Different Paradigms of Software Agents Applied to Modeling and Simulation of Complex Social Systems

Marco Remondino (remond@di.unito.it)
Department of Computer Science, 185 C.so Svizzera
Torino, 10149 - Italy

Abstract

The term agent, deriving from the Latin “agens”, identifies someone (or something) who acts; the same word can also be used to define a mean through which some action is made or caused. The term is used in many different fields and disciplines, such as economics, physics, natural sciences, sociology and many others. In computer science, the word is used to define very heterogeneous entities and sometimes is even abused. The main purpose of this work is to investigate various kinds of software agents that could be applied to modeling and simulation of complex social systems.

Introduction

The concept of software agent originates in the early fifties with J. McCarthy, while the term has been coined by O.G. Selfridge some years later, when both of them were working at the Massachusetts Institute of Technology. Their original project was to build a system which, given a goal, could be able to accomplish it, looking for human help in case of lack of necessary information. In practice, an agent was considered a software robot that lives and acts in a virtual world. In (Wooldridge and Jennings 1995): "... a hardware or (more usually) software-based computer system that enjoys the following properties:

- *autonomy*: they operate without the intervention of humans or others, and can control their actions;
- *social ability*: agents interact with other agents (and possibly humans) via some kind of language;
- *reactivity*: agents perceive their environment and respond in a timely fashion to changes that occur in it;
- *pro-activeness*: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.

Franklin and Graesser (1997) also try to find the typical features of agency, deriving them from the word itself: an “agent” is 1) one who acts, or who can act, and 2) one who acts in place of another with his permission. Since "one who acts in place of " acts, the second usage requires the first. Humans act, as do most other animals. Also, some autonomous mobile robots act, for example Brooks' Herbert (Brooks 1990; Franklin 1995). All of these are real world agents. Software agents "live" in computer operating systems, databases, networks, MUDs, etc.

Finally, artificial life agents "live" in artificial environments on a computer screen or in its memory (Langton 1989, Franklin 1995). Each is situated in, and is a part on some environment. Each senses its environment and act autonomously upon it. No other entity is required to feed it input, or to interpret and use its output. Each acts in pursuit of it's own agenda, whether satisfying evolved drives as in humans and animals, or pursuing goals designed in by some other agent, as in software agents. (Artificial life agents may be of either variety.) Each acts so that its current actions may effect its later sensing, that is its actions effect its environment. Finally, each acts continually over some period of time. A software agent, once invoked, typically runs until it decides not to. An artificial life agent often runs until it's eaten or otherwise dies. Of course, some human can pull the plug, but not always. Mobile agents on the Internet may be beyond calling back by the user.

These requirements constitute for sure the essence of being an agent, hence the definition by Franklin and Graesser (1997): “*An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future”.*

And the very general, yet comprehensive one by Jennings (1996): “*…the term is usually applied to describe self-contained programs which can control their own actions based on their perceptions of their operating environment”.*

A Simple Taxonomy

Agents themselves have traditionally been categorized into one of the following types (Woolridge and Jennings, 1995):

- *Reactive*
- *Deliberative*
- *Hybrid*

When designing any agent based (AB) system, it’s important to determine how sophisticated the agents' reasoning will be. Reactive agents simply retrieve pre-set behaviors similar to reflexes without maintaining any internal state. On the other hand, deliberative agents behave more like they are thinking, by searching through a space of behaviors, maintaining internal state, and predicting the effects of actions. Although the line between reactive and deliberative agents can be somewhat blurry, an agent with no internal state is certainly reactive, and one which bases its actions on the predicted actions of other agents is deliberative. In Mataric (1995) we read that reactive agents maintain no internal model of how to predict future states of the world. They choose actions by using the current world state as an index into a table of actions, where the indexing function's purpose is to map known situations to appropriate actions. These types of agents are sufficient for limited environments where every possible situation can be mapped to an action or set of actions. The purely reactive agent's major drawback is its lack of adaptability. This type of agent cannot generate an appropriate plan if the current world state was not considered a priori.

