

AGENT-BASED DESIGN MODELLING USING THE VISUAL SWARM BUILDER

ALESSANDRO PERRONE
Universita' di Venezia Ca' Foscari Italy

Abstract. In this paper we describe a new product written to simplify the construction of Agent based simulations called Visual Swarm Builder. VSB is an integrated environment for the design and realization of simulations using Swarm libraries, based on menu-driven and graphic selections of objects like graphs, “standard agents”, etc. This paper is divided into three sections. It commences with an introduction about the need of an instrument like VSB, then there are parts dedicated to the simulation environment, Swarm, and finally there's the last parts in which there's described the package, with all the present and future features, and a short example designed with VSB.

1. Introduction

The need of a common language for agent-based economic simulations is, by now, widely recognized and several attempts are currently underway to identify the fundamental characteristics that such language should have. Between the two extremes of “just add water” and “do it yourself” several options have emerged in the last few years. Swarm, Ascape, Repast, Starlogo, AgentSheets and few other simulation platforms are, as of today, the most credible contenders.

From the on-going discussion, a new trade-off has emerged. It is not anymore a matter of deciding whether the researcher should be constrained by the rules of an omni-comprehensive software environment (the “just-add-water” option) or, rather s/he should painstakingly construct every soft-brick of the artificial world modeled (the “do-it-yourself” alternative). Rather, the trade-off is between the ease of programming on the one hand, and the sophistication of the obtainable end-product.

Swarm in particular has attracted substantial interest from the economic profession (see for example Luna and Stefansson 2000), but is now challenged by simulation platforms like Ascape and Starlogo which

are proposed as more “user-friendly” reducing the initial investment for the professional economist (but amateur programmer).

Seldom in the economic history of the introduction of innovations, the adoption of a particular new technology over a contender has been dictated by the ascertained superiority. More often other factors like network effects and hysteresis played a more fundamental role. In a decentralized fashion—typical of the GNU philosophy which has inspired the design and diffusion of Swarm—several researchers have worked to make Swarm meet the challenges launched by other contenders to the role of esperanto of agent-based simulations.

For example, MAML¹ is a meta-language offering a set of commands whose wording and syntax is much closer to that of a “natural language” and, at the same time, will free the user from the need to keep up with the subsequent Swarm versions. With a different goal, SWIEE² enlarges Swarm’s possibilities building around it an environment amenable to the design and management of economic experiments. The interactions of human subjects will be observed in a laboratory-like situation (over the internet). By using Swarm as the background (and playing field) of the experiment individual actions and simulations will merge in a new tool for the study of economic behavior.

The next step in this quest for simplicity is offered by Visual Swarm Builder. VSB is an integrated environment for the design and realization of Swarm code based on a menu driven selection of objects like schedule of events, graphics, lists and arrays etc.

A visual building tool is clearly a useful extension to any programming environment, the more so for a large, complex and powerful simulation platform like Swarm. The absolute beginner as well as the expert programmer will find VSB useful. Because of its ease of use the former, and thanks to the speed with which a prototype can be obtained, the latter.

The rest of paper is devoted to the “step-by-step” construction of a model *f*. The code will first be presented in its Swarm “native” form and then replicated with simple “drag and drop” mouse actions plus very simple editing interventions on the generated source. The model will be generated around some graphical objects (histograms, XY line graphs and time series graphs) and the forecasting activities of a boundedly rational agent.

¹ For further info, see <http://www.syslab.ceu.hu/maml>

² For further info, see <http://swiee.econ.unito.it>

2. The Swarm Environment

According to original programmers of Swarm, “Swarm is a multi-agent software platform for the simulation of complex adaptive systems. In the Swarm system the basic unit of simulation is the swarm, a collection of agents executing a schedule of actions. Swarm supports hierarchical modeling approaches whereby agents can be composed of swarms of other agents in nested structures. Swarm provides object oriented libraries of reusable components for building models and analyzing, displaying, and controlling experiments on those models. The SWARM tool was developed at the Santa-Fe Institute with the intention to create a simulation tool for agent-based models. Concerning the people who use these kind of models, which often are not computer scientists, the main goal for the Santa-Fe people was to make an easy-to-use standard application. Chris Langton et al., who developed this tool, claim that SWARM offers the opportunity to focus on the model itself, instead of the implementation issues.

