The agents

Each agent has a population of (pairs of) expressions that represent possible behaviours in terms of what to say and what to do. This population is fixed in number but not in content. These expressions are taken from a formal language which is specified by the programmer, but the expression can be of any structure and depth. Each agent does not 'know' the meaning or utility of any expression, communication or action – it can only evaluate each whole expression as to the utility each expression would have resulted in if it had used it in the past to determine whether it would go to the bar or not and the other's behaviours had remained the same. Each week each agent takes the best such pair of expressions and uses them to determine its communication and action.

Each agent has a fairly small population of such models (in this case 40). This population of expressions is initially generated according to the specified language at random. In subsequent generations the population of expressions is developed by a genetic programming [11] algorithm with a lot of propagation and only a little cross-over.

The formal language that these expressions are examples of is quite expressive. The primitive nodes and terminals allowed are shown in figure 2. It includes: logical operators, arithmetic, stochastic elements, self-referential operations, listening operations, elements to copy the action of others, statistical summaries of past numbers attending, operations for looking back in time, comparisons and the quote operator.

Talk nodes:AND, OR, NOT, plus, minus, times, divide, boundedByPopulation, lessThan, greaterThan, saidByLast, wentLastWeek, randomIntegerUpTo, numWentLag, trendOverLast, averageOverLast, previous, quote

Talk terminals:IPredictedLastWeek, randomGuess, numWentLastTime

Action nodes:AND, OR, NOT, saidBy, wentLastWeek, previous

Action terminals:IPredictedLastWeek, IWentLastWeek, ISaidYesterday, randomDecision

Constants (either):1, 2, 3, 4, 5, 6, 7, 8, 9, 10, maxPopulation, True, False, barGoer-1, barGoer-2, barGoer-3, barGoer-4, barGoer-5, barGoer-6, barGoer-7, barGoer-8, barGoer-9 barGoer-10

Figure 2. The primitives allowed in the talk and action expressions

Some example expressions and their interpretations if evaluated are shown in figure 3. The primitives are typed (boolean, name or number) so that the algorithm is strictly strongly-typed genetic program following [13].

Talk expression:[greaterThan [randomIntegerUpTo [10]] [6]]

Action expression:[OR [ISaidYesterday] [saidBy 'barGoer-3']]

Interpretation: Say 'true' if a random number between 0 and 10 is greater than 6, and go if I said 'true' or barGoer-3 said 'true'.

Talk expression:[greaterThan [trendOverLast [4]] [averageOverLast [4]]]

Action expression:[NOT [OR [ISaidYesterday] [previous [ISaidYesterday]]]]

Interpretation: Say 'true' if the number predicted by the trend indicated by the attendance yesterday and four weeks ago is greater than the average attendance over the last four weeks, and go if I did not say 'true' yesterday or last week.

Talk expression:[OR [saidByLast 'barGoer-3] [quote [previous [randomGuess]]]]

Action expression:[AND [wentLastWeek 'barGoer-7'] [NOT [IwentLastWeek]]]

Interpretation: Say 'true if barGoer-3 said that last week, else say "[previous [randomGuess]]", and go if barGoer-7 went last week and I did not.

Figure 3. Some example expressions

The reasons for adopting this particular structure for agent cognition is basically that it implements a version of rationality that is credible and bounded but also open-ended and has mechanisms for the expression of complex social distinctions and interaction. In these respects it can be seen as a step towards implementing the 'model social agent' described in [3]. For the purposes of this paper the most important aspects are: that the agent constructs its expressions out of previous expressions; that its space of expressions is open-ended allowing for a wide variety of possibilities to be developed; that it has no chance of finding the optimal expressions; and that it is as free from 'a priori' design restrictions as is practical and compatible with it having a bounded rationality. This is described in more detail in [8, 10].

3.1.3 Communication

Each agent can communicate with any of the others once a week, immediately before they all decide whether to go to the bar or not. The communication is determined by the evaluation of the talk expression and is usually either 'true' or 'false'. The presence of a quoting operator (quote) allows subtrees of the talk expression to be the content of the message. If a quote node is reached in the evaluation of the talk expression then the contents of the subtree are passed down verbatim rather than evaluated. If this is returned as the result of an evaluation of the talk expression then this is the message that is communicated.

The content of the messages can be used by agents by way of the saidBy and saidByLast nodes. The other agents can use the message in its evaluation of its expressions – the whole expression will be substituted instead of the saidBy (or saidByLast) node and evaluated as such. The agent can also use the output of its own messages.

3.1.4 Implementation

The model was implemented in a language called SDML (strictly declarative modelling language), which has been developed at the Centre for Policy Modelling specifically for social modelling [16].

