

## **Chapter 3 Agent-Based Social Simulation**

### **Introduction**

The objective of this chapter is to identify the main theoretical and methodological perspectives of ABSS and to review recent work and key themes of discussion and debate in this field.

Sections 3.1 and 3.2 discuss the historical origins of ABSS, which puts this research field into its proper context by referring to distributed artificial intelligence, multi-agent systems, and the bottom-up approach to system design which has been gaining recognition in many different research areas. Section 3.3 describes the main objectives of modelling and contrasts different approaches, discussing the relative benefits and disadvantages of traditional mathematical modelling with newer, computer-based approaches to social simulation. Section 3.4 compares ABSS with discrete event simulation (DES), discusses the typical objectives of ABSS research, and discusses the type of research question to which this approach is well suited. In this section it is argued that ABSS is appropriate for the design of large, distributed and complex systems. From the argument outlined in this section, it should be clear that ABSS is an appropriate technique for modelling the types of issues discussed in chapter two.

Some philosophical arguments that are the subject of discussion by researchers in this field are reviewed. Then, section 3.5 explains how this thesis contributes to these discussions in terms of developing the methodology of ABSS. For example, the question of how different types of empirical data shape the model development in different ways. This is an important issue for this thesis, which seeks to incorporate qualitative with quantitative approaches. This section also discusses practical issues such as the grain of analysis of the model, the scale of the system and the complexity of agent architectures. A background on the SDML platform that was used to program the model is provided.

Section 3.6 describes how the design of simulation experiments should lead to a greater understanding of the behaviour of the model as well as to achieving the aim of

addressing the research propositions. Section 3.7 discusses the main techniques of validation that are available to modellers using the agent-based approach. There are different methods that take advantage of differences between qualitative and quantitative data, and which when used in combination can strengthen the confidence in the reliability of the results. The chapter concludes with a review of some models of value-chain issues, all of which involve networks of agents interacting in market systems and are representative of current research in this area. This allows a discussion of approaches to model development, validation methods, and potential applications: in other words, the issues that have been the themes of this chapter.

### **3.1 Distributed AI, Emergence and Complexity**

The roots of agent-based modelling are located within the field of Distributed Artificial Intelligence (DAI). The DAI approach is concerned with systems that consist of many interacting components which have some level of autonomy - that are both able to perceive their environment and also act so as to change that environment according to some goals. The distributed nature of components and thus the significance of local aspects of the system mean that the components often carry out very different tasks and have heterogeneous goals. In DAI, social interaction is the mechanism of coordinating these various components and their activities and achieving a useful outcome on the system level.

This stands in contrast to earlier research in AI, the so-called classical approach, which does not involve the social aspect and where intelligence is attributed to the cognitive abilities of the single unified system. Traditionally, performance is measured in highly simplified environments. All interaction is direct between the actor and the environment and therefore issues of coordination are not a primary concern. In DAI, the individual components are viewed in the context of their relationship to the other parts of the system, and 'intelligence' is best viewed in terms of the macro-level behaviour of the system. Functionality is therefore highly dependent upon how coordination of the separate components is achieved.

This brings us to discussion of how these relationships are organised. DAI is characterised by a 'bottom-up' approach to system design, in which lower level / component rules of interaction and behaviour are specified firstly, and then higher level or aggregate layers are built upon the lower ones. One of the earliest and most well known examples of the 'bottom up' design principle is that of Brooks and the subsumption architecture (Brooks 1985), applied to the design of control systems for autonomous robots, whereas in the area of social simulation, the focus of this chapter, some of the most commonly cited works are (Schelling 1978) and (Epstein and Axtell 1996).

In the ‘bottom-up’ approach there is no central control or blackboard system, rather control of the system is intended to emerge from the specification of interaction processes amongst the component parts. The designer of this type of system would exploit this property of complex systems to establish control and coordination rather than to program it directly. As a result of the influence of these interaction mechanisms, individual components will tend to behave in a regulated way. This will result in the system exhibiting structured behaviour on an aggregate level.

The reliance upon the phenomena of emergence – where macro behaviours are generated by micro rules of interaction – places the approach, and the techniques used to study it, within the domain of complexity research (Waldrop 1993), (Bar-Yam (in press)). The systems studied within DAI are typically very complex in the sense that the causal relationships - and the (often simple) behaviours that underlie them - cannot be identified merely by inspecting the macro behaviour of the system. In common with complex systems, DAI systems are designed with dense interaction and heterogeneity amongst the component parts, which sometimes generate patterned, structured, i.e., emergent behaviour that an observer can recognise. In this description, it should be pointed out that there are strong parallels with the way that many social systems are organised. For example, emergence of behavioural norms in human societies (such as fashion trends), population regulation in the food web, group behaviour of animals (flocking and herding). The examples are very numerous.

### **3.2 Multi-Agent Systems**

There are two strands to multi-agent research that can clearly be identified. The first is associated with the work of Wooldridge and Jennings (1998) and others, which aims to control the scope of the interactions between agents in order to create systems that have predictable, designed behaviour. The second approach places no such restriction on the design of interaction mechanisms: typically the rules of interaction are very important and vary considerably amongst agents. The relationships which are brought about by these interaction mechanisms can become quite complex. Edmonds (1998) distinguishes the two strands of research as the ‘engineering approach’ and the ‘social

simulation approach'. As this thesis follows the ABSS approach, the discipline will be more completely described in the following section.

