http://www.e-jemed.org/

ISSN : 1298-0137

e - JEMED
The Electronic Journal of Evolutionary Modeling and Economic Dynamics

Article number: 1013

Please cite this article as following:

Pietro Terna, 2002, Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming - By Benedikt Stefansson and Francesco Luna: A Review and Some Comments about "Agent Based Modeling", The Electronic Journal of Evolutionary Modeling and Economic Dynamics, n° 1013, http://www.e-jemed.org/1013/index.php

---

Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming - By Benedikt Stefansson and Francesco Luna: A Review and Some Comments about "Agent Based Modeling"

Pietro Terna

Dipartimento di Scienze economiche e finanziarie G.Prato,
Università di Torino, Italia

Abstract

There are three different "symbol systems" available to social scientists: the familiar verbal argumentation and mathematics, but also a third way, computer simulation. Computer simulation, or computational modeling, involves representing a model as a computer program. The key question is: What tools can we use in building our models, if we follow the "third way"? Simulation will have to be written in some Esperanto: it is obvious that the current Babel is against the emergence of a renewed enthusiastic effort in economic theory. Swarm is a library of functions offering tools in the middle between basic programming (Fortran, C, C++, Java, ….) and closed packages for dynamic simulation; it helps us to develop our own software, using a well-defined protocol and powerful tools to deal with agents' behavior, interaction and time sequences. So it can be considered an excellent candidate to play the role of this necessary Esperanto.

Keywords: Economic simulation, Swarm, Agent Based Modeling

JEL: B41, B59, C15

Copyright: Pietro Terna, 2002

Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming - By Benedikt Stefansson and Francesco Luna: A Review and Some Comments about "Agent Based Modeling"

Pietro Terna

Dipartimento di Scienze economiche e finanziarie G.Prato, Università di Torino, Italia
pietro.terna@unito.it

Introduction

Reviewing a book like that edited by Luna and Stefansson (2000) - mainly being one of the co-authors, as in my case - means getting involved in the general subject of model building and modeling techniques.

According to Gilbert and Terna (2000):

> Ostrom (1988) proposed that there are three different "symbol systems" available to social scientists: the familiar verbal argumentation and mathematics, but also a third way, computer simulation. Computer simulation, or computational modelling, involves representing a model as a computer program. Computer programs can be used to model either quantitative theories or qualitative ones. They are particularly good at modelling processes and although non-linear relationships can generate some methodological problems, there is no difficulty in representing them within a computer program.
>
> The logic of developing models using computer simulation is not very different from the logic used for the more familiar statistical models. In either case, there is some phenomenon that we as researchers want to understand better. This is the "target". We build a model of the target through a theoretically motivated process of abstraction (this model may be a set of mathematical equations, a statistical equation, such as a regression equation, or a computer program). We then examine the behaviour of the model and compare it with observations of the social world. If the output from the model and the data collected from the social world are sufficiently similar, we use this as evidence in favour of the validity of the model (or use a lack of similarity as evidence for disconfirmation).

The key question now is: What tools can we use in building our models, if we follow the "third way"? With Luna and Stefansson (2000), Introduction, p.xxiii:

> The volume Economic Simulations In Swarm has several goals:
>
> 1. it wants to propose a common language to those economists who already employ simulations as one of their tools of analysis;
> 2. it will present for the first time a language (Swarm) with a condensed, but rather exhaustive, tutorial;
> 3. it collects a rich variety of original contributions to economics, where simulations offer a natural mode of analysis of the dynamics of complex systems with heterogeneous agents.

Following Hahn (1991), Luna and Stefansson argue (p.xxiv, xxv):

> Perhaps, economics is destined to be *once more* an inductive discipline; whether it will be "softer" because of that (as Hahn implies) it is difficult to say. Undoubtedly, however, in

> their inductive efforts, theoreticians will find simulations of artificial "worlds" more and more useful to extract regularities and/or to push their "simple" thought experiments to "unforeseeable" outcomes. These results are very useful to help the intuition to the researcher and often suggest original interpretations of real world phenomena.
>
> Unfortunately, too often the acceptability of the frame of analysis and the transmissibility of its result are spoiled by the difficulty to *read* the model and hence by the "unreplicability" of the results reported. It is hence becoming pivotal to construct a common language, not dissimilar from French for diplomacy or more recently English. Simulation will have to be written **in** some Esperanto: it is obvious that the current Babel is against the emergence of a renewed enthusiastic effort in economic theory.