3.2 Case Studies from the Results

A more complete examination of the results from this model can be found in [7]. In figure 4 the attendance patterns of the agents during the runs is shown.

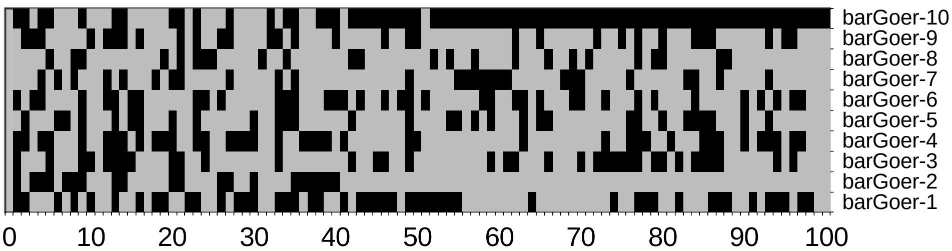

Figure 4. Attendances (grey=went, black=stayed at home)

This seems to be fairly stochastic with some specialisation between agents, but is more accurately described as a version of globally coupled chaos [6].

In figure 5, I show some of the specific causation between the talk and action expressions of the ten agents during the last three weeks of each run of the simulation. This only shows the causation due the saidBy, saidByLast and wentLastWeek primitives that are not logically redundant. So it does not show any causation via attendance statistics, or the self-referential primitives (e.g. ISaidYesterday, IPredictiedLastWeek and IWentLastWeek). There is a small box for the talk and action expression of each agent (numbered upwards from barGoer-1 to barGoer-10). The numbers in the boxes are a the total number of backward causal lines connected to that box if one followed the causation backward (restricted to the last three weeks only). This is an indication of how socially embedded the agent is – a larger number indicates that there is more complex causal chain determining its action (or communication), passing through many other agents. A detailed example is analysed below.

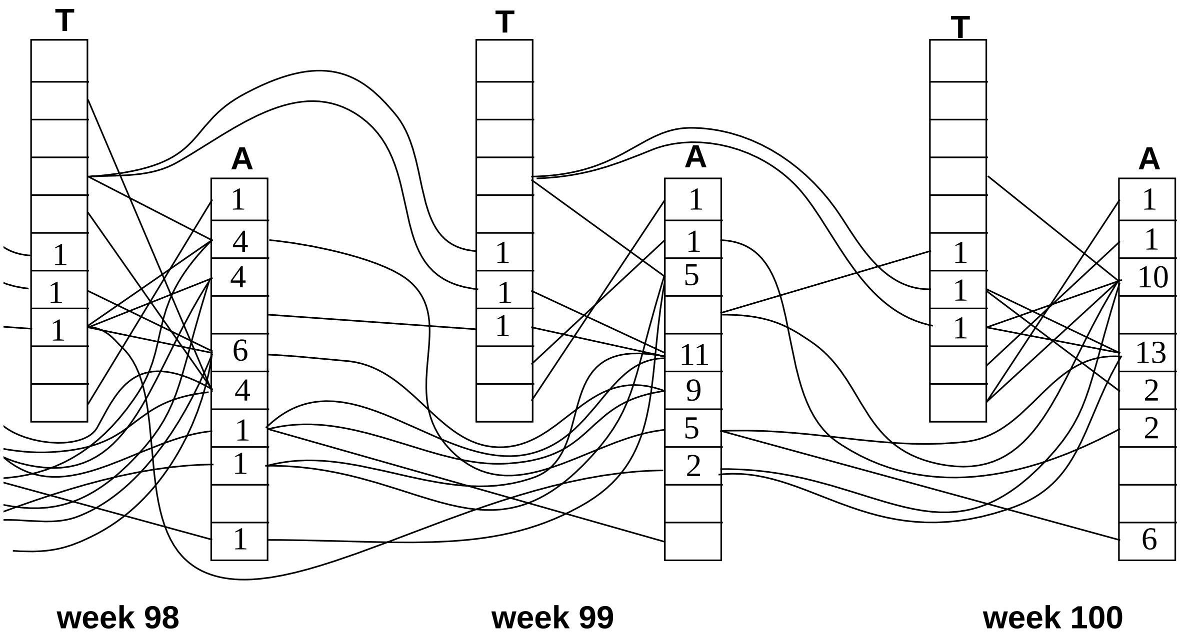

Figure 5. Causation net for the last 3 iterations of the simulation

In order to illustrate social embedding I analyse a more detailed case study of agent's behaviour and the cause one can attribute to it. To give a flavour of how complex a detailed explanation of behaviour can get I will follow back the chain of causation for the action of barGoer-6 at week 100.