Whilst these are both 'bottom-up' approaches because they commence with the design of agents and their interaction rules, the latter is obviously more concerned with a discussion of emergent phenomena than the former. Indeed, the first approach reduces interaction between agents precisely in order to remove the possibility of emergent phenomena occurring. The justification for this is that designers of computer-based control systems for many environments do not want any unpredictability to occur that is not anticipated in the design phase. Edmonds argues that the requirements for reliable systems are that they should be *predictable* and *transparent*, so that the user can work out the consequences of his/her actions before using the system, and the designer can make predictable design decisions. In general, such systems must be able to operate predictably and efficiently in situations where environmental factors are circumscribed and are well known in advance, and system behaviour is bounded, i.e. they must be tidy systems (Moss 2001).

In contrast, practitioners of the latter approach see emergence as a central to their work. Edmonds describes the objectives of ABSS and how this relates to the design perspectives of the 'engineering approach':

*"The interest in such simulations is often precisely because the resultant behaviour is surprising, in other words, that it is not transparent. Frequently the results of such simulations are not even predictable. For this reason, results and methods of researchers of these two perspectives are sometimes mirror-images of each other (in the sense of being opposite) – social simulators are often aiming to create exactly the type of situation that software engineers are trying to prevent."* (Edmonds 1998)

In complex and dynamic environments it is often not possible to foresee the effect of perturbations upon system behaviour, i.e. there are unexpected consequences that are difficult for the engineer to plan for. One suggestion is that such environments would be better served by systems designed to allow high interaction amongst agents. The idea is that emergence will bring order to the system, rather than letting it run outside the planned-for scope of system behaviour.

So far in this chapter we have discussed the scientific foundation of MAS and described the main characteristics of a new approach to distributed system design, ABSS, which places an emphasis upon dense interaction and emergent behaviour. Of course the other side of ABSS research - from whence it gains its inspiration - is the endeavour to directly model economic and sociological issues with computer simulations. We shall see that there is considerable overlap between these alternate premises: to build MAS appropriate for a given set of engineering goals, and to develop computational models of sociological phenomena, and argue that research efforts in ABSS modelling informs and aids development in MAS.

### **3.3 Objectives of Modelling and Social Simulation**

The very general purpose of modelling is to understand or illustrate some aspect of the target system of interest. In this chapter we shall be considering examples drawn from the field of social science (although the methodology described is also often applied to other areas of systems science). Gilbert and Troitzsch define a model as follows:

*“A model is a simplification – smaller, less detailed, less complex, or all of these together – of some other structure or system.”* (Gilbert and Troitzsch 1999, pg. 2)

This definition conceptualises the model as a simplification of some system, which can be either an artificial system or a real system - that attempts to capture some aspect or properties of that system. The model should be less complex than the real system, and lead to some improved understanding of how the real system functions or might function. In scientific research, developing a model corresponds to a process of abstraction from the target system, a process which must be made as transparent as possible so that the nature of the mapping between the system and its model is clearly understood.

Over the years, a large number of different approaches to modelling social phenomena have been developed. Much research carried out in this area often involved mathematical formalisms, with relationships amongst dependent variables described by sets of non-linear mathematical equations (often these are differential equations). This is the *mathematical modelling* approach. The advantage of interpreting social phenomena formally is that social entities are represented very precisely, and will be in a form to which mathematical or logical techniques can more easily be applied. We shall return to this point later.

Difficulties associated with mathematical modelling are that firstly it requires a high level of ability in mathematics to manipulate the equations, secondly they are often found not to be possible to solve (the issue of mathematical intractability). Thirdly, as a result of intractability, limiting assumptions or simplifications in the expression of relationships may be required in order to keep the problem amenable to the available techniques. For example, the mathematical treatment often necessitates that those quantities must be considered to be uniform and homogenous, which means that the method is really only suitable for the class of problems where entities can be represented as such. The strength of this technique is that given certain conditions, general solutions can often be found that solve, i.e. find the equilibrium states of, the model. The disadvantage is that the approach is limited in the sorts of problems to which it can be applied.

Many of the contemporary approaches to formal modelling involve the use of computer simulations, since computers allow fast numerical solutions to be generated for equations for models that contain many parameters. In general, they are appropriate to different classes of problems, or they provide an alternative to problems traditionally tackled with mathematics.

The text often quoted as a good introduction to the field of social simulation modelling is that of Gilbert & Troitzsch (1999). In the first chapters the authors present an overview of different modelling paradigms; in later chapters the focus is on how best to employ an agent-based methodology to research questions in social science. Four advantages of computer simulation over mathematical modelling for social science research are described:

1. Computer program code is more expressive than systems of equations.
2. Programs are better suited to modelling parallel processes.
3. Programs are (or can be) designed modularly and hence are easily modified.
4. Programs have flexibility to model heterogeneity amongst social actors.