Swarm[1] is a library of functions offering tools in the middle between basic programming (Fortran, C, C++, Java, ….) and closed packages for dynamic simulation; it helps us to develop our own software, using a well-defined protocol and powerful tools to deal with agents' behavior, interaction and time sequences. So it can be considered an excellent candidate to play the role of this necessary Esperanto.

About simulation and social sciences, some key recent references are Epstein (1999), Axtell (2000), Tesfatsion (2001).

1. *What is Swarm and how to use it*

Following the Swarm documentation[2], but also 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: (i) the definition of computer based experimental procedure and (ii) the software implementation of the problem.

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.

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, Java, 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).

In the Swarm context, we use the Object-Oriented Programming languages Objective-C or Java. According to the Swarm documentation, computation in a Swarm application takes

[1] http://www.swarm.org.

[2] Refer always to the Swam home page http://www.swarm.org.

place by instructing objects to send messages to each other. The basic message syntax is (Objective C):

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

or (Java[3]):

```
targetObject.messageArg1$Arg2(var1, 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.

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.

What those objects contain? This is the crucial question to understand what Swarm is: They can contain anything we put into their object oriented code, from simple rule execution to highly complicated logic and mathematics (artificial neural networks, classifier systems, genetic algorithms, …). We are completely free in using simple agents, like passive performers of fixed rules to full BDI (Beliefs, Intentions, Desires) cognitive agents (obviously, writing the specific code). We are nevertheless always completely able to manage all the details (as interactions, schedules of events, …) of our experiments, what is non completely feasible with other tools, directly devoted to BDI development, like SDML[4].

Technically speaking, 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.

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

[3] Where the *$* signs represent an added notation in Swarm (it is not a Java convention) to keep memory also in the Java version of Swarm, of the original position, and meanings, of the parameters in the more explicit Objective C notation.

[4] http://www.cpm.mmu.ac.uk/sdml/

> 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.

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. Figures 1 and 2 are related to the different levels of Swarm (figures come from Minar et al., 1996).

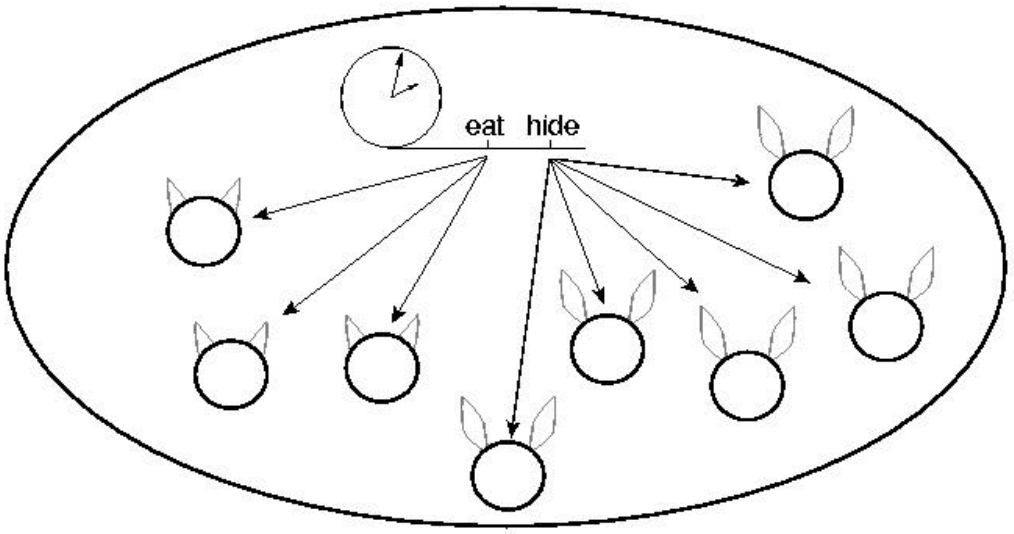

Figure 1: a Swarm intented as ModelSwarm (composed of agents).

We finally underline here the key role of Swarm, because the characteristics of the software we use are crucially important in assuring the success of this way of formalizing models. Only if we use high quality software we are able to communicate the details of our model, allow other scholars to replicate the results, and avoid difficulties in modifying poorly written code. The best way to improve the quality of the programming is to choose an object-oriented language. This choice simplifies the translation of the 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 the program. In addition, in a fully object oriented environment, events (or time steps) can be organized as objects. The key term here is object: a piece of code containing data and rules operating on them. A these characteristics are contained in Swarm.

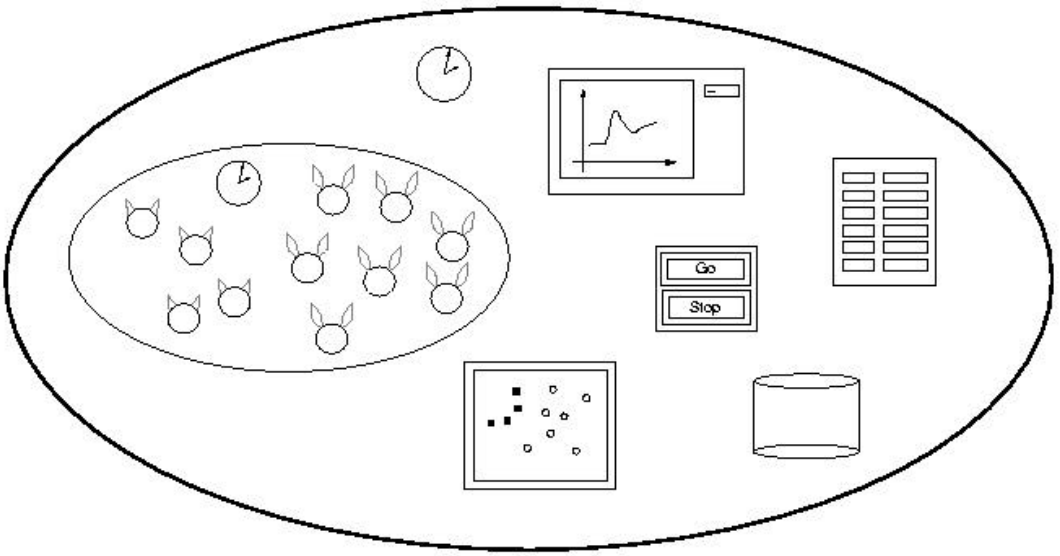

Figure 2: a Swarm of observer agents measuring a model

2. *An overview of the models contained in the book*

All the Swarm codes related to the papers contained in the book can be downloaded from the Swarm site[5].

The first part of the book ("The Grammar") reports a concise and extremely useful tutorial of Swarm (Objective C version); the presence of the tutorial is strategic in the first book devoted to Swarm, also because we effectively lack at present of a complete tutorial of Swarm (obviously, we do not forget the original Langton 1997 tutorial[6], which Staelin (2000) is now re-writing in Java).

