



[Pietro Terna](#) (1998)

Simulation Tools for Social Scientists: Building Agent Based Models with SWARM

Journal of Artificial Societies and Social Simulation vol. 1, no. 2,
<<http://jasss.soc.surrey.ac.uk/1/2/4.html>>

To cite articles published in the *Journal of Artificial Societies and Social Simulation*, please reference the above information and include paragraph numbers if necessary

Received: 6-Mar-1998

Published: 31-Mar-1998

Abstract

Social scientists are not computer scientists, but their skills in the field have to become better and better to cope with the growing field of social simulation and agent based modelling techniques. A way to reduce the weight of software development is to employ generalised agent development tools, accepting both the boundaries necessarily existing in the various packages and the subtle and dangerous differences existing in the concept of agent in computer science, artificial intelligence and social sciences. The choice of tools based on the object oriented paradigm that offer libraries of functions and graphic widgets is a good compromise. A product with this kind of capability is Swarm, developed at the Santa Fe Institute and freely available, under the terms of the GNU license.

A small example of a model developed in Swarm is introduced, in order to show directly the possibilities arising from the use of these techniques, both as software libraries and methodological guidelines. With simple agents - interacting in a Swarm context to solve both memory and time simulation problems - we observe the emergence of chaotic sequences of transaction prices.

Keywords:

agent based models (ABM), chaos, intelligent agents, social simulation, Swarm

Introduction

1.1

This paper is about requirements for tool boxes in social simulation. It starts with the

definition of agent based models and then introduces the main problems arising in their construction, mainly focusing on software problems. The Section "[Agent based models](#)" underlines the usefulness of agent based models in the social science perspective, also focusing on the main computational problems (memory management and time management in simulation); the Section "[Building agent models: A review of techniques and requirements](#)" deals with the difficult matter of agent definition in the various fields (AI, the social sciences, and so on); finally, in the Section "[Objects and agents: The Agent Based Chaotic Dynamic Emergence \(ABCDE\) example](#)", which is about technical definitions of objects in a Swarm and Objective C context, we introduce a small specific application of those techniques.

Agent Based Models

2.1

The starting point is the choice on model foundations: If we choose the agent based model paradigm we enter a wide unexplored world where methodology and techniques are largely "under construction." An interesting overview comes from the following Web sites: [Syllabus of Readings for Artificial Life and Agent-Based Economics](#) ; [Web Site for Agent-Based Computational Economics](#); [Agent-Based Economics and Artificial Life: A Brief Intro](#); [Complex Systems at the University of Buenos Aires](#) ; [Computational Economic Modeling](#); [Individual-Based Models](#). With respect to social simulation we refer also to Conte et al. (1997) and, in economics, to Beltratti et al. (1996); also of interest is [The Complexity Research Project](#) of the London School of Economics and Political Science.

2.2

At present, the best plain introduction to agent based modelling techniques is in Epstein and Axtell (1996). For a thoughtful review of the book, see Gessler (1997).

2.3

As Epstein and Axtell (1996: 1) note:

Herbert Simon is fond of arguing that the social sciences are, in fact, the hard sciences. For one, many crucially important social processes are complex. They are not neatly decomposable into separate subprocesses--economic, demographic, cultural, spatial--whose isolated analyses can be aggregated to give an adequate analysis of the social process as a whole. And yet, this is exactly how social science is organized, into more or less insular departments and journals of economics, demography, political science, and so forth. Of course, most social scientists would readily agree that these divisions are artificial. But, they would argue, there is no natural methodology for studying these processes together, as they coevolve.

The social sciences are also hard because certain kinds of controlled experimentation are hard. In particular, it is difficult to test hypotheses concerning the relationship of individual behaviors to macroscopic

regularities, hypotheses of the form: If individuals behave in thus and such a way--that is, follow certain specific rules--then society as a whole will exhibit some particular property. How does the heterogeneous micro-world of individual behaviors generate the global macroscopic regularities of the society?

Another fundamental concern of most social scientists is that the rational actor--a perfectly informed individual with infinite computing capacity who maximizes a fixed (nonevolving) exogenous utility function--bears little relation to a human being. Yet, there has been no natural methodology for relaxing these assumptions about the individual.