At week 100, barGoer-6's action expression was:

[OR [AND [OR [AND [AND [saidBy ['barGoer-4']] [OR [AND [NOT [wentLastWeek ['barGoer-3']]] [saidBy ['barGoer-3']]] [saidBy ['barGoer-4']]]]] [NOT [wentLastWeek ['barGoer-3']]]]] [saidBy ['barGoer-3']]] [NOT [wentLastWeek ['barGoer-3']]]] [wentLastWeek ['barGoer-4']]]

which simplifies to:

```
[OR
    [AND
        [OR
            [saidBy ['barGoer-4']]
            [saidBy ['barGoer-3']]]]
        [NOT [wentLastWeek ['barGoer-3']]]]
    [wentLastWeek ['barGoer-4']]]
```

substituting the talk expressions from bar goers 3 and 4 in week 100 gives:

```
[OR
    [AND
        [OR
            [saidByLast ['barGoer-7']]
            [wentLastWeek ['barGoer-7']]]]
        [NOT [wentLastWeek ['barGoer-3']]]]
    [wentLastWeek ['barGoer-4']]]
```

substituting the action expressions from bar goers 3, 4 and 7 in week 99 gives:

```
[OR
    [AND
            [OR
        [saidByLast ['barGoer-7']]
            [previous [OR [OR [T] [saidBy ['barGoer-2']]] [T]]]
        [NOT [previous [ISaidYesterday]]]]
    [previous [wentLastWeek ['barGoer-9']]]]
```

which simplifies to:

```
[OR
    [NOT [previous [saidBy ['barGoer-3']]]]
    [previous [wentLastWeek ['barGoer-9']]]]
```

substituting the talk expressions from barGoer-3 in week 99 gives:

```
[OR
    [NOT [previous [[wentLastWeek ['barGoer-7']]]]]
    [previous [wentLastWeek ['barGoer-9']]]]
```

substituting the action expressions from barGoers 7 an 9 in week 98 gives:

```
[OR [NOT [previous [previous [OR [OR [saidBy ['barGoer-10']] [OR [T] [OR
[randomDecision] [saidBy ['barGoer-2']]]]] [F]]]]] [previous [previous [NOT [AND [saidBy
['barGoer-2']] [AND [AND [saidBy ['barGoer-2']] [NOT [AND [saidBy ['barGoer-6']]
[wentLastWeek ['barGoer-6']]]]] [OR [AND [AND [AND [saidBy ['barGoer-2']] [OR [AND
[saidBy ['barGoer-2']] [NOT [AND [saidBy ['barGoer-6']] [wentLastWeek ['barGoer-6']]]]]
[saidBy ['barGoer-2']]]] [AND [saidBy ['barGoer-2']] [NOT [AND [AND [saidBy
['barGoer-2']] [AND [saidBy ['barGoer-2']] [saidBy ['barGoer-2']]]] [NOT [NOT [saidBy
['barGoer-2']]]]]]]] [AND [randomDecision] [NOT [saidBy ['barGoer-2']]]]]]]]
```

which simplifies to:

```
[previous [previous [NOT
[AND
    [saidBy ['barGoer-2']]
    [NOT [AND [saidBy ['barGoer-6']] [wentLastWeek ['barGoer-6']]]]]
```

substituting the talk expressions from barGoers 2 an 6 in week 98 gives:

```
[previous [previous [NOT
    [AND
        [greaterThan [1] [1]]
        [NOT [AND [[greaterThan [maxPopulation] [maxPopulation]]]
[wentLastWeek ['barGoer-6']]]]]
```

which finally simplifies to:

```
True
```

The above trace ignores the several important causal factors: it does not show the evolutionary processes that produce the action and talk genes for each agent at each week; it does not show the interplay of the agent's actions and communications upon events and hence the evaluation of expressions (and hence which is chosen next by agents); and in simplifying the expressions at each stage I have tacitly ignored the potential effects of the parts of the expressions that are logically redundant under this particular train of events. Even given these caveats the action of barGoer-6 at week 100 was determined by a total of 11 expressions, spread out over 6 other agents over three weeks.

On the other hand it is difficult to find models of the behaviour of barGoer-6 which does not involve the complex web of causation that occurs between the agents. It is not simplistically dependent on other particular agents (with or without a time lag) but on the other hand is not merely random. This agent epitomises, in a reasonably demonstrable way, social embeddedness.

