

Agent Design for Agent-Based Modelling

Jim Doran¹

Department of Computer Science
University of Essex

ABSTRACT:

To build an agent-based computational model of a specific socio-environmental system requires that answers be found to several important questions including: what social actors and contextual entities are to be modelled as software agents? With what mental functions must these software agents be equipped? What standard design or designs should be used to create the software agents? This paper concentrates upon the last of these questions. The currently available range of agent designs is considered, along with their limitations and inter-relationships. How best to choose a design to meet the requirements of a particular modelling task is discussed and illustrated by reference to the task of designing an informative agent-based model of a segmented, polycentric and integrated network ("SPIN") organisation of the type analysed by Gerlach in the context of environmental activism.

Introduction

This paper is about how best to design, not merely to implement, an agent-based model on a computer of a real-world socio-environmental scenario. Although much has been written on this topic in recent years (e.g. Doran 1997, 2001; Gilbert and Troitzsch, 1999, Ch. 8; Moss, 2001; Bonabeau, 2002), the various questions that arise have not been fully answered. Indeed, I fear that much agent-based model design continues to be arbitrary and ill conceived. Thus although some of what is to be said may seem commonplace, it is important to revisit the issues.

The start point is that given a real-world scenario (for example, a busy supermarket), and some questions about it to which we seek answers (for example, how best to reduce average evacuation time in case of fire), we must design a model capable of providing those answers:

SCENARIO + QUESTIONS → MODEL DESIGN

Major decisions to be made on the way to a suitable model design are: (a) whether to use a specific or a non-specific model, (b) the level of abstraction of the model, (c) the cognitive content of the agents, and (d) the choice of agent architectures. Notice that all of these decisions *must* be made whether explicitly or not and whether arbitrarily or in a considered and informed way.

Specific and Generic Models

A model may refer to a specific situation, for example a particular supermarket at a particular location. Or it may refer to the essential features of a type of situation, for

¹ Email: doraj@essex.ac.uk

example the essential features of any supermarket at any location. This distinction is an important one. Specific models require much detailed observation to justify their particular and detailed structure and potentially yield specific insights and predictions. By contrast, generic models require less detailed observation and more preliminary interpretation and recognition of just what are the essentials of a type of situation. They potentially yield general insights. Specific models are naturally used to predict the likely outcomes of specific actions (including no action). Generic models are used to discover the properties that a real-world scenario has as a result of its structure and dynamics. A current and persuasive example of the use of generic models to explore social phenomena is the work of Hemelrijk and colleagues (Hemelrijk, 2003) who are investigating how certain types of animal social behaviour, especially those found in some ape communities, may emerge from relatively simple elements of behaviour. Their work is important not merely because it illustrates a certain type of agent-based modelling, but because it could significantly change our understanding of how the social behaviour in question arises.

Which Agents?

To set up an agent-based model, whether specific or generic, one must decide what are to be the agents within it. These will probably correspond in some fashion to actors in the real world. Indeed it is natural to envisage a two-staged process. First to decide what actors are to be “recognised” in the real world scenario in question, and then to decide how the actors recognised are to be mapped into agents within the model. It is clear that the decisions made must depend both on the reality of the scenario and on the objectives of the modelling study.

The problem of recognising actors in the world is far from trivial once cultural pre-conceptions are discarded². For example, how many actors are there in a University? Is the Department of Mathematics an actor? Are overseas students collectively an actor? Is the pleasant cedar just outside the Vice-Chancellor’s window that from time to time calms down the VC, an actor? Are the fly on the windowpane and the expert system on the VC’s personal computer actors?

Given that a set of actors has somehow been systematically recognised in or “read into” the real-world scenario, according to current cultural norms or otherwise, the next question is how this set of actors is to be transformed into a set of agents within the model. A one-to-one mapping from actors to agents is likely to seem unnecessary or even impossible. Some notion is needed of what actors are relevant to the objectives of the modelling study and this involves the model as a whole.

What Level of Model Abstraction/Aggregation?

Deciding what are to be the agents in a model is closely connected with deciding the level of abstraction or aggregation of the model, that is, just how much detail is to be

² See the analysis and discussion in Doran (2002a, 2002b) with associated computational experiments and results.

included. The following principles seem applicable to deciding the level of abstraction/aggregation:

Principle: The model must be sufficiently detailed that it can address the questions to be answered. For example, if the question posed concerns how many members of a group will support its leader in a certain eventuality, modeling the group solely as a single collective agent is inappropriate. However, too much detail is both unnecessary and likely to be computationally overwhelming. Hence:

Principle: The model should be as abstract/aggregated as possible subject to the requirement:

EACH of a COMPLETE set of inter-variable relationships and processes in the model must:

EITHER be reliably set from empirical observation (even if the relationship or process is non-linear)

OR be feasibly subjected to experimental variation (and in due course be so examined)

OR demonstrably have no significant impact (in which case it can probably be discarded).