The authors also argue that computer-based approaches to social simulation can help researchers achieve a variety of different objectives (Gilbert and Troitzsch 1999, pgs. 4-6) (see also Edmonds (2001)). Researchers might use social simulation tools in order to:

- Obtain a better understanding of social processes, especially dynamic ones.
- Make predictions about the occurrence of certain social events (e.g. demographic predictions or business forecasting).
- Simulate human abilities by modelling knowledge with expert systems.
- Aid theory development by formalising theories and testing via simulation.

In the next section we describe the ABSS approach to social simulation and contrast it with the other main computer-based approach used nowadays, discrete event simulation. We shall consider when it is appropriate to use a particular technique, and outline the strengths and weaknesses of each one.

### **3.4 Agent-Based Social Simulation Modelling (ABSS)**

Agent-based modelling aims to employ agents as the core component units that compose the model. In this sense agents can be thought of as intelligent, autonomous programs that interact with other components of the system and their environment in order to affect a certain set of programmed goals. More sophisticated kinds of agents have the facility to construct their own internal representations of the agent-world, to form expectations about events, and to exhibit limited kinds of learning behaviour.

Agent-based modelling is clearly distinguished from other kinds of modelling research by this focus on the concept of agents. It is a relatively new and immature research field, and as such there is a lack of established theory and research methodology underpinning the design of models, of standards for programming platforms, verification and validation of models, techniques for comparison of models and for establishing the generality of models. However, agent-based modelling is now rapidly gaining attention in many different areas due to its interdisciplinary appeal. This appeal can be attributed to the flexibility in the construction of agent architectures, and the fact that some quite complex and interesting model behaviour can be generated with this framework once the basic concepts have been understood.

*“One of the themes of social simulation research is that even when agents are programmed with very simple rules, the behaviour of the agents considered together can turn out to be extremely complex.”* (Gilbert and Troitzsch 1999, pg. 9)

Historically, the complexity of social systems has been one of the greatest challenges for different modelling approaches: the representation of social actors and their relationships is very problematic. Difficulties arise because social actors are heterogeneous, operate in very large systems, are not rational, are subject to social influence, etc. This complexity problem has greatly limited the success of efforts at modelling systems of the sort typically found in the real world. As a result, researchers have tended to put aside these types of problems and focus on obtaining more theoretical results from very simplistic systems, whilst those requiring models of complex systems have turned to statistical models that are reasonably good at prediction of outcomes, but do not capture processes.

In the field of agent-based simulation therefore there is much use of ‘toy models’: simple systems which bear little resemblance to reality because they are based on assumptions derived from more general theory. This type of model may therefore be classified as exploratory and focussed on theory building. However, the situation can become misleading where people have built toy models and tried to peg them to observed systems, and have done so by distorting the abstraction process. One has to be very careful not to infer from toy models things about the social world without good reason to do so, i.e. the simplifications must not undermine any arguments that

are being made about their level of realism. For example, such constructions as the assumption of the availability of perfect information, the rationality of agents, and of centralised control mechanisms, have long been criticised for being unrealistic ones (Moss and Sent 1999) because they are not supported by empirical facts.

This is not, in any way, to suggest that researching simplistic systems is unworthy of pursuit. In fact, since complex behavioural patterns can often be detected as the emergent consequences of simple system rules, it can lead to the formulation of theories relating to the link between individual action and the emergence of group or system-level behaviour. It can also be useful to better understand the techniques used by modellers, such as agent decision-making algorithms, agent activation mechanisms, and the properties of their models, such as system scalability, for example. This point brings us on to a discussion of the distinction between foundational and representational ABSS.

Examining recent theory regarding modelling social phenomena, Moss (2001) defines foundational ABSS as research concerned with formulation and verification of social theory and design of agent architectures, and representational ABSS as the use of MAS to describe observed social systems, arguing that there are very few examples of the latter that have been shown to meet these requirements for empirical validation, and few overall that are being used to inform MAS applications. However, the boundary between the two strands of research is not always so clear-cut.

Whereas agent-based modelling utilises agents as the central concept around which the program is constructed, *discreet event simulation* (DES) is characterised by a parallel concern with events. The simulation proceeds by carrying out instructions defined by a series of events, which are processed according to an ordered queue. When the first event in the queue has been performed, all subsequent events move up one, and any subsequent events are added, as they are generated, to the end of the queue.

One drawback of both ABSS and DES is that these systems typically cannot be solved in a formal sense: although it may be theoretically possible to explore all possible simulation trajectories, in practice computational limitations mean that only a subset

of outcomes may be simulated, and any conclusions about model behaviour are inferred from the sample obtained (which should be an appropriately large one). Analysis of results, therefore, involves statistical techniques and conclusions are always made within certain degrees of certainty.

The main benefit has already been mentioned above: the ability to tackle problems with systems involving high levels of heterogeneity and interaction amongst autonomous components, which result in quite complex behaviour. These are the types of systems that are difficult to understand analytically. A second benefit of ABSS is the relative ease with which an interacting system can be designed and programmed in agents. The choice of using agents as a core part of model specification extends upon the role that objects play in many different computer applications of the day. Object oriented programming has become prevalent in many areas of computer science because its properties of encapsulation (data and the associated programming rules that manipulate that data are treated as a unit), access modification (public/private levels of access to data inside classes), inheritance, etc. and because these features lead to fast development, easy modular development through the use of class libraries etc. The same can be said of agent-based programming.