The second part (". . . and Some Prose") collects a series of paper written for the book using Swarm to investigate their subject of interest.

From Luna and Stefansson (2000) p. xxx, we read:

> The first contribution of the collection, *Economic Experiments with Swarm: a Neural Network Approach to the Self-Development of Consistency in Agents' Behavior* by Pietro Terna, investigates with much sensibility, the methodological implications of agent-based models for social sciences in general and economics in particular. Methodology and techniques are "largely under construction" according to Terna, who proceeds to present a *generalized Environment-Rules-Agent* scheme as a tool to assist the researcher in the design of a "bottom-up" model. A package **bp-ct** is proposed as an easy way in building and running artificial laboratories for social scientists. The package is finally employed to produce a spontaneous no-auctioneer Hayekian market.
>
> Charlotte Bruun and Francesco Luna present a model of *Endogenous Growth with Cycles in a Swarm Economy*. The term "Endogenous Growth" will appear misleading to the reader who expects to find a model of sustained growth in line with New Growth

[5] Direct link: ftp://ftp.swarm.org/pub/swarm/src/users-contrib/anarchy/lunabook/index.html.

[6] The tutorial is always packed in the Swarm Applications file that you can download from http://www.swarm.org.

Theory. As a matter of fact, the artificial economy described eventually reaches a "steady state". However, the trigger for the take off is hard-wired in the behavior of self-interested agents who face a complex environment which they try to tame. The emergence of original entrepreneurs lead to the growth of this "Schumpeterian" economy. Business cycles are caused both by the failure of sub-efficient firms and by the change in wealth distribution in this pure-credit economy. The model appears flexible enough to address an ample spectrum of issues, by modifying only marginally the original code: one of the great advantages of the Object-Oriented paradigm.

Luigi Mittone and Paolo Patelli present a model of *Imitative Behavior in Tax Evasion*. Thanks to their simulation approach they manage to tackle effectively a series of criticisms addressed to previous models. Psychological motives and experimental results are taken into consideration and integrated in the simulation framework to analyze the effects of "framing" and of "moral suasion" on tax-evasion decisions. Furthermore, the authors construct a model of cultural evolution: tax evasion can become the general attitude under some particular imitative and monitoring/enforcing behavior.

The spontaneous emergence of financial intermediation is the object *of An Experimental Approach to the Study of Banking Intermediation: The banknet Simulator* by Massimo Sapienza. The author builds his model on a pre-existing one, "BankNet," initially designed by one of the original Swarm programmers as an "exercise". The model is then employed to analyze a somewhat different topic. Here is an example of the re-usability of Object-Oriented code and of the possibility of "building upon" a pre-existing model. Transaction costs, economies of scale, agents' heterogeneity, and strategic interaction are the "real world" features that, according to this model, lead to the endogenous creation of intermediaries. One of Swarm's library allows for the "real time" graphical representation of the credit links being established in the system. Such (completely decentralized) emergent phenomenon is hence monitored very effectively.

Moving to a different issue in finance, *Numerical Modeling, Noise Traders, and the Swarm Simulation System* by Timothy E. Jares investigates the persistence of noise traders observed in the real world, a phenomenon that contradicts the predictions of traditional models. The presence of such traders is shown to affect long-term prices. Wealth appears to play a secondary role with respect to the configuration and characteristics of market institutions. Noise traders are shown to disappear when their beliefs are not highly correlated so that fundamental traders will dominate.