The fundamental concept of SWARM is the “swarm”. Agents are found as elements in a swarm, but this swarm may be part of another swarm, due to theoretically infinite nesting of swarms. This offers a good way to simulate different parts of a complex domain in terms of sub-entities. This is very convenient for biological models. As an example, just think about bees, ants, birds or roe deer. All these animals are found in a forest, the complex domain, but each species form own entities, which either are sub-populations of the same kind, gender dependent groups or different sub-species.

SWARM consists of libraries written in the object oriented language ObjectiveC and in Java. The object oriented concept is found throughout the implementation and you are offered an easy way to base your model on the given methods and objects. In order to present the model translation from a real world problem to a swarm model.

The main features of Swarm are:

- *General work.* Swarm imposes only minimal substantive restrictions on the modelling enterprise. It is necessary to represent individual agents by objects, but everything else is open to customization. As a result, Swarm is a useful environment for modelling in fields ranging from political science and economics to ecology and biology;
- *Multilanguage.* Routines of Swarm libraries can be accessed from programs written in Objective-C (the hystorical Swarm Language) and Java. Development versions of Swarm can be accessed from C++, COM Object and javascript, as well;
- *Multiplatform.* The package can be compiled on almost any Unix-like systems and in Windows platforms;
- *Simulation Management.* Swarm's activity libraries provides

scheduling tools to support "bottom-up" modeling of complex systems and emergent phenomena. A range of possibilities can be explored, including:

- totally decentralized behavior of agents who decide when and if to act
 - nested ordering of agent actions;
 - conventional cellular automata;
- *High Quality Support Libraries.* State of the art library support for generation of random numbers and simulation of draws from statistical distributions;
 - *Graphical Input/Output Interaction.* Graphs and displays can monitor a simulation as it proceeds. Code can be designed to allow user interaction with simulation (e.g. create new agents, reposition agents, change their attributes);
 - *Large Variety of example Applications.* The SDG provides a package of supported example programs as well as a Swarm Tutorial³

As shown in Figure 1, a simulation program based on SWARM schematically contains three types of objects (classes):

- 1) A ModelSwarm that creates and controls the activities of agents in the model;
- 2) An ObserverSwarm that collects information from the agents (observations) and outputs them to a file or to graphs. these graphs are nearly automatically generated by this module, using the corresponding elements from the SWARM library;
- 3) Classes for different agents in the model (firms, consumers) and for some aggregate agents (markets, industries, economies) that regroups actions and aggregate properties (i.e. market price, concentration index of the industry).

First two types are relatively standard even if they should be adapted to the particular structure of your model. the last type contains the very agents of your model and you can enrich them in an incremental way: beginning with very simple agents and adding more complex behaviour for agents that inherit from them.

3. The ideas behind VSB

In some sciences, especially in the study of complex systems, computer programs play an important role as scientific equipment. In the case of computer simulations, the programs under use, can be seen as

³ All the material can be downloaded free from SDG web site (<http://www.swarm.org>) or from the author new web site (<http://www.alessandroperrone.it> in Swarm section)

experimental devices built in software. We don't, however, forget that social scientists are not computer scientists, but their skills in the field will have to improve to cope with the growing subject of social simulation and agent based modelling techniques.