Principle: Assumptions based on pre-conceptions are to be avoided.

It should be clear that meeting these requirements is rarely easy, wherein lies much of the difficulty in agent-based modelling. Perhaps the central issue is just what potential properties of the model can reliably and completely be observed in the real-world scenario. This is a practical matter. Note that generic models are somewhat less dependent on reliable observation than are specific models.

What Agent Cognition and what Agent Architecture?

The choice of cognition and structure to be designed into the agents in the model is an aspect of the foregoing considerations. We have to decide what cognitively oriented computations the agents are to perform in the model. This is so whether all agents perform the same computations or different agents perform different computations. Furthermore, we must decide what information (knowledge, belief) each agent is initially to possess. Again different agents may well possess different initial information.

It must further be decided what actual agent architectures are to be used. In this context an agent architecture is a structural and process design for the agent. The major differences between agent types lie in their architectural concepts rather than in implementation software. There is a significant range of recognised software agent architectures available for use in agent-based models (Weiss, 1999; Wooldridge, 2002). They include architectures based upon:

1. sets of variables with associated condition-action rules (incl. “fuzzy” rules)
2. artificial neural networks
3. behaviours and subsumption
4. predictive planner and internal model
5. logic systems e.g. BDI
6. hybrid and/or multi-layer

Architectures with associated condition-action rules (1) can be very simple – no more than a couple of variables and a handful of (possibly fuzzy) rules – or can be highly complex and support advanced cognitive processes. Artificial neural networks (2) take as their building blocks simple computational abstractions of real brain neurones usually arranged in layers. These architectures are typically used to implement reactive agent behaviour with some kind of associated learning. Architectures based upon behaviours with subsumption relationships (3) between them were originally devised for robot control, where effective low-level behaviours are essential. In their original form they avoided the use of internal models or planning. Architectures that do use internal models and predictive planning (4) tend to be more “symbolic” in flavour and the more effective examples are based on the long tradition of AI work concerned with automatic planning systems. Formal logic based architectures (5), of which the best known is perhaps the BDI (“beliefs, desires intentions”) architecture, are derived from structures and processes in fragments of mathematical logic, but necessarily take a computational form in practice sometimes akin to (1) and (4) above. Finally, hybrid multi-layer architectures usually have a lower level that is reactive (an architecture of type (1), say), then a higher layer which uses modelling and planning (type (4), say) and often come with a further even higher layer with a more social function, perhaps involving communication and cooperative planning with other similar agents in a shared collective environment.

These architectures are not fully standardised. They vary not merely in their effective content, but in the degree to which they are well defined and available off the shelf. They can be mapped along at least three distinct dimensions: (a) by the type of their basic elements (e.g. rules, artificial neurones, symbols), (b) by the range of “cognitive” functions that are provided, and (c) by the definiteness with which they are specified. Most of these architectures are “open” in the sense that with due programming effort any cognitive function can be provided within them to at least some degree. In particular, all of these agent architectures can support adaptive agents.

Although their users often discuss particular types of architecture as if they were quite distinct from (and superior to) alternatives, in fact these architectures are not wholly independent. Furthermore there are some commonly encountered erroneous beliefs concerning them. For example, it is often assumed that control architectures typically deployed in robot contexts are quite different from and irrelevant to software modelling and vice versa; that “rule-based” architectures cannot involve learning and are necessarily limited in their human social significance because people are not “rule bound”; and that there is a major difference between “symbolic” and “neural network” architectures. In fact all of these beliefs are questionable if not downright mistaken. They arise from a too superficial and conventional view of the architectures in question. Further points that merit reiteration are that usually anything can be programmed in any agent architecture or software system (see next section), and that

“logic-based” software rarely gains any real power or reliability from its association with mathematical logic.

Software Platforms

Software platforms designed to support agent-based modelling typically provide some specific support for agent design and implementation. Platforms such as SWARM, CORMAS, SDML and RePast (see Gilbert and Troitzsch, 1999, Ch. 8; Gilbert and Bankes, 2002) make it easier to design and implement models and agents of certain types but do not necessarily guide the user to the right choices. Also, there are many software multi-agent support platforms that, although not designed with social modelling in mind, could easily be used for that purpose.