Different from reactive agents are the deliberative ones. The key component of a deliberative agent is a central reasoning system (Ginsberg, 1989) that constitutes the intelligence of the agent. Deliberative agents generate plans to accomplish their goals. A world model may be used in a deliberative agent, increasing the agent's ability to generate a plan that is successful in achieving its goals even in unforeseen situations. This ability to adapt is desirable in a dynamic environment. The main problem with a purely deliberative agent when dealing with real-time systems is reaction time. For simple, well known situations, reasoning may not be required at all. In some real-time domains, such as robotic soccer, minimizing the latency between changes in world state and reactions is important.

Hybrid agents, when designed correctly, use both approaches to get the best properties of each (Bensaid and Mathieu, 1997). Specifically, hybrid agents aim to have the quick response time of reactive agents for well known situations, yet also have the ability to generate new plans for unforeseen situations.

Multi Agent Systems (MAS)

A multi agent system can be thought of as a group of interacting agents working together to achieve a set of goals. To maximize the efficiency of the system, each agent must be able to reason about other agents' actions in addition to its own. A dynamic and unpredictable environment creates a need for an agent to employ flexible strategies. The more flexible the strategies however, the more difficult it becomes to predict what the other agents are going to do. For this reason, coordination mechanisms have been developed to help the agents interact when performing complex actions requiring teamwork. These mechanisms must ensure that the plans of individual agents do not conflict, while guiding the agents in pursuit of the goals of the system.

Analytical and Simulation Modeling

Modeling is a way of solving problems that occur in the real world. It is applied when prototyping or experimenting with the real system is expensive or impossible. Modeling allows to optimize systems prior to implementation. It includes the process of mapping the problem from the real world to its model in the world of models, – the process of abstraction, – model analysis and optimization, and mapping the solution back to the real system. We can distinguish between analytical and simulation models. In analytical, or static, model the result functionally depends on the input.

However, analytical solution does not always exist, or may be very hard to find. Then simulation, or dynamic, modeling may be applied. A simulation model may be considered as a set of rules (e.g. equations, flowcharts, state machines, cellular automata) that define how the system being modeled will change in the future, given its present state. Simulation is the process of model “execution” that takes the model through (discrete or continuous) state changes over time. In general, for complex problems where time dynamics is important, simulation modeling is a better answer. In Figure 1, *metaphor based approach* (Remondino, 2003) is shown, depicting how to step from a real observed situation (problem in the real world) to a computer model and hence to a simulation in order to obtain results that can be scaled back to be applied to the original problem.

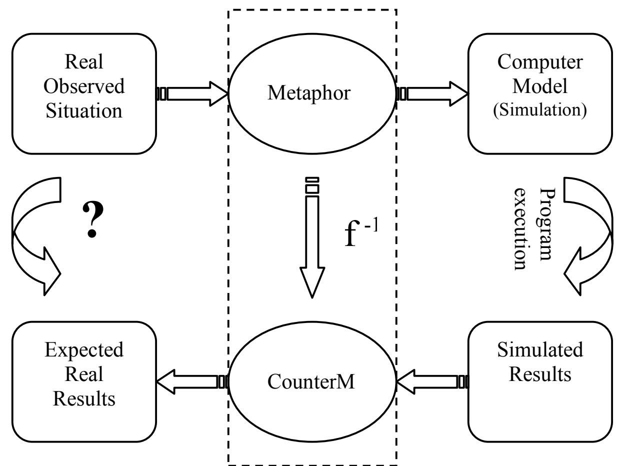

Figure 1: from the Real Problem to the Model

The metaphor layer is a conversion one, and works like a function, which maps a real situation onto a computer program, that can be executed by a machine. The results obtained by the simulation built with this approach, don’t necessarily apply one-to-one to the real situation. Therefore, an inverse function is required, which makes them suitable for the observed reality; this inverse function, called *counter-metaphor*, allows applying the results obtained from the simulation to the real world analyzed problem.

According to (Troitzsch, 1996), computer simulation in the social sciences has at least two types of origins: on one hand, it continues mathematical modeling and is no more than the numerical treatment of difference equations or the various kinds of differential equations (including partial and stochastic differential equations). On the other hand, computer simulation is used in its own right, not as a substitution for more elegant mathematical solution algorithms, but as a means of manipulating the symbols of the symbol system of programming languages.