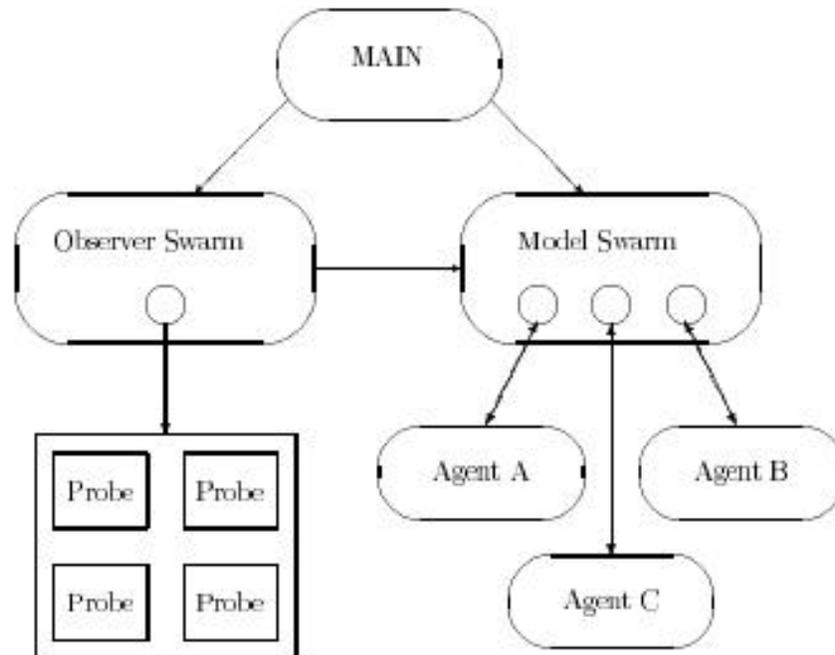


Figure 1. Architecture of a standard swarm simulation model

The Swarm project can be viewed as an aid to researchers to create a standard tool to build simulations. But there's a problem. Programming with Swarm is not easy, the scientist have to spend a few hours (50 hours) to learn the philosophy of the package, how to use the libraries, how to call the graphic routines, how to call data to plot, etc. etc, and how to program in objective-c. Actually, there's also a java version of Swarm and the learning time may be decreased a bit, because it is very easy to find a java programmer, but the problem of learning the use and the philosophy of Swarms still remains.

For this reason we have thought and programmed a package named VSB (Visual Swarm Builder). It can be viewed as the first visual Rapid Application Development (RAD) tool designed exclusively for the Swarm Simulation toolkit. It is a complete development environment that provides the researcher with a rich set of tools already coded and

components necessary to develop and deploy high-performance simulations.

The way to program, during these years, has changed a lot, because of the availability of visual languages. With simple “clicks of mouse” everyone can build simulations, even if the potential user does not know anything about the language he is using. The role of the “pure programmer” is not the same as before, he is not an extraterrestrial being, who can, writing a few lines of code, build programs from scratch.

One of the advantages of using visual tools, is that the programmer does not have to write always the same boring code, he may just input a few parameters, and then the Visual tools will fill the other pieces of code. For example, in Swarm, if the simulator wants to insert a “bar chart” in her/his own program, the user has to fill only the dialog on Figure 2 and, once VSB generates the code, the compiler will generate the “boring” code relative to the bar-chart.

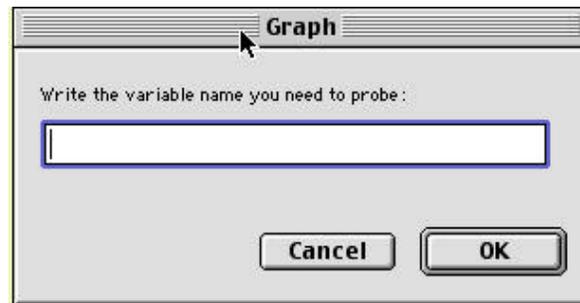


Figure 2. Bar chart dialog box

Every program is composed of 50% of the interface, 30% of problem analysis, and 20% of code typing. VSB let the user to concentrate on the 30%+20% of the program development, while it takes care of the other 50%.

VSB has been programmed in Macintosh environment, and it runs on every MacOS from 8.5 to new MacOSX.

4. Working with VSB

VSB is the first Rapid Application Development (RAD) environment for creating simulations using Swarm libraries. It offers a full suite of integrated tools, including a project manager, code editor, and a “standard preview of the final simulation”. For many tasks, you do not have to work with raw code at all.

In this section, the reader will be guided to the programming of simple Swarm simulations (without writing any lines of code) to a fairly simple financial simulation.