Choosing the Agent Architectures

Given previous choices of agents and of their required cognitive and information storage capabilities, how can we best to choose an agent architecture to meet the requirements of a particular modelling task? First we must consider why we do not design an agent architecture (or architectures) suitable for each particular model. The answer is that it is no easy matter to design and implement such an architecture. To attempt to do so is likely either to result in a trivial architecture, or to set up a major and unnecessary design task. To ignore architectures already to hand is surely a mistake unless they can be demonstrated to be unsuitable.

It is sometimes argued that in a model which is to address social phenomena, since the agents must behave socially, we should design specifically social agent architectures, no doubt with a major emphasis on support for inter-agent communication. This argument is often supported by an explicit or implicit claim that the agent architectures listed earlier are *not* social. But the fact is that, *in software*, any distinctive “social cognition” that an agent may have will inevitably be built upon its non-social cognition, so that the semi-standard architectures remain relevant.

To sum up, we should aim to choose from amongst the semi-standard architectures, possibly embedded in a multi-agent software platform, by reference to the principles stated earlier. This is, of course, easier said than done!

An Example: Gerlach’s SPIN organisations

By a SPIN, Gerlach (2001) means a social movement that is a segmentary, polycentric and integrated network. He illustrates and discusses SPINs by reference to the environmental movement in the USA over the last four decades (Gerlach and Hine, 1970), and its component groups such as *Friends of the Earth* and the *Earth Liberation Front*, and, by contrast, to the *Wise Use* movement that sought to counter environmental activism.

Gerlach (*loc cit*, p. 289) focuses attention on the segmentary nature of SPINs. They are “composed of many diverse groups, which grow and die, divide and fuse,

proliferate and contract”. They are also polycentric. They have “multiple, often temporary, and sometimes competing leaders or centers of influences”. And finally they are networked, “forming a loose, reticulate, integrated network with multiple linkages through travelers, overlapping membership, joint activities, common reading matter, and shared ideals and opponents”.

Gerlach argues that SPINS are adaptive and offers seven reasons why this is so (Gerlach, *loc cit*, pp 302-6):

1. "... prevents effective suppression by the authorities and the opposition."
2. "Factionalism and schism aid the penetration of the movement into a variety of social niches."
3. "Multiplicity of groups permits division of labor and adaptation to circumstances."
4. ".... contributes to system reliability."
5. "Competition between groups leads to escalation of effort."
6. "facilitates trial-and-error learning through selective disavowal and emulation."
7. "...promotes striving, innovation, and entrepreneurial experimentation in generating and implementing social change."

Using Agent-Based Modeling to verify SPIN Adaptive Functionality

To use agent-based modelling to test these claims for the adaptive functionality of SPINs it is inappropriate to model a specific instance of a SPIN (see discussion earlier). Rather we should create a computational version of a typical SPIN. This is made easier because Gerlach’s discussion, although directed to the environmental movement and counter-movement in the USA, is largely pitched in general terms. Therefore to proceed we must define SPIN and traditional non-SPIN organisations, and then a range of strategies suitable to be used to attack them. Following Gerlach, we would then expect to demonstrate experimentally that the SPIN organisations fare better in most circumstances³.

But what does testing Gerlach’s claims for SPIN functionality require by way of agent structure and process? Gerlach’s discussion implies that the SPIN actors deploy a comprehensive range of high-level cognitive functions. The issue is therefore whether we need to incorporate some or all of them in the agents of a SPIN model. Consider, however, a network’s response to a “hostile” process that successively deletes actors, network members, at random – (see (1) above). It is difficult to see how the effect on

³ If our framework of consideration were more mathematical, we might reasonably hope to prove this conjecture as a theorem.

the network of a deletion of a single actor in the network can be reliably predicted other than by empirical observation and subsequent generalisation. To try reliably to incorporate agent decision-making and cognitive processing within the model so that the “right” network responses will emerge is to aim at a level of complexity that is computationally and observationally overwhelming, ultimately leading, one can predict, to model and agent structures that are essentially arbitrary and unverifiable. This is so even if (implausibly) rationality is assumed in the model as a way of achieving predictability.

The conclusion is therefore a somewhat negative one: it is that no significant internal structure can safely be associated with the agents in the model. We reach this conclusion for this particular application because the actors in the real-world scenario (the members of the environmental movement) are not performing systematically observable role driven actions, or systematically observable routine learning, that can be reliably replicated within a software agent architecture. As discussed earlier, without reliable observation at a particular level of abstraction and aggregation, a reliable model cannot be grounded at that level.