Relatedly, it is standard practice in the social sciences to suppress real-world agent heterogeneity in model-building. This is done either explicitly, as in representative agent models in macroeconomics ([Kirman, 1992](#)), or implicitly, as when highly aggregate models are used to represent social processes. While such models can offer powerful insights, they "filter out" all consequences of heterogeneity. Few social scientists would deny that these consequences can be crucially important, but there has been no natural methodology for systematically studying highly heterogeneous populations.

Finally, it is fair to say that, by and large, social science, especially game theory and general equilibrium theory, has been preoccupied with static equilibria, and has essentially ignored time dynamics. Again, while granting the point, many social scientists would claim that there has been no natural methodology for studying nonequilibrium dynamics in social systems.

2.4

The response to this long, but exemplary, quotation, is social simulation and agent based artificial experiments. Following the Swarm documentation ([Swarm home page](#); for Swarm see also the position paper by Minar et al. [1996](#)) we introduce a general sketch about how one might implement an experiment in the agent based modelling field. An idealized experiment requires, first, the definition of: (i) the computer based experimental procedure and (ii) the software implementation of the problem.

2.5

The first step is that of translating the real base (the physical system) of our problem into a set of agents and events. From a computational point of view, agents become objects and events become steps activated by loops in our program. In addition, in a full object oriented environment, time steps are also organized as objects.

2.6

We can now consider three different levels of completeness in the structure of our software tools:

1. At the lowest level (i.e., using plain C) we have to manage both the agent memory

structures (commonly with a lot of arrays) and the time steps, with loops (such as "for" structures) driving the events; this is obviously feasible, but it is costly (a lot of software has to be written; many "bugs" have to be discovered);

2. at a more sophisticated level, employing object oriented techniques (C++, Objective C, etc.), we avoid the memory management problem, but we have nevertheless to run time steps via the activation of loops;
3. finally, using a high level tool such as Swarm, we can dismiss both the memory management problems and the time simulation ones; in high level tools, also the events are also treated as objects, scheduling them in time-sensitive widgets (such as action-groups). The [ABCDE](#) Swarm example introduced below is built following these design techniques.

2.7

Obviously, there are many alternatives to a C or Objective C environment. The Lisp perspective - with implementations such as [XLISP-STAT](#), developed by Luke Tierney - is a powerful one. XLISP-STAT is a public domain implementation of the Lisp language, in an object oriented flavour, integrated with a set of powerful statistical functions. So one can develop a model or an experiment with agents, runs it and do statistics upon it.

2.8

More generally, we have to consider the multiplicity of tools aimed at developing agent based software applications. For an idea of these tools, visit the Web sites:

www.eecs.uc.edu/~abaker/JAFMAS/compare.html and [Linux AI/Alife mini-HOWTO](#).

2.9

Any tool is useful in several domains of application. From the user's point of view, we face a cost-benefit analysis problem in order to choose the best, with the additional problem of the accelerating pace of computational product renewal. For example, two years ago Swarm was only in its first alpha test phase.



Building Agent Models: A Review Of Techniques And Requirements

3.1

In the previous Section we have outlined the main problems arising when we build agent based models to run artificial experiments. We have also indirectly introduced artificial agents, with a weak implicit definition, of the type "you know what an agent is, so an artificial one is . . ."

3.2

From a software engineering point of view, things are unfortunately not so easy. We have to explore and to delimit the field of what we call agents in the domain of social science simulation, mainly to understand what the other disciplines are doing and how to use their work. So we have to deal with the "Intelligent Agents" matter.

3.3

We have also the obvious aim of reducing, or not increasing, the [Babel](#) effect coming

from language specialisation. The Babel metaphor has at least two meanings: a positive one, in the direction of scientific discovery, and a negative one, in the sense of methodological inconsistency, bringing the impossibility of cooperation and of the exchange of results. The second meaning is propagating itself in the field of simulation experiments in the social sciences.

3.4

Now the agent idea - both in Agent Based Modelling (ABM) and in computer science - can be introduced, with a [scheme](#) showing the interactions of ABM construction and Intelligent Agent (IA) techniques.