4.1. GETTING STARTED

To get started, first click on the VSB icon in the Finder. Once the application is started, following screen will appear, Figure 3.

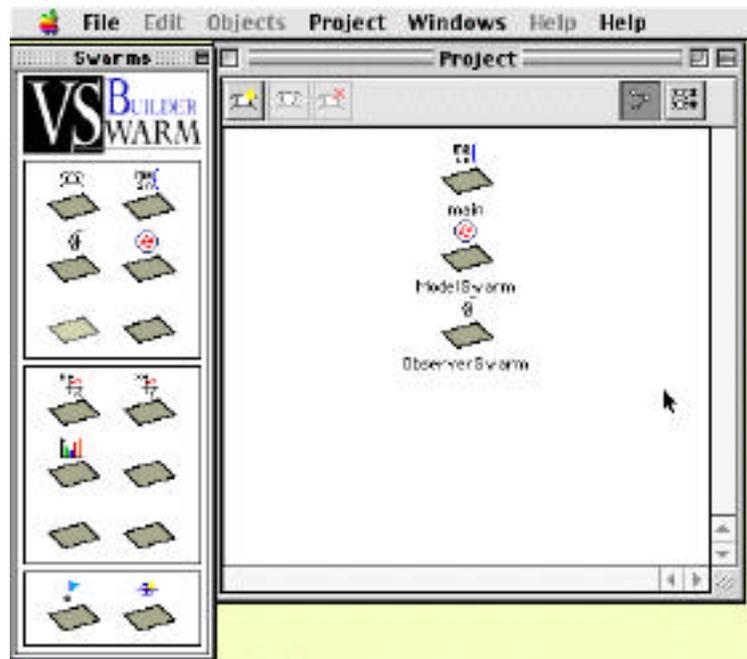


Figure 3. Initial screen of VSB

The screen is divided into

- 1) a **menu** bar with the usual Macintosh programs item;
- 2) a **Project Window**, that contains the collection of items that make up the particular simulation you are developing. The project window displays a list of these elements to give easy access to them. In a standard simulation project we have, for example, the main, the ModelSwarm, and the ObserverSwarm objects. In the project you can insert as many objects as you want;

- 3) a **Control Window (Swarms one)** that displays a group of various UI objects that can be used to build an interface/simulation.

These UI objects use the "drag and drop technique", so, to use one of these, simply click on it, and drag it to the UI in project window. The control will be added to your main window, and specifically to the relative object.

4.2. WORKING WITH PROJECTS

All of the windows, probes, objects, graphs and their programming code that make up a single simulation are stored in a Project window.

Projects simply give you a convenient way to organize the objects that make up your application.

Projects can contain any of the following items

- main object
- ModelSwarm
- ObserverSwarm
- SwarmObject
- Other kind of objects

VSB permits to insert into a simulation, with the above described drag&drop technique, the following items

- Graph XY
- Graph X versus time
- Bar Chart Graphs
- Initial Values
- New methods

Double-clicking on an item in the principal project window or in the subproject window, will either display the item or open it for editing.

4.3. WORKING WITH THE PALETTE WINDOW

The palette allows to pick widgets to be added to current active VSB project. There are three major modes of operation.

- The first mode is the clicking of the UI under the Palette window. In this mode a short help of the UI appear.
- The second is the widget placement. It permits to insert a widget to the project window. To add an element in the Project window, simply select the element in the Palette and drag it into the Project Window near the master object the element refers to. Once the element is in the right position, an alert will ask to fill the parameters relative to the element.

- The third mode of operation is the click of a widget within the project window. This will cause a dialog box to appear. It contains the parameters of the object and the button "view code" which permits to modify directly the code of the UI selected. In this mode of operation, the author can change the values of the parameters, or, if s/he wants, s/he can change directly its code.
A list of all widgets supported is shown in Figure 4.