3.3 Comments

The simulation exhibits most of the effects listed previously. This is, of course, unsurprising since I have been using the model to hone my intuitions on the topic. In particular:

- the expressions that the agents develop resemble constructs rather than models, in that they are opportunistic, they do not reflect their social reality but rather constitute it;
- the constructs can appear highly arbitrary – it can take a great deal of work to unravel them if one attempts to explicitly trace the complex networks of causation;
- the agents do frequently use information about the communication and actions of others in stead of attempting to explicitly predict their environment – this is partly confirmed by a general analysis of the general distribution of primitive types in the expressions chosen and developed by agents in figure 6;
- the agents do specialise as they co-develop their strategies – this is examined in greater depth elsewhere [6];

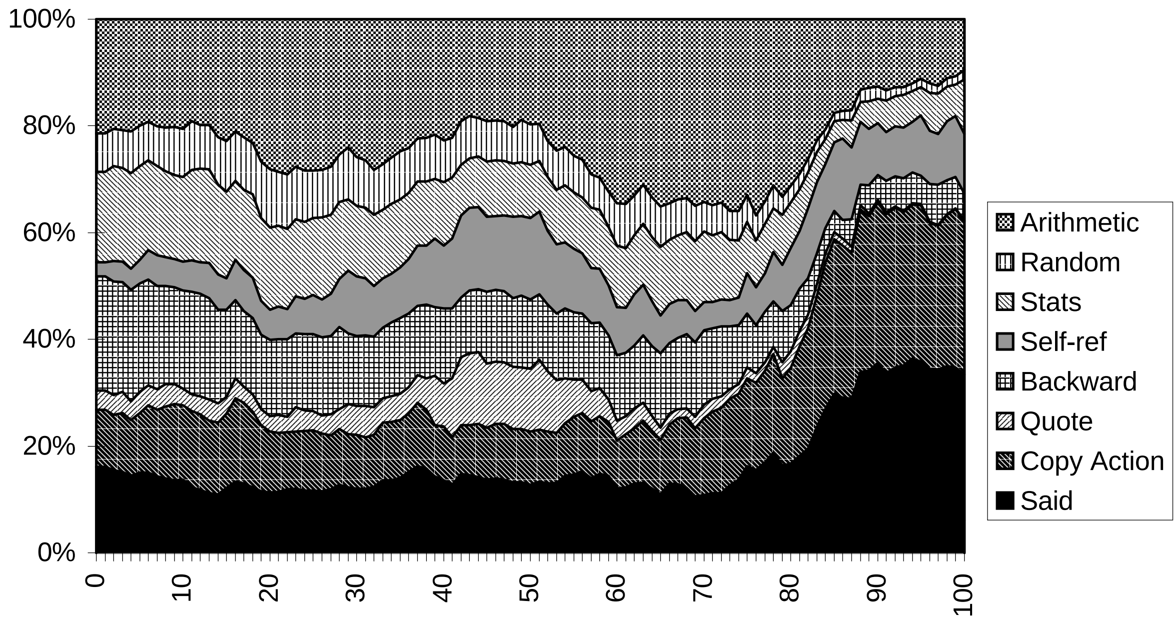

Figure 6. Distribution of the relative proportions of some primitive types

4 Possible Conditions for the Emergence of Social Embedding

What might enable the emergence of social embeddedness? At this point one can only speculate, but some factors are suggested by the above model. They might be:

- the ability of agents significantly to effect their environment – so that they are not limited to an essentially passive predictive role;
- the co-development of the agents – for example, if agents had co-evolved during a substantial part of the development of their genes then it is likely that this evolution would have taken advantage of the behaviour of the other agents; this would be

analogous to the way different mechanisms in one organism develop so that they have multiple and overlapping functions that defy their strict separation [18];

- the existence of exploitable computational resources in the society – so that it would be in the interest of agents to use these resources as opposed to performing the inferences and modelling themselves;
- the possibility of open-ended development by the agents – if the space of possible constructs was essentially small, then the optimal model of the society that the agent inhabited would be feasible for it;
- mechanisms for social distinction, hence the ability to form one-one relationships
- the ability to develop the *selection* of information sources;
- the ability to frequently sample and probe social information (i.e. gossip), (as in the 'social intelligence hypothesis' of [12]).

5 Consequences for the Development of Social Agents

It is interesting that the conditions for social embedding listed above, seem point in an opposite direction to some of the accepted guidelines for building multi-agent systems for engineering purposes. To illustrate this I will quote from Wooldridge and Jennings' paper on the pit-falls of agent-orientated development [20], from section 7.2 entitled "You have too many agents".

