Agents ruling and ruled, rules and meta-rules: does the world belong to anarchists?

Pietro Terna[1], University of Torino
terna@econ.unito.it

Abstract

We propose a framework within which to investigate three topics: first, how enterprises and organizations arise, behave, and fall; second, how they interact; and finally, how we can improve them. The tool that we introduce here to help us in this research effort is a large agent-based simulation framework which is able to reproduce the enterprise context in a detailed way.

The basis is an agent based computational experiment on rules and rule emergence - named AESOP (Agents and Emergencies for Simulating Organizations in Python) – to use actions, agents (acting and deciding people) and the scheduling of events into an agent based framework. Agents may use fixed rules but they can also learn to improve these rules in a changing environment. They can also be modeled so as to be aware of the consequences of their behavior. In the simulator, single agents can also be modeled as neural networks, so that the system appears as a "net of neural networks". Alternatively, it is possible to introduce Bayesian learning processes in our models.

The core of the simulator is the capability to reproduce, in detail, the decision making process of organizations, firms and individual agents. The basis of the method is the reconstruction of the investigated phenomenon via the action and interaction of minded or no-minded agents. We can integrate both no-minded agents - agents capable of just performing tasks they are ordered to carry and minded ones - agents capable of taking decisions within the model.

[1] Acknowledgment for financial support of the basic framework of this research to the "tocai.it - Tecnologie Orientate alla Conoscienza per Aggregazioni di Imprese in Internet" project, http://www.dis.uniroma1.it/~tocai/index.php, whithin the FIRB (Fondo per gli Investimenti della Ricerca di Base) fund of the Italian Ministero dell'Università e della Ricerca.

Rules are not static here and can be modified by special agent, via meta-rules, creating highly complex environments, useful to investigate both actual and abstract economic context and their implication for economics. (A first sketch of the simulation structure is experimentally on line at http://eco83.econ.unito.it/terna/storep).

The simulator employs two independent components to build a description and representation of the world. Our simulated organizations or firms have both production units that perform the different steps of the production process and orders to accomplish the production. The orders are described by recipes that contain the "What to Do" (WD) component of the process; the production units represent the "which is Doing What" (DW) component of the same process. A third formalism relates to the time sequence of the events (the orders to be executed) that occur in the environment we are simulating; this is the "When Doing What" (WDW) component.

The simulator is currently based on Swarm (www.swarm.org) as basic layer, but a new structure, the AESOP simulator, is now under implementation as a Python (www.python.org), using SLAPP (Swarm-Like Agent Protocol in Python, http://eco83.econ.unito.it/terna/slapp/), to join the easiness of Python, and its openness, to the clarity of the Swarm protocol. Python is connected here to the R statistical system (R is at http://cran.r-project.org/), via the rpy library, at http://rpy.sourceforge.net/).

1 Why this proposal

In Clarkson and Simon (1960) we have a well known warning on classic economic theories analyzing decision-making within firms:

> (p.924) With great ingenuity, we axiomatize utility, as a cardinal quantity; we represent risks by probability distributions. Having done this, we are left with the question of whether we have constructed a theory of how economic man makes his

> decision, or instead a theory of how he *would* make his decision if he could reason only in terms of numbers and had no qualitative or verbal concepts.

From the point of view of the decision-making process in organizations (Simon, 1979) the situation becomes increasingly complicated, as in this realm prices do not operate at all and classical economics have very poor explanatory capabilities. Other fields of science also fail to offer a strong framework to allow us to understand, explain, and modify organization activities. The problem is strictly linked with understanding how human beings make choices. As Simon (1979) notes in his introduction:

> Administrative Behavior has served me as a useful and reliable port of embarkation for voyages of discovery into human decision making; the relation of organization structure to decision making, the formalized decision making of operation research and management science, and in more recent years, the thinking and problem solving activities of individual human beings.

Starting from administrative decision-making, Simon introduced the key idea of bounded rationality in human behavior. He then extended this insight to political science, economics, organization theory, psychology, and artificial intelligence. Following Simon, we can point out that organizations make it possible to formulate decisions because they reduce the set of possible choices to be considered. In other words, they introduce an additional set of bounds on possible choices.