Shifting now to Industrial Organization, *Nonlinear Stochastic Dynamics for Supply Counterfeiting in Monopolistic Markets* by Marco Corazza and Alessandro Perrone, studies the strategic interaction between a monopolist and a counterfeiter from a dynamical perspective. Initially, some closed form solutions are obtained for the model proposed. It has to be noticed that this model becomes not only *computable*, but also tractable thanks to the appropriate choice of the demand function. Still, the inherent complexity due to nonlinearities and the presence of stochastic disturbances requires a simulation investigation of the possible outcomes. The authors (for the first time) implement in Swarm a graphical representation of a state-space dynamics.

Using Swarm for Simulating the Order Fulfillment Process in Divergent Assembly Supply Chains is the title of the contribution proposed by Fu Ren Lin, Troy J. Strader, and Michael J. Shaw. The effective management of a supply chain requires the coordination of different and sometimes contradictory interests. A multi-agent simulator can turn out to be a powerful support in the decision-making process. In particular, the effect of information sharing on order fulfillment in divergent assembly supply chain is found to attain a substantial reduction in inventory costs while maintaining acceptable order fulfillment cycle times. In other words, information as a source of enhanced coordination and uncertainty reduction is a substitute for inventory.

Christoph Schlueter-Langdon, Peter Bruhn, and Michael Shaw present *their Online Supply Chain Modeling and Simulation*. They analyze one of the most dynamic markets of this last few years: the online services market. Digital interactive services are still an

> industry in its infancy and rapidly evolving. What are the most effective strategies to deal with entry decision in such an environment? The approach followed by the authors is to design an advanced "flight-simulator" implemented in Swarm to be used as a sophisticated aid to decision making. Also in this case it is interesting to discover that the original version of this model was built upon two pre-existing programs: the so called "Anazasi village" and the supply-chain model by Fu-Ren Lin.
>
> The last contribution, *The Coevolution of Human Capital and Industrial Structure* by Francesco Luna and Alessandro Perrone is mainly designed as an exercise for the "interested reader". The model tries to depict the dynamical interaction between the structure of enterprises (as captured by their size) and human capital represented by the skills available in the pool of workers in the economy. A successful entrepreneur can share with her/his workers part of her/his profit. Once an imitation process is introduced among workers, these initially successful "professions" will spread among workers modifying the pool of human capital from which new entrepreneurs will draw their work-force. The ex-ante distribution of skills is hence modified and evolves "interactively" with the size distribution of firms. Certain regularities are pointed at and hypotheses formulated. Unanswered questions are left to the reader to be investigated.

So, an impressive series of very different applications, showing both the flexibility of the simulation technique based on agent models and the usefulness of Swarm as a general tool.

Reading the book we can fully appreciate the crucial capability of Swarm, which is a tool which rather than giving a ready-made structure, offers a rich set of libraries to be combined as the researcher thinks it more appropriate to present the model.

This aspect guarantees the flexibility necessary to a creative effort. At the same time, the use of those libraries will made each component of the model perfectly intelligible to other scholars. In particular, all graphical interfaces are tested and their use and interpretation is made patent; a series of pseudo-random numbers for very different distribution are given so that even those simulations relying on random numbers can be replicated.

A new volume is now appearing (Luna and Perrone, 2001), that continues the development, with Swarm, of this kind of agent based applications to economics and finance[7].

3. *Methodological considerations about agent based models: (a) putting a "mind" into the agents*

A new question now is: if our computer simulation model is based upon agents (e.g. built with Swarm, as the models presented here), to what extent must our agents be complicated? Should we provide them with a "mind"? The answer ranges from Axelrod's simplicity principle (Axelrod, 1997) to the use of full BDI (Beliefs, Intentions, Desires) cognitive agents.