> *"... If a system contains many agents (many is often interpreted as greater than 10), then the dynamics can become too complex to manage effectively.*
>
> *There are several techniques that one can use to try to manage a system in which there are many agents. First, one can place it under central control... Another way... is to severely restrict the way in which agents can interact... one can ensure that there are few channels of communication...* [or] *by restricting the way in which agents interact. Thus very simple cooperation protocols are preferable over richer ones, with "one-shot" protocols... being both adequate and desirable for many applications."*

Now these are sensible warnings for the software engineer, but they are not necessarily relevant for social simulation[1], since the unforeseen behaviour that the engineer is try to prevent is what the social simulator is interested in. For the social simulator, the issue of how society can impact upon individual behaviour is at least as important as how individuals impact on society [9].

The point is that it is *possible* that one can not *engineer* truly social agents because a critical aspect of their sociality might come from their social embeddedness. In other words you might get a different result if you design agents and later put them together to interact, than if the agents learn significant parts of their 'cognitive content' in the context of each others learning. If this is the case then if one wants truly social agents then there may be no choice but to allow such agents to embed *themselves* into a society by a such a process of co-development – it would be something that we could not do for them.

1. Of course, many of the other warnings are relevant.

References

[1] Arthur, B. (1994). Inductive Reasoning and Bounded Rationality. *American Economic Association Papers,* **84**: 406-411.

[2] Bednarz, J. 1984. Complexity and Intersubjectivity. *Human Studies*, **7**:55-70.

[3] Carley, K. and Newell, A. (1994). The Nature of the Social Agent. *Journal of Mathematical Sociology*, **19**:221-262.

[4] Edmonds, B. (1997). Complexity and Scientific Modelling. 20th International Wittgenstein Symposium, Kirchberg am Wechsel, Austria, August 1997. http://www.cpm.mmu.ac.uk/cpmrep23.html

[5] Edmonds, B. What is Complexity?: the philosophy of Complexity per se with application to some examples in evolution. In F. Heylighen & D. Aerts (eds.): *The Evolution of Complexity*, Dordrecht: Kluwer.

[6] Edmonds, B. (forthcoming). Modelling Bounded Rationality In Agent-Based Simulations using the Evolution of Mental Models. In Brenner, T. (ed.), *Computational Techniques for Modelling Learning in Economics*, Dordrecht: Kluwer.

[7] Edmonds, B. (1998). Capturing social Embeddedness: a constructivist approach. CPM Report 98-34, MMU, Manchester, UK. http://www.cpm.mmu.ac.uk/cpmrep34.html

[8] Edmonds, B. (forthcoming in 1998). Modelling Socially Intelligent Agents. *Applied Artificial Intelligence*, **12**(7).

[9] Edmonds, B. and Dautenhahn, K. (1998). The Contribution of Society to the Construction of Individual Intelligence. Workshop on Socially Situated Intelligence, SAB’98. Zurich, August 1998.

[10] Edmonds, B. and S. J. Moss, (1997). Modelling Bounded Rationality using Evolutionary Techniques. *Lecture Notes in Computer Science*, **1305**:31-42.

[11] Koza, J.R. (1992). *Genetic Programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.

[12] Kummer, H., Daston, L., Gigerenzer, G. and Silk, J. (1997). The social intelligence hypothesis. In Weingart et. al (eds.), *Human by Nature: between biology and the social sciences*. Hillsdale, NJ: Lawrence Erlbaum Associates, 157-179.

[13] Montana, D. J. (1995). Strongly Typed Genetic Programming, *Evolutionary Computation*, **3**:199-230.

[14] Moss, S.J. and Edmonds, B. (1998). Modelling Economic Learning as Modelling. *Cybernetics and Systems*, **29**:5-37.

[15] Moss, S. J., Edmonds, B. and Wallis, S. (1997). Validation and Verification of Computational Models with Multiple Cognitive Agents. CPM Report 97-25, MMU.

[16] Moss, S. J., H. Gaylard, S. Wallis, and B. Edmonds, (1998). SDML: A Multi-Agent Language for Organizational Modelling. *Computational and Mathematical Organization Theory*, 4:43-69

[17] Shoham, Y. (1993). Agent-Orientated Programming. *Artificial Intelligence*, **60**:51-92.

[18] Wimsatt, W. (1972). Complexity and Organisation, in Scavenger and Cohen (eds.), *Studies in the Philosophy of Sciences*. Dordrecht: Riddle, 67-86.

[19] Wooldridge, M. (1997). Agent-Based Software Engineering. *IEE Transactions on Software Engineering*, **144**:26-37.

[20] Wooldridge, M. and Jennings, N.R. (1998). Pitfalls of Agent-Oriented Development. In K. P. Sycara and M. Wooldridge, editors: Agents '98: Proceedings of the Second International Conference on Autonomous Agents ACM Press, May 1998.