3.5

If we define Intelligent Agents according to their computational capabilities (to solve computational problems; to apply search techniques; to move from a computer to another to perform their actions; etc.) we immediately accept the left double arrow of the figure. We can imagine having both the possibility of employing ABM techniques to develop simulated environment to test IA behaviour and employing IA as tools in the construction of ABMs, especially in the field of social sciences.

3.6

To look to some examples of tools in the Intelligent Agents field, see the Web sites at [JAFMAS](#) or at [IBM Aglets Workbench - Home Page¹](#) or at [Mobile agents at Dartmouth College](#).

3.7

Returning to our [scheme](#), we have to emphasise the meaning of the arrow linking ABMs to the "complexity" and "emergence" paradigms (in this paper a concrete example is introduced, with the [ABCDE](#) model). We have also to explain the more complicated meanings both of the broken arrow going from Intelligent Agents to Complexity and Emergence and of the question mark accompanying it. It is possible and useful to employ Intelligent Agents as metaphors of economic agents and of their behaviour, considering the possibility of interpreting the outcomes of the actions of these pieces of intelligent software as economic or social acts. This is not easy, obviously, but the chance to exploit the enormous effort that computer scientists are doing in the field is the advantage of operating in this way.

3.8

For a complete guide to Artificial Intelligence in the Intelligent Agent perspective, refer to Russell and Norvig ([1995](#)). A review of current literature can be found hyperlinking to [Isbister and Layton](#), *Intelligent Agents: A review of current literature*. Finally, to read some interesting and up to date critical consideration on this matter, refer to the position paper by [Wooldridge and Jennings](#), *Intelligent Agents: Theory and Practice*. They write, concerning a weak notion of agents and a stronger one:

A Weak Notion of Agency. Perhaps the most general way in which the term agent is used is to denote a hardware or (more usually) software-based

computer system that enjoys the following properties: (i) Autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state; (ii) social ability: Agents interact with other agents (and possibly humans) via some kind of agent-communication language (. . .); (iii) reactivity: Agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it; (iv) pro-activeness: Agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative. A simple way of conceptualising an agent is thus as a kind of UNIX-like software process, that exhibits the properties listed above. This weak notion of agency has found currency with a surprisingly wide range of researchers. For example, in mainstream computer science, the notion of an agent as a self-contained, concurrently executing software process, that encapsulates some state and is able to communicate with other agents via message passing, is seen as a natural development of the object-based concurrent programming paradigm (. . .) This weak notion of agency is also that used in the emerging discipline of agent-based software engineering: [Agents] communicate with their peers by exchanging messages in an expressive agent communication language. While agents can be as simple as subroutines, typically they are larger entities with some sort of persistent control.

A Stronger Notion of Agency. For some researchers - particularly those working in AI - the term 'agent' has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualised or implemented using concepts that are more usually applied to humans. For example, it is quite common in AI to characterise an agent using mentalistic notions, such as knowledge, belief, intention, and obligation (. . .) Some AI researchers have gone further, and considered emotional agents (. . .) (Lest the reader suppose that this is just pointless anthropomorphism, it should be noted that there are good arguments in favour of designing and building agents in terms of human-like mental states . . .).

3.9

We cannot discuss here the alternative interpretations, weak or strong, of the IA world. Nevertheless, it is clear that the debate is not irrelevant for the social science community.

3.10

Coming back again to our [scheme](#), we still have to explain the question mark. This is representative of the difficulty (which is however a general difficulty) of managing the results of Intelligent Agent experiments, where the tools used are quite different from those of social experimenters. We risk having a lot of data about observed behaviour and no capability to use them in a direct way.

3.11

All our observed behavioural data are the products of a metaphorical interpretation of tools aimed at solving other types of problems. So we have to use this kind of powerful tool considering always the costs and benefits of the effort and the results in doing experiments.

3.12

All this represents a prologue to turning our attention to the Swarm world again. The term "world" is appropriate, because Swarm is a system of software libraries, but also a vigorous interactive community working with them, as we can observe by taking part to the [Swarm mailing lists](#).

3.13

In the Swarm context, we use the Object-Oriented Programming language Objective-C. According to the Swarm documentation, computation in a Swarm application takes place by instructing objects to send messages to each other. The basic message syntax is

```
[targetObject message Arg1: var1 Arg2: var2]
```

where `targetObject` is the recipient of the message, `messageArg1:Arg2:` is the message to send to that object, and `var1` and `var2` are arguments to pass along with the message.