For all these reasons we propose a framework within which to investigate three topics: first, how enterprises and organizations arise, behave, and fall; second, how they interact; and finally, how we can improve them. The tool that we introduce here to help us in this research effort is a large agent-based simulation framework which is able to reproduce the enterprise context in a detailed way.

2 Simulation techniques and agent-based models

If we understand that organizations must act in the context of those decision-making processes by which sets of possible choices are built, we can improve them by remembering that the effects that arise from decision-making in actual organizations are non-linear. Because of this non-linearity, consequences frequently seem explainable only in terms of complexity.

Partly because of this non-linearity and complexity and also partly because of the non-quantitative and non-rational basis of a large part of decision making in organizations, we can hardly use traditional equation-based models to investigate organization behavior, including enterprise behavior.

As noted in Burton (2001), simulation requires that we specify the world we want to investigate. It can be complex or simple, and it can begin simply and evolve into complexity. Either way, we must specify its "black boxes" and metaphorically open them; we cannot just assume they exist. Thus, in simulation, we make behavioral specifications, not behavioral assumptions. The central issue is that we know more about our simulated world than about "real" world. With this necessary specification, the simulated world is a laboratory where we know important parameters because we specify them.

The rich world of simulation is versatile; we can perform many different kinds of studies. We can test hypotheses, explore new ideas, create large datasets, help solve problems, and go outside the boundaries of the "real" world. The simulated world can be used to understand the limits of our "real" world, extending the limits of the possible. It also can give a picture of what is likely or what might be. With the simulator, we can reproduce the behavior of a firm or of an organization in a detailed way if we build the simulation model employing agent-based techniques.

2.1 A specific tool for simulating enterprises and organizations

To run enterprise simulations, we introduce here the jES (Java Enterprise Simulator) that you can find on line at http://web.econ.unito.it/terna/jes. With our tool, we describe, in a detailed way, a two-sided world. We consider *both* the actions to be done, in terms of which orders are to be accomplished (the "What to Do" side, WD), and the structures able to do them, in terms of production units (the "which is Doing What" side, DW). Thus, our simulation model is, first and foremost, a description of the enterprise as it is. Just like the various flight simulator programs put at our fingers the control of the simulated airplane and then execute our commands, the simulator executes exactly what we ask take place into the simulated enterprise with respect to the two components described above. The plane can land gracefully or crash depending on our commands; likewise, the enterprise produces or stays clogged if our WD and DW choices are inconsistent.

We introduce here the basic ideas and the principles upon which the simulator is built in order to clarify the goals of the project; the main reference here is the detailed online "How to Use" of the program,

2.2 The technique

The simulator employs two independent components to build a description and representation of an enterprise world. Our simulated enterprise or organization has both production units that perform the different steps of the production process and orders to accomplish the production. The orders are described by recipes that contain the "What to Do" (WD) component of the process; the production units represent the "Which is Doing What" (DW) component of the same process. A third formalism relates to the time sequence of the events (the orders to be executed) that occur in the environment we are simulating; this is the "When Doing What" (WDW) component. Production units can be within the enterprise-organization or outside of it. If they are outside the simulated entity, they may constitute other enterprises or organization or they may stand alone as small unit actors.

The term "recipe" is typical of industrial economics. A recipe to cook something contains data about the quantities of certain actions, their timing, and whether the actions are parallel or sequential; our recipes here contain similar data. At this point, it is useful to introduce a dictionary of our terms: (i) a *production unit* is a productive structure within or outside our enterprise; a production unit is able to perform one or more of the steps required to accomplish an order; (ii) an *order* is the object representing a good or service to be produced; an order contains technical information (the recipe describing the production steps) and accounting data; (iii) a *recipe* is a sequence of steps to be executed to produce a good or a service.

The core of the model is the clear separation between the orders and the production units. WD and DW are completely independent, both in formalism and in code. Therefore, while running the model, we check the consistency of the two components as we would in the actual world, since the output of an enterprise arises from a complex interaction among products and production tools. As we will see below, recipes can also describe internal parallel production paths, computational steps, batch activities, and assembly phases. As such, these comprise ways in which the typical procurement problems of a supply chain can be reproduced and tested.