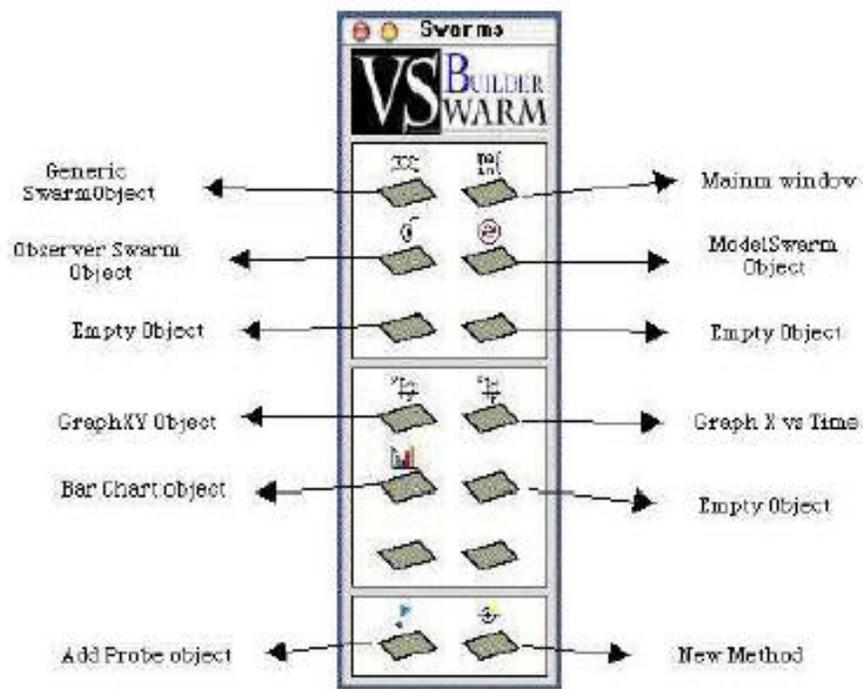


Figure 4: The widget supported in the palette window

5. Visual Swarm Builder Experiments

5.1 FIRST EXPERIMENT WITH VSB

This section takes the user along the steps required to create a simple simulation shown in Figures 5 and 6.

Here are the steps

- Create a new project by double click on the New item on File Menu
- Try to replicate the following structures on the project window using drag&drop technique
- Run the project using the Run command from Project Menu

- Generate the code relative to the simulation using “Generate Code” item in Project menu
- Compile and run the first simulation on you favourite Swarm environment.



Figure 5. First Experiment

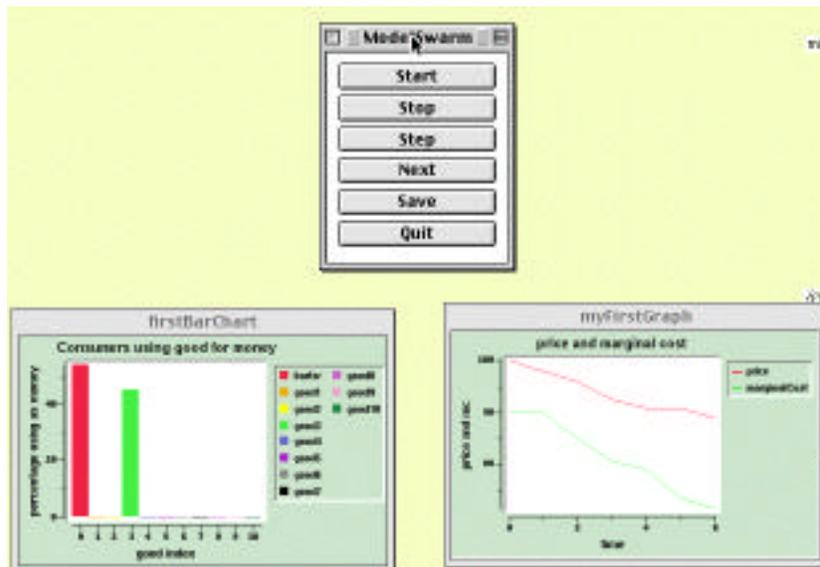


Figure 6. Output of first experiment

5.2 FIRST EXPERIMENT WITH AGENTS

In this section we develop, a real simulation, using a few tools of the VSB, such as the code editor inside.