This argument also suggests that there are many potential application areas of ABSS in systems engineering, i.e. developing MAS applications. Characteristic of the ABSS approach is that it leads to the development of systems that are flexible and scalable and robust to failures of individual component parts. This is because these properties are also the properties of the social systems which are being modelled. Therefore, in the process of development of models for social simulation, some techniques will necessarily be developed that can be applied to the problems of control and coordination of large complex systems.

For the reasons highlighted in this section, we argue that it is more appropriate to use the ABSS approach in situations where we wish to study complex phenomena, involving interactive systems of heterogeneous entities. The justification for choosing this approach over DES is that in the latter, an agent interacts only with its environment but not directly other agents. ABSS is therefore appropriate for exploring

social processes and developing explanations based on behavioural aspects of the system components (i.e. making the link between micro and macro).

Social simulation has limited usefulness for prediction (because of the inherent unpredictable nature of complex systems) but can be used to anticipate the types of events and the likelihood that they may occur. We shall call this type of prediction, which predicts not *when* there might be changes in behaviour, but *what* are the characteristics of that behaviour (and *why* does it occur), *vague* prediction or second-order prediction.

From the point of view of representational ABSS modelling, an obvious requirement is that simulation outcomes should be clearly related to aspects of real systems. In order to increase the likelihood of this happening, ABSS modellers have recognised the importance of having a problem-orientated approach to the design of models and the choice of techniques, and equally, the importance of validation procedures. These aspects will be discussed in greater detail in sections 3.5 – 3.7.

### **3.5 Modelling Methodology**

In the previous section we stated the objectives of ABSS modelling as being to improve scientific understanding and explanations of aspects of social systems composed of many interacting components and characterised by emergent social processes. Over the following sections we shall discuss the methodological approaches to this new scientific discipline.

According to Edmonds (2001), the methodological process can be broken down into several steps:

1. Abstraction of the target system to a conceptual model focussing on aspects of the target relevant to the study.
2. Formalisation of that abstraction according to some theoretical framework(s) and development of the computer model.

3. An inference step in which the model is executed and the behaviour is explored via simulation.
4. Interpretation and analysis of the results of the simulation and clarification of our understanding of the model.
5. Concluding with a discussion of what can be inferred about the original target system from the analysis of its model.

Some commonly used concepts that shape the model design are those of bounded rationality (Simon 1972) and social embeddedness (Granovetter 1985; Edmonds 1998).

The principle of social embeddedness is defined by Edmonds as follows:

*“An agent is socially embedded in a collection of other agents to the extent that it is more appropriate to model that agent as part of the total system of agents and their interactions as opposed to modelling it as a single agent that is interacting with an essentially unitary environment.”* (Edmonds 1998, pg. 2)

This implies that agent cognition and behaviour can be understood only in the context of interactions with other agents. In this framework, multi-agent systems that are developed will always incorporate some specifications of social interaction such as, for example, imitation, communication, persuasion, trading, bargaining, etc. as an integral part of the model design, and it implies that these processes will themselves be a central object of study. However, the form of social embeddedness should be chosen on the basis of what is known about the modelled system, i.e. it should be possible to validate the chosen model of socialisation.

Bounded rationality as originally defined (Simon 1972), expresses the idea that social agents have limited computational ability in terms of processing information to inform decisions as to how they should act, i.e., they are not rational in the sense of making optimal choices. Another limitation often considered is that agents have imperfect information about the environment in which they operate, often being only familiar with events that are locally significant. For agent research, this implies that systems should be designed that do not attribute unrealistic levels of ability to agents. Systems

that are designed in such a way will not only be based upon questionable assumptions but will also be handicapped in terms of system speed, scalability, and flexibility.

Model verification is carried out with respect to some theory or formalism (Moss 2001). Thus a model will be designed to be consistent with some a priori theoretical framework or interpretation. This framework constrains the model specification, so the modeller must ensure that it is an appropriate one for the research problem that is being addressed. Verification is made in terms of the formal language used to describe the model. For example, if the language in which the model is expressed is mathematics or if it is program code, then the model is verified according to that.

The modeller then has to decide upon the grain of analysis and level of formalism at which to specify the model. The objective is to construct a system that is significantly less complex than the target system. It should be simpler - in terms of being easier to understand the micro- and macro- behaviour and how these are linked – whilst as accurately as possible representing the relevant aspects of the target. Moss (1999) describes the abstraction process involved in the design of agent-based models as involving two distinct design choices: grain of analysis and degree of formalism.