Looking to the contents our collection of papers (Luna and Stefansson, 2000) the following statements may constitute a tentative conclusion:

- we can fully accept the Axelrod simplicity principle, since it is easily shown that also very simple agents can generate complex emerging structures;
- more complicated agents facilitate the emergence of theoretical results, as the Hayekian market in Terna paper.

The general idea is that in many cases the complexity of the economic situation is explained by the interaction of simple agents, mainly if behaving in a structured environment. So the question is: Are human agents so far from the complexity of the economic system, as ants are from their anthill? We can also see - and this is quite obvious - that in many cases

[7] With papers on: new Swarm based tools; financial simulations; economic simulations.

only with the presence of a micro "mind" a sort of intelligent behavior, useful to the agents, can emerge. The presence of "minded" agents affects the market, modifying the complexity of its structure.

From the external point of view of the social scientist observing actual phenomena this is a puzzle difficult to solve! Have we always to search for the effect of agents' rationality ("using" their mind) or can we open our minds and accept the Axelrod's (1997) KISS principle? KISS stays for Keep It Simple, Stupid; Rosaria Conte in a discussion of ICCS & SS II in Paris, last September, proposed to reinterpret it as Keep It Simple as Suitable; my counterproposal is: Keep It Sufficiently Simple (to correspond to real life agents.)

4. *Methodological considerations about agent based models: (b) reductionism or methodological individualism?*

Epstein and Axtell (1996 Chapter 1., par. "Beyond Methodological Individualism") state:

> Our point of departure in agent-based modeling is the individual: We give agents rules of behavior and then spin the system forward in time and see what macroscopic social structures emerge. This approach contrasts sharply with the highly aggregate perspective of macroeconomics, sociology, and certain subfields of political science, in which social aggregates like classes and states are posited *ab initio*. To that extent our work can be accurately characterized as "methodologically individualist." However, we part company with certain members of the individualist camp insofar as we believe that the collective structures, or "institutions," that emerge can have feedback effects in the agent population, altering the behavior of individuals. Agent-based modeling allows us to study the interactions between individuals and institutions.

Methodological individualism, which is at the bases of agent based models, is not widely accepted. On one hand, this position is criticised by methodological organicism or holism, who deny the possibility of understanding any social or economic system by studying its components. On the other hand, it is criticised also by scholars rejecting the reductionist choice of the so-called representative agent, highly structured in terms of rules, optimisation ability, knowledge of data and of the economic model underlying the actual world (paradoxically, as Sargent observes (1993), the same model unknown to the econometrician studying the economic structure where the representative agent is supposed to act).

Misunderstandings emerge especially when the first type of scholars - those denying the interest of the representative agent based methodology, being not possible to disregard structures (classes, institutions, ...) - rejects the experiments built with simple agent models, again to refuse the same type of reductionist choice.

Instead, we can explicitly admit the role of institutions and structures, looking for the emergence of them, and considering essential the heterogeneity of agents (always, if possible, simple), to experiment the non linearity of aggregated effects of their behaviour. We have to remember that not necessarily the whole corresponds to the sum of the parts in the presence of interactions: This is the main source for the emergence of complexity and here we find the added value of artificial experiments with agent based simulation models.