- open a new project
- insert a graphXvsTime object into the project window
- create an empty object named “NASDAQ”
- dit the ObserverSwarm by a double click on its icon. The code editor will appear, and change the following lines of code, Table 1.

TABLE 1 Code editor

| Original Lines | Lines to be substituted |
|--|---|
| <pre>-(float)getValue { float valore; valore=[uniformDbIRand getDoubleWithMin: 10 withMax: 100]; return valore }</pre> | <pre>-(float)getValue { float valore; valore=[[modelSwarm getNasdaq] getMyValue]; return valore }</pre> |

- Build the code, compile it and run it.
The output is shown in Figure 7.

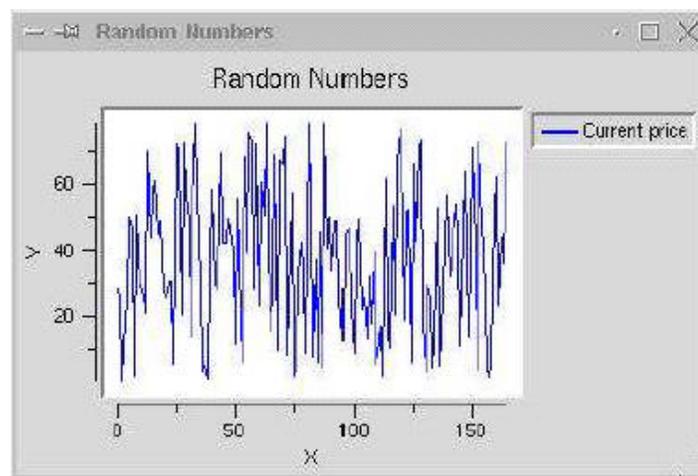


Figure 7. Output of the experiment

Now the potential user has all the preliminary information to build its own simulation using VSB and Swarm libraries. All the examples can be

found into the "Examples" folder in VSB package. We have inserted, also, an evolution of last model, in which there are two stock index (nasdaq and Dj), their projection and their percent variation using not only a single line graph, but also a barchart and a multiple lines graph. This example is a bit hard to explain in a few lines, because we had to "patch" a few objects and several lines have been changed. All the examples are fully explained in every part and before inserting the new lines, the old ones are commented.

6. Future Work

The package is a very dynamic work in progress. For an update on the environment development, to download the latest release or for updated information about the VSB, including sample code, errata and preview of further versions, visit the site <http://www.alessandroperrone.it> in Agent Based simulations section.

Acknowledgements

I would like to thank all the people starting the project of VSB, in particular the original programmer, Marco Tenuti.

Bibliography

- Apple Computer Inc: 1985, *Inside Macintosh*, Addison Wesley, Reading, MA.
- Epstein, JM and Axtell, R: 1996, *Growing Artificial Societies from the Bottom-Up*, MIT Press, Cambridge, MA
- Boero, R: 2002, <http://swiee.econ.unito.it/>.
- IBM,: 1995, *Visual Age for Java, version 2.0*
- Luna, F and Stefansson, B (eds): 2000, *Economic Simulations in Swarm: Agent- Based Modeling and Object Oriented Programming*, Kluwer, Dordrecht, Netherlands.
- Luna, F and Perrone A (eds): 2001 *Agent Based methods in Economics and Finance: Simulation in Swarm*, Kluwer, Dordrecht, Netherlands.
- Minar, N, Burkhart, R, Langton, C and Askenazi, M: 1996, *The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations*, Santa Fe Institute Working Paper 96-04-2.
- Perrone, A. SwarmJournal, available <http://www.alessandroperrone.it> in Swarm section
- Pinson, W: 1991 *Objective-C, Object-Oriented Programming Techniques*, Addison Wesley Publishing company
- Terna, P: 1998, Simulation tools for social scientists: Building agent based models with Swarm, *Journal of Artificial Societies and Social Simulation* 1(2) available at <http://www.soc.surrey.ac.uk/JASSS/1/2/4.html>
- Tesfatsion, L: 2001, Introduction to the special issue on agent-based computational economics, *Journal of Economic Dynamic and Control* 25(3-4): 281-293.