The grain of analysis relates to the closeness of correspondence between the components of the model, and the components of the target. Fine grain models will specify one agent as a direct representation of one actor in the target system, whereas more coarse grain models will use one agent to represent a composite group, or perhaps a population, of social actors. Moss describes a compositional verification procedure as the development of suites of models in which “...*the behaviour of coarse grained models is required to be consistent with the behaviour of models at finer grains.*” (Moss 1999, pg. 20)

What is a particularly welcome consequence of this approach is that validation carried out in respect of fine grain models should, through the compositional verification procedure and its insistence of consistency amongst models, justify the confidence placed in the realism of more coarse-grain, generalised models. It is easier to validate at the fine grain level of analysis because components of systems are better understood than are aggregates of those components (i.e. the principle of reductionism

in science). For this reason, it seems logical that a compositional verification should be carried out starting with finer grain models and proceeding to increasingly coarse-grain models. This is in fact the opposite direction to that suggested by Moss. Good validation methods would give confidence that fine-grain models were representative of the target, and consistency of coarser-grain models with the emergent features of fine-grain models would allow development of more generalised findings and theories – although less well-validated, potentially more useful to understanding classes of such systems.

Central to all modelling research is the issue of power of expression and descriptiveness of models (typified by qualitative approaches such as ethnography) on the one hand, and clarity of expression and precision of formal methods (typified by quantitative approaches such as mathematical modelling) on the other. As explained by Moss (1999), due to very different backgrounds and training, researchers normally approach the problem from just one point of view, and there will be not much traffic between those aligned on opposite sides. Moss argues that agent-based modelling occupies a middle ground, and is a very powerful approach because it combines the rigour of formal logic with the descriptiveness of the agent paradigm for representing social actors and their interactions. Through this approach it will be possible to ‘shift out the trade-off’ (Moss 2000) between relevance and rigour in models of social processes.

What is meant by this is that agent-based modelling allows some of the advantages of both perspectives: models have a syntactic richness, i.e. they are informed by qualitative data and generate behavioural outcomes of a similarly descriptive nature, whilst they incorporate rigorous techniques involving formal specification of the model and how the components of the system are related.

This thesis contributes to this discussion because it is placed at this crossing point of qualitative and formal approaches. It brings to light some key methodological issues associated with this strand of representative ABSS, and demonstrates by means of a case study involving e-commerce how this approach can bring valuable insights.

There is a trade-off on a more practical level in the design of ABSS models, which is made between agent functionality and system scale. The functionality of an agent is the scope of the behaviour available to it, i.e. how complex are the mechanisms involved in perceiving external stimuli (through the environment and through other agents), in forming internal models representing aspects of the system, and upon various actions undertaken that might affect that system. The scale is measured in terms of the number of agents active in the model. Some models are composed of simple agents that are completely specified in a small number of behavioural rules, whereas some contain agents that have vast numbers of such rules and procedures and involve much information processing. Some agents execute very complex and adaptive decision-making mechanisms, involving, for example, genetic algorithms, neural networks, or re-enforcement learning, etc. The trade-off exists between the complexity of the agents, the scale of the system, and the computational resources required to simulate it. As computers become more powerful, the main barrier remaining will be that of system complexity, which is exponential in the number of agents composing the system (i.e. system scale) and the number of inputs to those agents (i.e. agent functionality).

### **3.6 Experimental Design, Implementation, and Analysis**

The model design and experimental scenarios should be well defined in the planning stages before model implementation begins. However, it is quite possible that further questions may arise during implementation that demand clarification of issues relating to the model design, or that analysis of simulation results may provoke the design of further experiments. On the other hand, we might simply be interested in some aspect of model behaviour that is poorly understood. This latter is what we call exploration with the model and is a distinct activity from experimentation because it does not aim to test some predefined hypothesis. Experiments can range in sophistication from the variation of one key parameter along one dimension up to a detailed exploration over many different model assumptions and inputs. Scenario analysis involves stipulating plausible alternatives of the model specification and drawing conclusions about the consequences by inspection of the results. This involves the comparison of different

agent interaction mechanisms, learning mechanisms, parametric values, environmental constraints, or exogenous inputs to the system in terms of their effects upon model outcomes.

The choice of programming language in which to implement the model will determine how quickly the simulation can be built, how easily it can subsequently be modified, how fast it runs, and how accessible are the results. The modeller may decide to use a procedural programming language or, alternatively, a declarative language, two approaches which involve very different frameworks for carrying out simulation research. Procedural programming involves the specification of processes that will take place if certain conditional states are met, whereas in the declarative paradigm, the program rules assert truth statements onto databases that are then operated upon by an inference engine. In the former, it is *states* of the system that hold interest for the researcher, whereas in the latter, it is the unfolding of *processes* that are of interest because the model is explicitly coded in states (Edmonds, Moss et al. 1996). Therefore certain types of modelling question will be more suited to one paradigm than to the other. Of course, the same model may be implemented in any programming language: the difference lies in the ease with which it can be implemented, and explored. In this section we present an overview of the Strictly Declarative Programming Language (SDML) (Moss, Gaylard et al. 1998) which is the declarative programming language used for the case study described in this thesis.

SDML is based on Smalltalk, which is a declarative, object oriented, programming language. SDML was specifically designed for developing agent-based models of social and economic scenarios, and has a large range of functionality that makes it highly suited to this task. SDML provides a modular development framework and a multiple inheritance structure for modules and for the types defined within the modules. There are standard **types** (in C/C++/JAVA terminology, these are equivalently known as classes or methods) that are instructive to understanding SDML and which will be described here:

- **Objects** are used to represent all information on databases.
- **Agents** are a special type of Object having rules and databases defined.