Ostrom (1988) described simulation as a third symbol system in its own right and as an alternative to mathematical formalization of social science theories and verbal argumentation. The former is highly computable, but it's very difficult to express real observed situations just by numerical means and equations. The other alternative is natural language, which has a huge capability in representation but it's not computable at all. Ostrom stated that "any theory that can be expressed in either of the first two symbol systems can also be expressed in the third symbol system" and that computer simulation has the advantages of both the other symbol systems, without their disadvantages, since it "can be used for representing both qualitative, natural language constructs and quantitative, mathematical constructs".

Thus, simulation can be used to detect which conclusions may be drawn from complex antecedents. This is what used to be called *concept-driven simulation* (Henize, 1984). A target system is represented by a verbal, mathematical, or computer model (with all the necessary simplifications). The question is about the possible futures of such a target system: will it stabilize overtime or be destabilized? What happens if we change something in the initial conditions? Can the system be optimized, regarding some core parameters? This is the core of the simulation process, sometimes referred to as *what if analysis*; a simulation can indeed give some very useful results about what we can expect from the target system, when this is carefully modeled. Of course simplifications are needed – a model, by definition, is a scaled down representation of reality – but even then the results can apply to real situations.

Agent Based Simulation

In an AB model there is not a place where the global system behavior (dynamics) would be defined. Instead, the modeler defines behavior at individual level, and the global behavior emerges as a result of many (tens, hundreds, thousands, millions) individuals, each following its own behavior rules, living together in some environment and communicating with each other and with the environment. That is why AB modeling is also called bottom-up modeling. Instead of creating a simple mathematical model, the underlying model is based on a system comprised of various interacting agents. Therefore, its structure and behavior have potential to resemble the actual economic theory and reality better than simple mathematical models, especially when the underlying real relationships are complex. In (Bonabeau, 2002), we read that AB paradigm can be used successfully to model different situations, like flows, markets, organizations, social diffusion of phenomena

Complex Social Systems

There are many accepted definition for the word *complexity*, when applied to a social system, i.e.: a system in which the single parts interact among them. The most straightforward one is in (Pavard and Dugdale, 2000): *A complex system isone for which it is difficult, if not impossible to restrict its description to a limited number of parameters without losing its essential global functional properties.*

Formally, a system starts to have complex behaviors the moment it consists of parts interacting in a non-linear fashion. According to this, a complex system is defined as *the interaction of many parts, giving rise to difficulties in linear analysis due to the nonlinearity of circular causation and feedback effects (Calresco Glossary).*

It is thus appropriate to differentiate between a complicated system (such as a plane or computer) and a complex system (such as ecological or economic systems). The former are composed of many functionally distinct parts but are in fact predictable, whereas the latter interact non-linearly with their environment and their components have properties of self-organization which make them non-predictable beyond a certain temporal window.

A truly complex system would be completely irreducible and it would be impossible to derive a model from it without losing all its relevant properties. However, in reality different levels of complexity exist. If we are interested in situations which are structured and governed by stable laws, then it is possible, without loosing too many of the system's properties, to represent the system by simplification.

Reactive Agents Applied to Game Theory

Game Theory (GT) is a distinct and interdisciplinary approach to the study of strategic behavior, founded by John von Neumann. The disciplines most involved in game theory are mathematics, economics and the other social and behavioral sciences. GT was intended to provide a theory of economic and strategic behavior when people interact directly, rather than through the market. In GT, "games" are a metaphor for serious interactions in human society.

Many problems belonging to the GT field focus on complex social systems; in fact we have the interaction among individuals, and the aggregate results can show a non linear and emergent behavior. For this reason, they can be modeled through agent based simulation and, in particular, reactive agents can be suited for that. Here follows an example of a model which faces a well known problem belonging to the GT field, the Minority Game (MG), simulated with the use of a community of reactive agents.