Methodological individualism and agent based models are therefore non coincident fields. With agent-based models we study the emergence of structures, institutions and behaviour, previously unexpected or unpredictable, looking for many interpretation levels about projects built on several layers, all interacting one with the other. We can also have agents made up of more simple parts (e.g., an enterprise and its units, with composing agents); a model can provide for the interaction among entities made up of agents themselves, and so on (in Swarm

terminology, we can have swarms of swarms); we can also use fruitfully such methodologies in the organisation domain.

Studying the emergence of phenomena we use complexity as a two-way process, where the non linear interaction of agents produces social effects and the emerging social structures (via evolution, genetic selection, co-determination) affect the agents' behaviour.

Summarizing, and to state it quite provocatively: (i) we say no to the reductionism of methodological individualism that attempts to reproduce the complexity of the system directly in the agents, complicating them to the extreme, and overburdening them with knowledge and abilities; (ii) we say no to organicism, which considers the agents' approach unfeasible, based on the belief that there are no structures, at the agent's level, able to reproduce the complexity of reality; (iii) we say yes to an approach based on simple agents, that tries to find in interaction and in ex ante structures (if any) the way to generate complex landscapes, comparable to the real ones. Obviously we have to look to this complex situations at the appropriate level, which is not necessarily that of each specific agent.

5. *Unifying the structure of simulation models*

Coming back to the Babel problem and to the Esperanto perspective (see the Introduction), we have to look in a more deep way at the use and reuse of the code of our simulation models. As recalled in Bruun (2000), in his discussion about programming simulation models, Axelrod (1997) claims that the programming of a simulation model should achieve three goals:

- *validity*: the program must correctly implement the model;
- *usability*: it must be easy to run the program, interpret the output and understand how the program works;
- *extensibility*: It must be possible for future users to adapt the program for new uses.

With Bruun, the goal that we shall concentrate on here is the third goal, which may be divided into:

- *reuse of code*: increase efficiency of model building and reduce bugs.
- *modularity*: makes it easier to understand the code, and reuse elements.
- *increase communication*: only make new models when there is a need to.

The project proposed in Bruun (2000) constitutes a key passage to achieve an effectively large diffusion of agent based simulation techniques, hopefully using Swarm.

> There is, however, also the possibility of extending Swarm specifically for economic simulations. This could be done by making an economics library. If we take a look at the latest developments in the software industry, there has however not been full contentment with the library solution. Reuse of code is simply not promoted enough by making components available in a library. The structure in which the elements from the library is to be integrated must also be taken into consideration. From the software industry the suggested solution bas been frameworks and patterns.
>
> By making an economics framework to Swarm one may enable the model-builder to add much more than water to a pre-defined structure. Components already existing may be combined to make a new model, but for special features the user may apply his own components to the pre-defined structure.
>
> Possible gains from using a common framework:
>
> 1. Make it easier for newcomers to construct their first model, and make it easier to replicate models.

2. Develop certain standard ways of handling different auction types, different ways of performing bankruptcies etc. This will case model construction but also increase communication and reduce the risk of bugs.
3. Make it possible to test a model against different model components. It will be easier to test a models sensitivity to specific model elements, e.g. bankruptcy rules or behavioral assumptions.
4. Make it easier to provide microeconomic models with a macrofoundation. As a way of testing partial models, they may be completed (closed) by adding standard components from a framework, and in this way cheek feedback effects from the macrolevel.