- **Looping Agents** have rules and states (databases) defined over one or more time levels.
- There is always one agent of type **Universal Agent**, containing all other agents, and being the starting point for the inference process.

SDML allows the modeller to create new sub-types inheriting from the above standard types. This means that when agents are created, they will inherit all the functionality of their own type, of all super-types to which they are linked, as well as from the types of agents in which they are contained.

The modelling language also allows flexibility in the defining of new clauses: any number of arguments (consisting of any combination of types, e.g. numbers, objects, or agents) may be added to the predicate. Every agent, in every time level, has associated with it both a **rulebase**, and a **database**. Rules consist of an antecedent part and a consequent part, whereas databases consist only of clauses. The inference engine is an algorithm which tests which rule antecedents are true and which are false by logical inference. When the conditions of a rule antecedent are satisfied (by retrieving information from the relevant databases if necessary), then we say that the rule has ‘fired’. The algorithm will then proceed to the next time cycle for that agent and assert the consequent clauses on the database. In this way, the program will step through all stages in the simulation, tracing one trajectory out of a possible many, and will arrive at the end state.

One advantage of the SDML language over its alternatives is that simulation databases are saved along with the results and remain accessible to subsequent enquiry. This means that at any time step, for any agent, the modeller can easily retrieve and keep track of any information that resides on the database. This feature has two advantages: one is that it makes the system more transparent to understanding what is going on, and the second is that it may save much time in that it makes program error detection easier, and that it might avoid the task of reprogramming the model in order to obtain new results. A second advantage is that some graphical capabilities are built into SDML: these provide a quick method of visualisation.

The implementation process will be the same regardless of programming paradigm or platform, and should be approached in a modular or step-by-step way. This involves writing a few rules at a time, usually limited to within just one or two rulebases, and then executing the model to test that the behaviour is as expected. In this way it is relatively simple to understand what effect individual rules or groups of rules have upon outcomes, because there are very few changes to the program.

If there are parts of the program code that can be usefully identified as self contained units, or that could be reused in another context, then they would best be placed in a separate module. In the early stages, small, test runs can be very useful if there are computational constraints making the simulation run slowly. In this case, the modeller would choose parameters that produce small but functional test simulations. One possibility is to define some default model parameters that provide test, full, and custom options, with small datasets constructed if necessary. This permits quick initialisation of simulations, which may save a lot of time in the long run.

Some consideration must be given to the definition of the outputs to be written to file: which variables are most appropriate to illustrate the model behaviour in the subsequent analysis of simulation results. They may need to be output in a format suitable and convenient for importing into other programs for analysis, if necessary. Paying careful attention to the outputs can make this process very straightforward and greatly reduce the amount of data preparation work involved later on.

Simulations are carried out by executing the model and exploring a number of model trajectories. It is not normally feasible to simulate every model trajectory because the possible outcomes are of a large order of magnitude. In other words, we cannot generate a proof of the properties of the model that could be shown to hold for all simulation runs. However, in order to understand the characteristic behaviour of the model, simulations are carried out over a large search space covering many possible parameter variations: the intention is to capture examples of each classification of outcomes. We argue that in order to understand all possible outcomes, we must execute many simulation runs, i.e. repeating the experiment a lot of times. However, the question of when to stop simulating, i.e. a sufficient number of runs have been

carried out such that the behaviour of the model has fundamentally been captured, is a decision made by the modeller and should always be justified.

In the ideal situation where computational constraints are not a big problem, as a guideline, the model may require in the order of several hundred or thousand simulations for each experimental set-up, to ensure reasonable certainty in the results. However, in the case where there are definite computational constraints, then the best approach is to continue running simulations until all subsequent runs describe qualitatively similar behaviour to that already identified. This obviously implies that some analysis will be carried out whilst still in the process of doing simulations.

Models often incorporate many random elements which stand in for unknown processes or processes which are known but are considered to lie outside the scope of the model. Control of these random elements is very important. A random seed is used to generate a string of pseudo random numbers or choices: each random seed will produce a unique string and therefore each simulation will follow a unique trajectory. There are two scenarios where careful control over random elements is vitally important. In the first case, where the modeller wishes to explore as many different simulation trajectories as possible with the model specification, different random seeds will be used to test the robustness of the model over many different parameters and initial conditions. To the contrary, in a second scenario where the modeller wants to explore the impact of tuning one particular parameter over a certain range, repeated use of the same random seed would enable a better comparison of results, since these would be less perturbed by random elements if the same string is used.

This brings us to the important issue of the robustness of results to parametric values and model assumptions. There should always be some attempt made at testing the robustness of the results via a sensitivity analysis, in which small changes are made in the model specification. Many agent-based models are characterised by self-regulating mechanisms, which generate fairly stable model behaviour provided that inputs are below a certain threshold, but may very suddenly switch into periods of very volatile or chaotic behaviour. This result is very familiar to researchers working in the field of complex systems research, where agent-based models are regarded as a

tool for developing systems theory. ABSS modellers are also interested in identifying critical thresholds and chaotic behaviour in their models because it helps them to understand the robustness or generality of their results.