Minority Game with Communication

The Minority Game (MG) is a simple, generalized framework, belonging to the GT field, which represents the collective behavior of agents in an idealized situation where they have to compete through adaptation for some finite resource. While the MG is born as the mathematical formulation of "El Farol Bar" (EFB) problem considered by (Arthur, 1994), it goes way beyond this one, since it generalizes the study of how many individuals may reach a collective solution to a problem under adaptation of each one's expectations about the future. The original formulation of EFB problem is as follows: N people, at every step, take an individual decision among two possibilities. Number one is to stay at home; number two is to go to a bar. Since the space in the bar is limited (finite resource), the time there is enjoyable if and only if the number of the people there is less than a fixed threshold (aN, where a<1). Every agent has his own expectation on the number of people in the bar, and according to his forecast decides whether to go or not. The only information available to the agents is the number of people attending the bar in the recent past; this means that there is no deductively rational solution to this problem, but there can be plenty of models trying to infer the future number according to the past ones. The EFB problem has been applied to some proto-market models: at each time step agents can buy (go to the bar) or sell an asset and after each time step, the price of the asset is determined by a simple supply-demand rule. The MG has been first described in (Challet and Zhang, 1997) as a mathematical formalization and generalization of EFB problem. It is assumed that an odd number of players take a decision at each step of the simulation; the agents that take the minority decision win, while the others loose.

The EFB problem, as well as the MG in its original formulation state that there is no communication among the agents involved in the simulation; the original idea in this example is to introduce in the model a sort of a social network (Remondino and Cappellini, 2004), in order to see how the links among certain agents can influence them. A social network is defined as "*a set of nodes - e.g. persons, organizations - linked by a set of social relationship (edges) of a specific type*" (Laumann, et al., 1978). The following example is useful to show that even using simple, reactive agents, we can have realistic results from a simulation of a complex social phenomenon.

The Simulation Framework

During the setup, we create a simple world populated by *N* agents. These agents can be considered as the vertices of a social network and the links among them (relations) as the edges. Every agent has a list of *F* (friends) other agents to whom he can ask, linked to the examined vertex (the agent).

At the beginning of each simulation step, every agent has its own forecast. The forecast is absolutely random between two choices –1 and +1. The decision taken by each agent (before communicating with others) is denoted with a "certainty index" equal to 1 (100%). Now an agent is randomly chosen. He starts asking to the first in the list; if this one has the same prevision, then the certainty index is increased by a value of *1/F*, while if the prevision is different, than the certainty index is lowered by *1/F*. After having asked to all the friends in his list, the agent takes the final decision: if the certainty index is equal or greater than 1, then the decision will be the original one. If it's lower than 1, then the decision will be the other possible one. Another agent is then randomly chosen, and so on.

Simulated Results

In the output graph we can read the time on *x-axis* (1000 iterations of the game). On *y-axis* we read the number of decisions. The lower line represents changed decisions, while the upper one the unchanged ones. In the first run we have a world of 100 agents and 500 relations (Figure 2), in which 65 out 100 preserve their original decisions.

In a second run we imagine a different situation, in which the agents have many more relations among them: an average of fifty for every inhabitant (Figure 3). A simple common sense rule states that the more relations, the higher is the probability to change opinion.

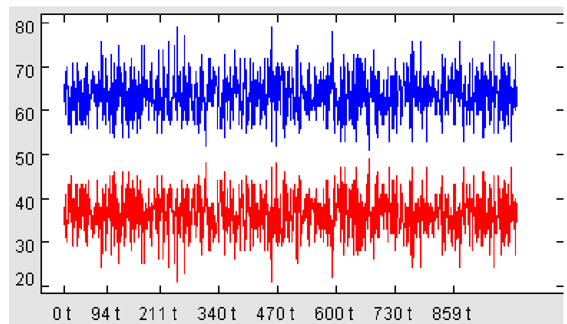

Figure 2: 100 Agents and 500 Links

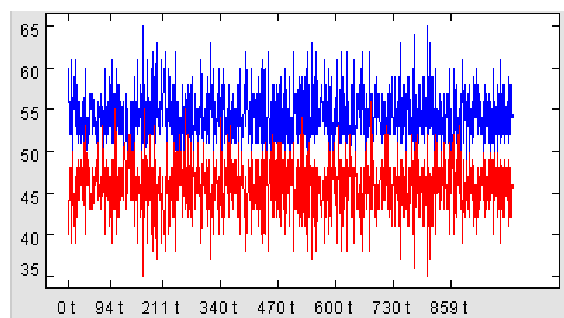

Figure 3: 100 Agents and 5000 Links

This example proves the rule to be right and our model to be consistent with real world results; we can now try a counter example, i.e. a poor relations world, as the one in Figure 4; one thousand inhabitants with a total of just five hundred relations.