By having a framework within the agent-based computational economics community, some standard models for addressing families of problems could emerge and a positive feedback mechanism set of. Models can no longer be accused of *being one damned thing after the other* since modelers extends and modifies each others models in order to reject or strengthen their conclusions. In this way the community can accumulate knowledge in a set of conclusions that are developed, not by one modelbuilder in one model, but by a number of modelbuilders making a number of variations of a given model.

6. *Swarm perspectives*

As a conclusion, and coming back to our simulation tool, we have to ask to ourselves wheter Swarm can effectively become the Esperanto for agent based simulation, or for social science simulation at all.

A recent message from the Swarm announce list[8] shows a lot of important novelties for the future of the project, manly in the direction of Bruun design.

1) Swarm models will be sharable as self-contained documents.
2) Describing the "big picture" for a model will not involve programming. In simple cases, the "little picture" (e.g. "step" methods) will not involve programming, either.
3) Swarm models can run inside a web browser (. . .). Swarm is already equipped to run JavaScript Swarm models off the web.
4) Swarm models can reference subcomponents via the web. For example, an ObserverSwarm can sit on website A and the ModelSwarm it manages could be on website B. When run, it would all be built and run inside the users browser. There are all sorts of potential elaborations on this: one might be using popular search engines to pick up compatible subcomponents to use. Real-time collaborative work environments are another possibility.
5) Swarm models can reference subcomponents developed outside of the Swarm community. For example, there is already an genetic algorithm optimizer module.
6) Swarm modelers can move in a low-investment way from Swarm to different scheduling paradigms (and back!)

With this kind of development Swarm will break the main barrier preventing a complete diffusion of these techniques, i.e. the necessity of being able to write code, to assemble it, to look for bugs etc. with a substantial advance in spreading the knowledge emerging from artificial experiments and simulation.

[8] Check http://www.santafe.edu/projects/swarm//archive/list-archive.0010/0079.html.

References

AXELROD, R. (1977). Advancing the Art of Simulation in the Social Sciences, in R.Conte, R.Hegselmann and P.Terna (eds), *Simulating Social Phenomena*, Lecture Notes in Economics and Mathematical Systems 456, pp.21-40, Berlin: Springer.

AXTELL R. (2000). *Why agents? On the varied motivations for agent computing in the social sciences*. Center on Social and Economic Dynamics, http://www.brookings.edu/es/dynamics/papers/agents/agents.

BRUUN, C. (2000). Prospects for an Economics Framework for Swarm, work in progress, at http://www.socsci.auc.dk/~cbruun; forthcoming in Luna and Perrone (2001).

EPSTEIN J.M. (1999). Agent Based Models and Generative Social Science. *Complexity*, IV (5), pp.41-60.

GILBERT, N. and TERNA P. (2000). How to build and use agent-based models in social science, *Mind & Society*, no. 1, pp.57-72.

HAHN, F. (1991). The Next Hundred Years. *Economic Journal*, 101, pp.47-50.

LUNA, F. AND STEFANSSON, B. (eds.) (2000). *Economic Simulations in Swarm: Agent-Based Modelling and Object Oriented Programming*. Dordrecht and London: Kluwer Academic.

LUNA, F. AND PERRONE, A. (eds.) (2001). *Agent-based Methods in Economics and Finance: Simulations* in *Swarm*. Dordrecht and London: Kluwer Academic.

MINAR, M., BURKHART, R., LANGTON, C., ASKENAZY, M. (1996). *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*, Santa Fe Institute, http://www.santafe.edu/projects/swarm/overview.ps.

OSTROM, T. (1988). Computer simulation: the third symbol system. *Journal of Experimental Social Psychology*, vol. 24, 1998, pp.381-392.

SARGENT, T., J. (1993). *Bounded Rationality in Macroeconomics*. Oxford: Clarendon Press.

STAELIN, C., P. (2000). *jSIMPLEBUG: a Swarm tutorial for Java*, forthcoming.

TESFATSION, L. (2001). Introduction to the special issue on agent-based computational economics. *Journal of Economic Dynamic and Control*, Vol. 25, Issue 3-4, pp.281-293.