Orders enter the simulation from two sources: from an *order generator*, in which case they are created randomly as outcomes of a predetermined scheme, or from an *order distiller*, in which case they are extracted from an archive of preexisting orders and normally represent a given reality to be reproduced via the simulator.

2.3 A simplified view

Fig, 1 provides a snapshot of the technique used here. This is an introductory view, with the recipes written in a simplified way, i.e., as a sequence of steps to be executed without information about the time required by each step. Observing the recipe 8-28-27-7, we can see that the front end (FE) of an enterprise can take charge in the first step, which will be executed by unit 8 within the enterprise (in this simplified version, production unit and step numbers coincide).

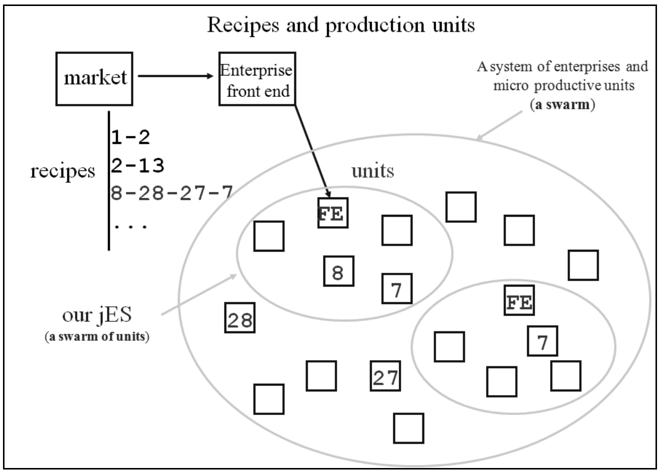

Fig. 1. A simplified view of the simulator components; recipes are reported here in a simplified way, without time specifications.

Fig. 2 introduces a more dynamic interpretation of the world we are describing. We have here three simple phases (*a*, *b*, *c*) in which the order containing the recipe 8-28-27-7 goes from one production unit to another. In this sequence, all the necessary information is contained in the order. When the activity of a production unit (as an example, unit 8) is concluded, the production unit consults the order to find out the next step to be performed, and it then asks all the production units to reply if they are able to execute that task. In this way, the order makes its journey from unit 8 to unit 28 (which is outside the enterprise and can be assumed to be a simple business unit) and then to unit 27 (also outside the enterprise). In the next step, designated with an *X* in Fig. 2, we have a choice problem, as two production units are able to perform task 7. We will introduce a set of criteria that allow the simulator to deal properly with this kind of problem.

While it may seem abstract, it is worth noting that one of the two units able to perform step 7 belongs to another enterprise. Therefore, we can imagine having to open a dialog with the front end of the second enterprise. We also have to take into account the possibility of a direct link with the production unit within the other enterprise. The idea of linking together the subunits of more complex enterprises to create temporary production organizations brings us directly to the concept of virtual enterprise as an organizational tool.

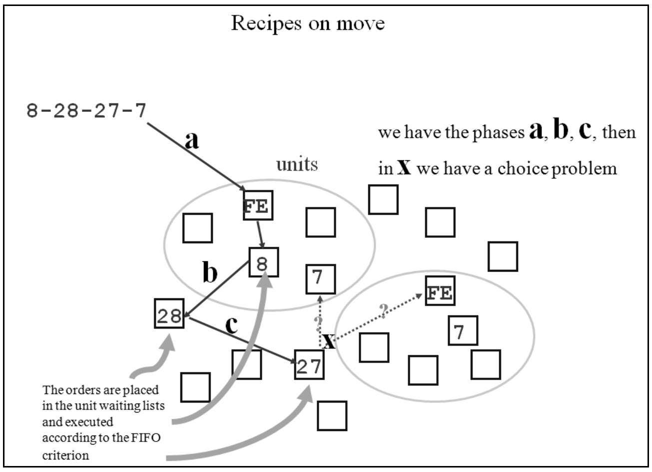

Fig. 2. A dynamic view of the simulator components; recipes are reported here in a simplified way, without time specifications.