This very simple example proves that reactive agents, i.e. software entities simply endowed with the capability of sensing the environment (in this case, using a social network to know what others will do) and with a simple function (the one used to take the decision) can present a realistic aggregate behavior. In this case, with some very simple rules a community of agents is simulated and a rule of thumb emerges: the more relations in a community of agents, the more the probability of changing the original opinion. This can apply, for instance, to marketing studies or even to political campaigns.

A very different use of software agents is done with deliberative agents, i.e. those endowed with the ability of reasoning about actions and consequences. In the next paragraph and example is shown of a theoretical model employing BDI agents, a subset of deliberative ones.

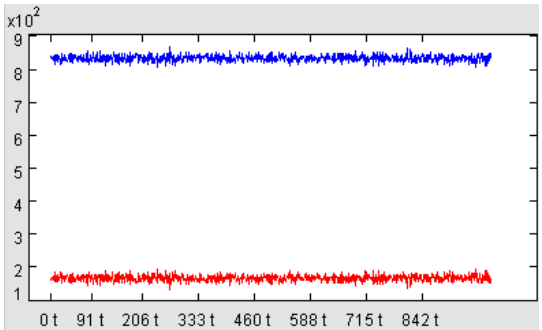

Figure 4: 1000 Agents and 500 Links

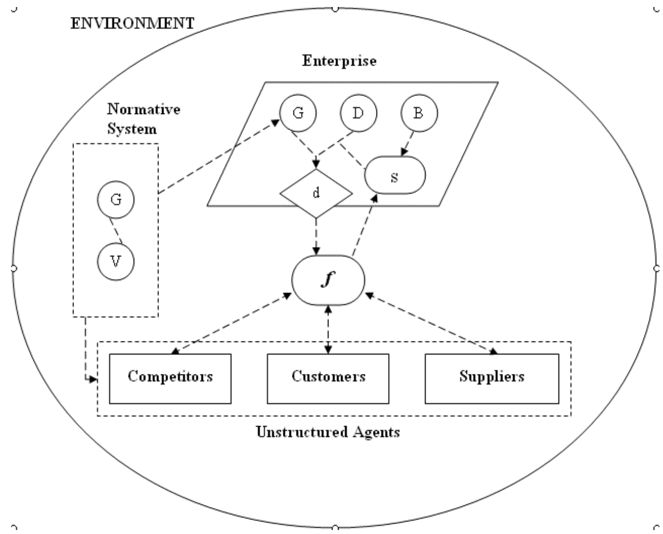

Figure 5: BDI Enterprise at a Macro Level

BDI Agents Applied to Enterprise Modeling

According to (Kinny et al., 1996), the BDI paradigm provides a strong notion of agency; agents are viewed as having certain mental attitudes, *Beliefs*, *Desires* and *Intentions*, which represent, respectively, their informational, motivational and deliberative states. In the BDI architecture an agent can be completely specified by the events that it can perceive, the actions it may perform, the beliefs it may hold, the goals it may adopt, and the plans that give rise to its intentions.

In the following, the enterprise is considered as a BDI meta-agent, that's an agent grouping other ones, representing the functional areas. Rather than a component based or object based approach here we model the different levels as agents which are "real" only as long as all the ordinary (human or software) agents believe they are.