3.14

According to the Swarm documents:

Objective C messages are keyword/value oriented, that is why the message name `message Arg1: Arg2:` is interspersed with the arguments. The idea of Swarm is to provide an execution context within which a large number of objects can "live their lives" and interact with one another in a distributed, concurrent manner.

In the context of the Swarm simulation system, the generic outline of an experimental procedure takes the following form.

- i. Create an artificial universe replete with space, time, and objects that can be located, within reason, to certain "points" in the overall structure of space and time within the universe., and allow these objects to determine their own behavior according to their own rules and internal state in concert with sampling the state of the world, usually only sparsely.
- ii. Create a number of objects which will serve to observe, record, and analyze data produced by the behavior of the objects in the artificial universe implemented in step i.
- iii. Run the universe, moving both the simulation and observation objects forward in time under some explicit model of concurrency.

- iv. Interact with the experiment via the data produced by the instrumentation objects to perform a series of controlled experimental runs of the system.

3.15

The second step is the most important point because it provides a technical response to the problem introduced via the question mark in our [scheme](#). We have a tool also capable of dealing with the representation of the results and, most of all, specialized in object-agents, an object - in the Objective C sense - being a piece of program which understands messages and reacts to them.

3.16

A remark about the consequences of publishing the result of a simulation: Only if we are using a high level structured programming tool, it is possible to publish simulation results in a useful way. Quoting again from Swarm documentation:

The important part (. . .) is that the published paper includes enough detail about the experimental setup and how it was run so that other labs with access to the same equipment can recreate the experiment and test the repeatability of the results. This is hardly ever done (or even possible) in the context of experiments run in computers, and the crucial process of independent verification via replication of results is almost unheard of in computer simulation. One goal of Swarm is to bring simulation writing up to a higher level of expression, writing applications with reference to a standard set of simulation tools.

3.17

For this, the fact that the Swarm structure has two different levels is very useful. There is the model level (and we can have nested models of models, or swarms of swarms) and the observer level which considers the model (or the nested models) as a unique object to interact with, in order to obtain the results and to send them to various display tools and widgets.

3.18

Finally, the diffusion effect is a cumulative one, both for the production of reusable pieces of programs and for the standardisation of techniques to allow experiments to be replicated easily. However, standardisation is not always considered in the same positive way. See, for an example, the message of February 24, 1998, in the [Swarm mailing lists](#) archive which states that the Agilis Corp. - which is using Swarm to build enterprise simulation models - is rewriting it using Java vs. Objective C.

Objects And Agents: The Agent Based Chaotic Dynamic Emergence (ABCDE) Example

4.1

This example concerns an agent based experiment in the field of negotiation and exchange simulation. One can replicate or modify the experiment applying Swarm

(v.1.0.5 or above) to "make" the executable file *consumer* from the content of the archive compressed file [abcde.tgz](#).

4.2

The experiment shows the emergence of chaotic price sequences in a simple model of interacting consumers and vendors, both equipped with minimal rules.

4.3

Mainstream chaos supporters look for sets of equations that produce apparently non-deterministic data series when parameters lie in a particular range. But what about the plausibility of these synthetic constructions? In the ABCDE model we are not seeking to produce chaos: it emerges as a side effect of the agents' behaviour.

4.4

There are ten consumers and ten vendors; in other words, we have twenty agents of two types. Each agent is built upon an object, i.e. a small Objective C program, capable of reacting to messages, for example, deciding whether it should buy at a specific offer price. Both agents-objects-consumers and agents-objects-vendors are included in lists; the simulation environment runs the time, applying at each step the actions included in a temporal object (an action-group) and operates with the agents sending messages to their lists. There is a shuffler mechanism to change the order in which the agents operate and to establish random meetings of the members of the two populations.

4.5

At every simulation step (i.e., a tick of the simulation clock), artificial consumers look for a vendor; all the consumers and vendors are randomly matched at each step. An exchange occurs if the price asked by the vendor is lower than the level fixed by the consumer. If a consumer has not been buying for more than one step, it raises its price level by a fixed amount. It acts in the opposite way if it has been buying and its inventory is greater than one unit.