In presenting the results, the modeller can choose between two approaches: averaging of results over many simulations or presenting results of typical simulation runs. There are advantages and problems with both approaches: the first technique has the advantage of being a more systematic approach and therefore the results would be considered more reliable, whereas in the second approach, the modeller faces the problem of how to select representative cases, and this may allow a certain bias to creep in. On the other hand, there is a danger of misinterpretation in drawing averages since aggregate results can distort the conclusions. In this respect the second technique, which allows the researcher to analyse specific cases, may lead to more insightful findings.

The model specification and simulation procedure should be well documented. One important reason is that documentation allows the model to be re-implemented by other researchers for checking the results, and for docking two or more very similar models (Axtell, Axelrod et al. 1996; Edmonds and Hales 2003), both of which facilitate collaboration and sharing of ideas.

### **3.7 Validation Methods**

As pointed out in the above section 3.4, validation by assessing the predictive accuracy of models is not usually attempted for the class of problems associated with ABSS modelling. Vague prediction is possible, however, allowing us to describe and explain behaviour that is characteristic of these types of system and to assess the likelihood of its occurrence. The validation procedure usually focuses on two methods: one qualitative and one quantitative approach. The qualitative part consists of comparison of qualitative behaviour of the model with empirical observations or descriptive accounts drawn from relevant literature and case studies. Quantitative validation is provided by comparing the statistical analyses across time-series outputs

from the model with empirical data. These two validation methods are complimentary as they target different aspects of the model: the micro and the macro levels.

Descriptive statistics are used to measure and characterise the macro behaviour of real world social systems. Normally this is the most easily accessible type of data that is available. The statistical data that are collected from observation of the empirical system are subsequently analysed for characteristic statistical patterns. In addition to first order statistics, we can also identify ‘statistical signatures’. These are well-documented patterns which are evident in many real-world social and artificial systems. Network analysts have devised statistical measures of the properties of networks of which some examples are: small world (Watts and Strogatz 1998), power law distributions, cliquishness, leptokurtosis, etc. and properties of the individuals within those networks such as range and prominence (Burt and Minor 1983) etc.

ABSS researchers have also identified these types of signatures in simulation outputs, and they argue that their models “*appear to produce data with empirically relevant properties because they capture features of social order that are the subject of sociological enquiry*” (Moss and Edmonds 2003, pg. 1), and that the existence of such statistical signatures should compose one of the main tests of the validity of models. This test would identify which agent architectures are inappropriate for the system (Moss, Edmonds et al. 2000). However, statistical data are not appropriate for understanding the micro behaviours of models because they do not describe the trajectories and causal relations of individual agents. This information is valuable in the untangling of processes and adds considerably to our understanding. To identify the micro behaviours involved we need to develop more penetrating techniques of enquiry.

Qualitative research is a highly semantic approach to understanding the social world, which involves the analysis of descriptive accounts and explanations relating to the beliefs, expectations, attitudes, etc. of different people, and requires synthesis of multiple points of view. A qualitative approach to modelling would involve using descriptive data to inform the model specification coupled with qualitative methods of validation. The specific techniques would be chosen on the basis of what data sources there are available, and how appropriate they are to the research objectives of the

modelling project. In this thesis it is proposed that a cross-disciplinary approach would best capture relevant aspects of micro behaviour. The reason that qualitative methods are an appropriate means of validation of the micro behaviour of agents is that they aim to uncover richness of detail of social organisation that quantitative methods do not. The objective of ABSS is to develop similarly detailed models that exhibit complex behaviour.

Moss & Edmonds (2003) discuss these issues in some depth, giving examples of qualitative validation techniques they have been using, and conclude:

*“When such micro level phenomena can be demonstrated to describe aspects of observed social behaviour and interaction and, at the same time, to generate the macro level phenomena sharing the statistical signature of real social data, then we shall say that the model has been cross-validated.”* (Moss and Edmonds 2003, pg. 3)

Taking the cross-validation approach means that there are several independent methods of validation of simulation models, and taken together they ought to lead to empirically well-validated model designs, and increasing confidence in the outputs of such models and the explanations they can provide of social phenomena. We shall return to discuss issues of validation in chapter five, where we present a detailed account of the validation methods employed in the current project.

### **3.8 Review of Models of Exchange**

In this section an overview is made of several market models that are related to this thesis in that they describe autonomous agents trading or sharing information, that represent identifiable economic actors such as companies, customers, intermediaries, etc. The models have different theoretical frameworks, aim to explore a variety of different research questions, and are more or less well validated by empirical data and. They are drawn from various sources: the Centre for Policy Modelling, the Journal of

Artificial Societies and Social Simulation, INGENTA searches, etc<sup>2</sup>. The following three models will be discussed:

- The simple supply chain model (Parunak and Vanderbok 1998) and the modelling framework for simulation of inventory control policies (Swaminathan, Smith et al. 1998).
- The information filtering economy (Moss, Edmonds et al. 2000)
- Model of the structure of textile production in the Prato industrial district (Fioretti 2001).

What these pieces of research have in common is that they all model supply chain structures and dynamics, and simulate the flows of information and products in a market. The next paragraphs explain in more detail the model aims, experiments, and results of the three examples and will summarise how they demonstrate the utility of agent-based models for addressing many kinds of questions about markets and for developing MAS in many areas of application.