Macro Level Description

The whole enterprise can be considered as a meta-agent, grouping a series of other agents, sharing some common goals, desires and beliefs. The environment in which the enterprise operates is shared with other subjects, mainly competitors, customers and supplier, which have a direct influence on the facts (f) and are somehow connected with the enterprise itself. These are also agents, but we shouldn't be interested in giving them a real BDI structure, since that's not directly visible from the enterprise we model; though, the enterprise can have a representation (through its own beliefs) of what is supposed to be the structure of the outer agents. For example, it will know for sure that the competitors will try to overcome it in terms of market share, while the customers need to be satisfied and will try to get the best possible goods for the least money. Other external agents can exist: for example a normative system, created "a priori". According to (Boella, 2003) this is defined as a social constructed entity, to which the other agents in the world attribute to it regulative norms.

Since this is not the main focus of the model, the normative agent is simplified from the one proposed in (Boella, 2003) and acts just as a regulator for the actions of the agents involved. This agent is useful to avoid improper actions, like treacherous competition, prices out of the logical range and so forth. Thus the normative system will just feature goals (rules), from which a set of what is considered a violation (V) is derived. By observing the facts and considering its beliefs, the enterprise has its own representation of the state of the world (s).

Enterprise Level Description

The enterprise agent is divided into several sub-agents, each of which has its own beliefs, desires and goals. These are strictly linked with the general ones, belonging to the whole enterprise (B,D, G), in the sense that the attributes of the single functional area contribute to defining the general ones, but these, in turn, influence the ones of the various areas. This construction is realistic, since the areas constituting the enterprise must share with it some knowledge and objectives in order to make it work, but at the same time there could be some attributes which are not shared by all the subjects or that could not be of any interest for some of them, or can't be pursued by the enterprise as a whole. Besides with this model we have an indirect link among all the areas through a super-agent (the enterprise) which doesn't affect directly the behavior of the sub-agents, but is at the same time modifier and modified. For example, the enterprise seen as a BDI agent will pursue the highest possible profit, the best efficiency, the highest customer satisfaction and so on; some of these goals – or part of each - can flow to all the sub-agents, while others should be shared just by some of them. It's thus very important to introduce two levels of views, one corresponding to the whole enterprise and one to the single areas.

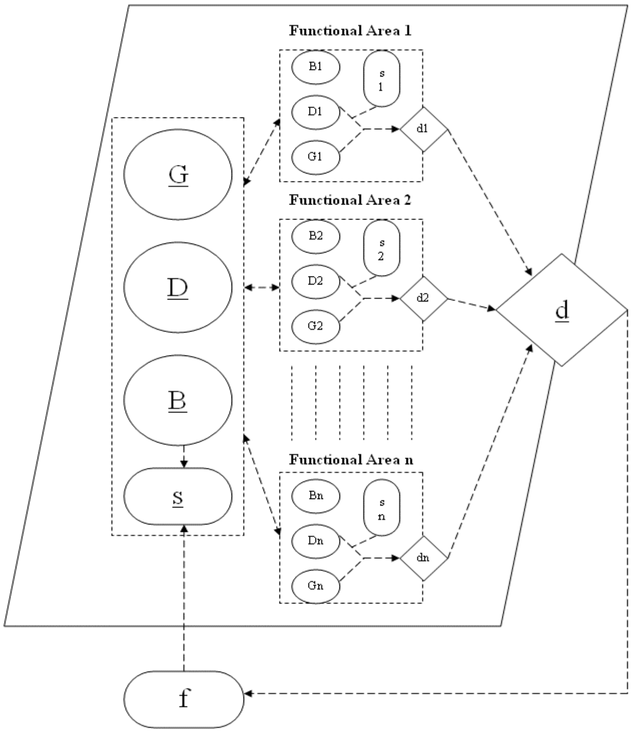

Figure 6: BDI Enterprise at a Micro Level

Each functional area can perform actions (d1, d2, …, dn) according to its own goals and desires. At a macro level, the set of these actions creates those (d) performed by the enterprise in the environment, that are the only ones which control some of the facts (f), while the ones performed by the functional areas don't directly affect them. It's crucial to determine how the various actions performed by every area affect the global actions of the enterprise. The mechanism, in the real world, is complex and certainly not deterministic.