4.6

A simulated vendor behaves in a symmetric way: it chooses the offer price randomly within a fixed range. If the number of steps for which it has not been selling is greater than one, it decreases the minimum and maximum boundaries of this range, and vice versa if it has been selling.

4.7

In detail, we have the following steps:

1. At each time step t , each artificial consumer (an agent) meets an artificial vendor (another agent), randomly chosen.
2. The vendor fixes its selling price, randomly chosen within a small range.
3. The consumer accepts the offer only if the selling price falls below its buying price level.
4. At each time step t each agent (consumer or vendor) increases its transaction

counter by 1 unit, but only if it makes a transaction; it decreases it by 1 unit in the opposite case.

5. When their counters are less than -1, agents change their internal status: consumer-agents raise their buying price by a small amount; vendor-agents reduce the range within which they choose the selling price.
6. When their counters are greater than 1, agents change their internal status in the opposite direction: consumer-agents reduce their buying price by a small amount; vendor-agents widen the range within which they choose the selling price.

4.8

In this experiment, the starting points are a buying price level of 50 (on a scale from 0 to 100) and a selling price range from 45 to 55 on the same scale. Initially, all the consumers and vendors have the same parameters, but during the simulation, they evolve on an individual basis. One could say that the memory of the system lies in the consumers/vendors random interaction.

4.9

The result is that the mean price behaviour emerges as cyclical, with chaotic transitions from one cyclical phase to another. From a methodological point of view there are two kinds of emergence:

1. Unforeseen emergence: While building the simulation experiment, I was only looking for the simulated time required to obtain an equilibrium state of the model with all the agents exchanging nearly at each time: The appearance of a sort of cyclical behaviour was unexpected.
2. Unpredictable emergence: Chaos is obviously observable in true social science phenomena, but it is not easy to make a reverse engineering process leading to it as a result of an agent based simulation.

4.10

The results are reported in the [ABCDE price series](#). A typical chaotic phenomenon can easily be recognised from the trajectory of the mean offer price (the red line).

4.11

We have to remember that we are working here with twenty agents. According to the authors of [Growing Artificial Societies](#):

Our socioeconomic system is a complicated structure containing millions of interacting units, such as individuals, households, and firms. It is these units which actually make decisions about spending and saving, investing and producing, marrying and having children. It seems reasonable to expect that our predictions would be more successful if they were based on knowledge about these elemental decision: How they behave, how they respond to changes in their situations, and how they interact. In comparison to agent-based modeling, micro-simulation has more of a top-down character since it models behavior via equations statistically estimated from aggregate data, not

as resulting from simple local rules.

4.12

Can our experiments reproduce such a complexity, going from a toy world to the actual one? This is a methodological and philosophical question, but it is also a technical one. Are our simulation tools capable of dealing with large number of agents in a reasonable computational time? Explorations in this direction are lacking at present, but one suspects that the principal constraint involves hardware characteristics more than software specifications.

Note

¹In this case IBM is the company, not the acronym of Individual Based Models.

References

BELTRATTI, A., Margarita S., Terna P. 1996, *Neural Networks for Economic and Financial Modelling*. London: ITCP.

CONTE, R., Hegselmann, R., Terna, P. (eds.) 1997, *Simulating Social Phenomena*. Berlin: Springer.

EPSTEIN, M.E. and Axtell, R. 1996. *Growing Artificial Societies - Social Science from the Bottom Up*. Washington: Brookings Institution Press. Cambridge, MA: MIT Press.

GESSLER, N. 1997, Growing Artificial Societies - Social Science from the Bottom Up. *Artificial Life* 3: 237-42.

KIRMAN, A. 1992. Whom or What Does the Representative Agent Represent. *Journal of Economic Perspectives* 6: 126-39.

MINAR, M., Burkhart, R., Langton, C., Askenazy, M. 1996, [*The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*](#). Santa Fe Institute.

RUSSELL, S.J and Norvig, P. 1995. *Artificial Intelligence - A Modern Approach*. Upper Saddle River: Prentice Hall.

[Return to Contents of this issue](#)

© [Copyright Journal of Artificial Societies and Social Simulation, 1998](#)