SPINS and Terrorist Networks

Gerlach’s first reason why SPINS are adaptive is that their structure “prevents effective suppression by the authorities and the opposition.” His discussion of this point foreshadows ongoing work by Carley and her colleagues (Carley et al, 2003) that uses agent-based modelling to address the effectiveness of strategies for disrupting networks, in particular for destabilizing terrorist networks. Carley and colleagues, unlike Gerlach, are concerned with fully covert organisations, but this does not prevent there being close similarities in their work. Carley’s investigation has already provided interesting and important preliminary results notably that different strategies are required to destabilize cellular and distributed organisations from those that are effective on hierarchical organisations. Furthermore, different destabilisation strategies impact differently on different measures of the performance of the target organisation..

Conclusion

There exist a range of semi-standard software agent designs. But these are not always well understood by those creating agent-based models, and are rarely explicitly used by them. It is not clear how an agent architecture should be chosen on a particular occasion. At best a few broad principles can be stated, as we have done earlier. This lack of guidelines is surely important. Bonabeau’s (2002) recent comments that “ABM is a mindset more than a technology” (p. 7280) and that model building remains “an art more than a science” (p.7287) confirm the difficulty of our situation.

Is it a limitation that we cannot create software agents to build into our agent-based models that display human levels of intelligence and consciousness⁴? It is tempting to

⁴ Consideration of consciousness is timely because there is much ongoing work. See, for example, the ESF Exploratory Workshop on “Models of Consciousness” held in Birmingham, UK, Sept 1-3rd 2003.

answer “not at all”, on the grounds that models can and should be much simpler than that which they model. Yet human society is surely the emergent product of human intelligence and consciousness. It would be surprising if these fundamental human characteristics could be entirely ignored in models of human society.

References

Bonabeau E (2002) Agent-based modeling: methods and techniques for simulating human systems. *PNAS, May 14th 2002 Vol. 99 suppl 3, 7280-7287.*

Carley K M, Reminga J, and Kamneva N (2003) Destabilizing Terrorist Networks, *NAACSOS conference proceedings, Pittsburgh, PA*

Doran J E (1989) Distributed Artificial Intelligence and the Modelling of Socio-Cultural Systems. In *Intelligent Systems in a Human Context* (eds. L A Murray and J. T. E. Richardson) Oxford Science Publications

Doran J E (1997) "From Computer Simulation to Artificial Societies", *Transactions SCS, 14(2)*, 69-77, June 1997 [Special Issue: Multi-Agent Systems and Simulation]

Doran J. E. (2001) Can Agent-Based Modelling REALLY be Useful? In *Cooperative Agents: Applications in the Social Sciences* (Eds. N. J Saam and B. Schmidt) Kluwer Academic Publishers Pp 57-81

Doran J.E. (2002a) “Agents and MAS in STaMs”. In: “Foundations and Applications of Multi-Agent Systems: UKMAS Workshop 1996-2000, Selected Papers” (Eds. M d’Inverno, M Luck, M Fisher, C Preist), Springer Verlag, LNCS 2403, July 2002, Pp. 131-151

Doran J E (2002b) Detecting Agents: the McCarthy and Hayes Martian Problem, *Working Notes of UKMAS 2002* (The Fifth UK Workshop on Multi-Agent Systems) held at the Foresight Centre, University of Liverpool, December 18-19th, 2002

Doran J E and Gilbert N, (1994). Simulating Societies: an Introduction. In *Simulating Societies* (N. Gilbert and J. E. Doran eds.) UCL Press.

ESSA (2003) First Congress of the European Social Simulation Association, Groningen, September 2003

Gerlach L P. (2001) The Structure of Social Movements: Environmental Activism and its Opponents. In *Networks and Netwars: the Future of Terror, Crime and Militancy*. (Eds. John Arquilla and David Ronfeldt) Santa Monica, CA: Rand

Gerlach L P and Hine V H (1970) *People, Power, Change: Movements of Social Transformation*. Indianapolis: Bobbs-Merrill.

Gilbert N and Bankes, S (2002) Platforms and methods for agent-based modeling *PNAS, May 14th 2002 Vol. 99 supplement 3, 7197-7198*

Gilbert N and Troitzsch K, (1999) *Simulation for the Social Scientist* UCL Press: London

Hemelrijk C (2003) Social phenomena emerging by self-organisation in a competitive, virtual world ('DomWorld') *First Congress of the European Social Simulation Association (ESSA)*, Groningen, September 2003

Moss S (2001) Messy Systems – The Target for Multi Agent Based Simulation *Multi-Agent Based Simulation* (S. Moss and P Davidson eds.) Springer LNAI 1979

Moss S. (2003) Presidential Address to the *First Congress of the European Social Simulation Association (ESSA)*, Groningen, September 2003.

Weiss G ed. (1999) *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. MIT Press.

Wooldridge M (2002) *An Introduction to MultiAgent Systems* John Wiley & Sons, Ltd