The beliefs about the state of the world are also different if considered at the macro or micro level; the enterprise derives its own beliefs about the state of the world (s), from the facts it observes in the external environment (f) and from its own general beliefs (B). On the other hand, the beliefs about the state of the world belonging to the functional areas (s1, s2, …, sn) are derived from what they observe in their own environment, which is the enterprise itself, and/or from the external facts (f) filtered by the enterprise (i.e. by s) and always with the influence of their own beliefs (B1, B2, …, Bn). This is a rather closed and unrealistic vision: in the real world, the persons belonging to a functional area have their personal beliefs about the state of things from the outside, with no mediation by the enterprise. Though, since the focus is on the structure of the enterprise itself, it would be useless and time consuming to model different – and more complex - relations.

Conclusion

Two different agent based paradigms have been examined in this work; both reactive and deliberative agents can be used in modeling and simulating complex social systems, but with many differences among them. While the former paradigm is suitable for situations in which the aggregate behavior is more important than the individualistic one, the latter can be used to model problems where the complexity lies also in the single parts of the system. A crucial difference is also to be found in the computational complexity of the models built with those approaches; while in the first case, using an Object Oriented computer language is enough to build large communities of reactive agents, the second paradigm involves the use of Logic and thus is heavier and very difficult to be computed. One of the strong point of this approach is its capability of modeling complex situations in a clear way, reducing all the relations among the agents to logical links. So, while the reactive paradigm is a strong possibility to dynamically simulate complex social systems, the deliberative (particularly BDI) paradigm is useful to analytically and theoretically describe complex problems made up of heterogeneous agents.

References

Arthur B. (1994). Inductive Reasoning and Bounded Rationality, *American Economic Review (Papers and Proceedings)* pp.84,406-411

Boella G. (2003). Normative Systems as Agents, working paper

Brooks, R. A. (1990). Elephants Don't Play Chess, *Designing Autonomous Agents*, Pattie Maes, ed. Cambridge, MA: MIT Press

Bensaid N., & Mathieu P. (1997), A hybrid architecture for agents

Bonabeau E., (2002). Agent-based modeling: Methods and techniques for simulating human systems, *PNAS 99 Suppl. 3* (pp. 7280-7287)

Challet D. & Zhang Y.C. (1997). Emergence of cooperation and organization in an evolutionary game, *Physica A246*.

Franklin, S. (1995). Artificial Minds, Cambridge, MA: MIT Press

Franklin, S., & Graesser, A. (1997). Is it an Agent, or just a Program? *Proceedings of the ATAL Workshop, Berlin* (pp. 193-206). Springer.

Ginsberg M. (1989). Universal planning: An (almost) universally bad idea. *AI Magazine 10(4)* pp. 40-44

Henize, J. (1984), Critical issues in evaluating socio-economic models. *Simulation and Model-Based Methodologies.* vol. 10, (pp. 557-590).

Jennings, N. R. (1996). Software agents. *IEEE Review* (pp. 17-20)

Kinny D., et al. (1996). Modelling and Design of Multi-Agent Systems, *Intelligent Agents III: Proceedings of ATAL-96*

Langton, C. (1989). Artificial Life, Redwood City, CA: Addison-Wesley

Mataric, M. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems 16* (pp.321-331)

Ostrom T. (1988). Computer simulation: The third symbol system. *Journal of Experimental Social Psychology, 24* (pp. 381-392)

Pavard B. & Dugdale J. (2000). The contribution of complexity theory to the study of socio-technical systems, *Proceedings of 3rd International Conference on Complex Systems*

Remondino, M. (2003). Agent Based Process Simulation and Metaphors Based Approach for Enterprise and Social Modeling, *ABS 4 Proceedings* (pp.93-97). SCS Europ. Publish. House – ISBN 3-936-150-25-7

Remondino M. & Cappellini A. (2004). Minority Game with Communication: an Agent Based Model, *Simulation in Industry 2004,* SCS Europ. Publish. House – ISBN 1-56555-286-5

Troitzsch, K.G. (1996). Chaotic behaviour in social systems. *Modelle sozialer Dynamiken*. Ordnung, Chaos und Komplexitat, (pp. 162-186).

Woolridge, M., & Jennings, N. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review 10(2)* pp.115-152