3 Using the simulator

While the tool proposed here can be applied to actual enterprise cases, the goals of this kind of simulation are primarily theoretical. Through virtual enterprises or organizations built in this rich framework, we can investigate not only how those units originate and how they interact in social networks (Burt, 1992; Walker et al., 1997) of production units and structures, but also how they behave in *would be* situations.

The theoretical work that our tool makes possible can be understood in terms of two different approaches to "the quest of the enterprise", an important theoretical subject unsolved in economics. The first approach involves interpreting firms and organizations as systems of conveniences in which increasing returns for cooperative behavior emerge, but these returns are limited in size. Therefore, agents have relatively low incentives to expend large efforts in relatively large organizations, since their share of the aggregate results is only somewhat affected by their efforts. Moreover, as the number of so-called "free riding" agents increases, agents migrate to other firms. As explained by Axtell (1999), "successful firms are ones that can attract and retain productive workers." The second approach of our investigation presents the enterprise as a place where entrepreneurial ideas and choices operate, both in the Kirzner (1997) sense and in Burt (1992) analysis of structural holes.

Following Kirzner (1997) our models look for the trial and error process that generates both the creation of enterprises and organizations and their decline by focusing on the role of the entrepreneurs or of the managers in decision making.

4 Evolving the simulator

From a technical point of view, the simulator is currently based on Swarm (www.swarm.org) as basic layer. The Swarm protocol is a ”classical” reference in the relatively young world of the agent-based simulation, mainly for social sciences. The Swarm project, born at Santa Fe Institute, has been developed with an emphasis on three key points (Minar et al., 1996): (i) Swarm defines a structure for simulations, a framework within which models are built; (ii) the core commitment is to build a discrete-event simulation of multiple agents using an object-oriented representation; (iii) to these basic choices Swarm adds the concept of the "swarm," a collection of agents with a schedule of activity.

The Swarm protocol is now under implementation as a Python (www.python.org) layer upon (on line at http://eco83.econ.unito.it/terna/slapp) with our simulator built upon that layer to use the easiness of Python and its openness, i.e., connecting it to the R statistical system (R is at http://cran.r-project.org/; Python is connected to R via the rpy library, at http://rpy.sourceforge.net/).

* * *

About the contents, the main idea for the evolution of the simulator framework is now that of introducing rule modifications (WD side of the model) and agent adaption to changing rule (DW side of the model), both to search for easier ways to apply the new rules and to evolve counter-rules.

A sketch version of this new perspective is on line, with frequent modifications, at http://eco83.econ.unito.it/terna/storep.

Acknowledgments. This research is supported by the ICT4LAW project financed within the Converging Technologies framework of the Regione Piemonte.

References

AXTELL R. (1999), *The Emergence of Firms in a Population of Agents: Local Increasing Returns, Unstable Nash Equilibria, And Power Law Size Distributions*, Center on Social and Economic Dynamics, Working Paper No. 3, http://www.brook.edu/es/dynamics/papers/firms/firmspage.htm.

BURT R.S. (1992), *Structural Holes – The Social Structure of Competition*. Cambridge, MA, Harvard University Press.

BURTON R.M. (2001), *Afterword*, in A. Lomi and E.R. Larsen (eds), Dynamics of Organizations – Computational Modeling and Organization Theories. AAAI Press / The MIT Press, Menlo Park, CA, Cambridge, MA, London

CLACKSON G.P.E. AND SIMON H.A. (1960), *Simulation of Individual and Group Behavior*, American Economic Review, Vol. 50, No. 5, (1960) pp. 920-932.

KIRZNER I. (1997), *Entrepreneurial discovery and the competitive market process: an Austrian approach*, in Journal of Economic Literature. Vol. 35, no. 1, pp. 60-85.

MINAR N., BURKHAR R., LANGTON C. AND ASKENAZI M. (1996), *The Swarm simulation system: A toolkit for building multi-agent simulations*. Working Paper 96-06-042, Santa Fe Institute, http://www.swarm.org/images/b/bb/MinarEtAl96.pdf.

SIMON H.A. (1997), *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations*. THE FREE PRESS, Simon & Schuster, New York.

WALKER G., KOGUT B. AND SHAN W. (1997), *Social Capital, Structural Holes and the Formation of an Industry Network*, Organization Science. Vol. 8, No. 2, pp. 109-125.