Parunak and Vanderbok (1998) provide an illustration of how the agent-based modelling approach can be applied to coordination and control problems in industry by considering the supply chain model. Supply chains are modelled by a simplified network forming an hourglass shape, structured with a single manufacturer in the centre layer, connected via intermediary layers to a large number of consumers and suppliers at either end.

The model structure is based on data from two industrial firms. A very simple supply network is simulated incorporating four agents, one agent occupying each tier in the network. The agents interact in a fixed hierarchical structure. Unfortunately, no justification is given to explain why the model contains only four agents, which seems

---

<sup>2</sup> A methodological log record of searches made was not kept. However, a periodic search of various research databases, websites and conference proceedings was carried out including the following keyword terms: Electronic Commerce, Internet, EDI, supply chain, value chain, innovation diffusion, adoption, ICT. It is likely that work exists of which the author is unaware, since agent-based modelling publications are very scattered across different sources.

to be an unrealistic assumption in the context of real world supply chains. It is clear that a model containing so few agents would greatly constrain the complexity of behaviour that could be observed in relation to issues of coordination and competition. Also, there is no indication as to how the system will scale to sizes more appropriate to the study of real-world systems.

Consumer-agent (site 1) demands are drawn from a probability distribution with mean 100 units and variance 10 units, and company agents (sites 2, 3 and 4) are instantiated with three alternative rules for predicting demand. Other properties modelled are: order processing time, site capacity limitations, and goods delivery time. The authors describe several interesting aspects of model behaviour. The model exhibits amplification of variations in order size between sites 2 and 3, and between sites 3 and 4, increasing as we move further up the hierarchy away from the site of the consumer.

A weighted forecasting algorithm has the effect of correlation of variation in sites 2 and 3, and of persistence of large variation in sites 2 and 3 when customer orders are governed by a step function. The persistence, and magnitude of disturbances in order sizes implies that all three order management systems are not very good at adapting quickly to market changes such as changes in base levels of customer demand.

This paper was a groundbreaking work in the sense that it captured behavioural aspects of companies' response to supply chain dynamics by directly modelling them as agents. This is in contrast to most previous research analysing supply chains, in which the system dynamics approach is used to express the problem as a set of differential equations to be integrated over time. The problem with this latter approach is that complex interaction and multi-layered structure is very difficult to analyse. The main advantages of the agent-based approach are gained from the direct correspondence of agents with entities in the system being modelled, and from the very natural way that algorithms describing processes or state transitions can be implemented in computer models:

*“The behaviour of companies in their interactions with one another are driven primarily by non-Newtonian information dynamics that are most naturally represented algorithmically.”* (Parunak and Vanderbok 1998)

Swaminathan, Smith et al. (1998) have also worked in this area and produced a framework for modelling inventory control policies. They propose that their framework - which is based on discrete event simulation - can be used to develop customised simulation models of supply chain dynamics. Decision makers can use these simulation tools to perform risk benefit analyses of supply chain reengineering issues. They argue that one benefit of having such a framework is that it makes the development of models more efficient:

*“The modular architecture of our framework enables one to develop executable models for different situations with limited additional effort.”* (Swaminathan, Smith et al. 1998)

However, the lack of published work in this area suggests that these ideas have not yet been fully realised. Swaminathan, Smith et al. go further with their framework, suggesting that model specifications can be determined by the users themselves depending on their particular requirements:

*“Our vision is that simulation models are configured (not programmed) by selecting, instantiating, and composing sets of components to form an executable simulation model, without the need for extensive programming expertise. Thus our framework could be utilized directly by supply chain managers who are faced with specific configuration, contracting, or coordination issues.”* (Swaminathan, Smith et al. 1998)

Whilst there is no doubt that using or re-using pre-programmed sets of components will allow speedy development and, if suitable libraries are created, flexibility of design, the view taken in this thesis - based on experiences of working with industrial collaborators, as shall become clear in later chapters - is that simulation modelling is a very involved process requiring a number of specialised skills, and that it is therefore not feasible to expect managers to invest time in acquiring the skills to enable them to design, implement, and interpret the output of simulation models. An alternative approach, promoted in this thesis and elsewhere, is that stakeholder participation in models developed by teams of experienced researchers would facilitate the proposed customisation and speeded-up development.

Moss, Edmonds, et al. (2000) work on the ‘information filtering economy’ is a good example of an ABSS model producing statistical signatures of real markets, and having a well defined application area. The model environment consists of a digit string, where the values at each position represent information. A number of source agents exist, that have this information or parts of this information, and a number of customer agents each having demands for information relating to the value of the string at certain positions. The model is also populated by ‘broker’ agents that act as intermediaries by buying ‘chunks’ of information, breaking bulk and selling on individual digits to their customers. In contrast to the Parunak, Vanderbok, et al. model, the structure of the supply chain is very flexible: consumer agents can purchase either directly or from a broker, and there is no restriction on the number of intermediary tiers that can appear. The only condition imposed is that intermediaries must remain profitable to stay in the simulation.

The parameter that was found to determine market efficiency was the density of agents upon the grid. The density is determined by both the relation between the number of agents and the dimensions of the grid upon which they are located. The authors explored this by varying the grid size whilst keeping the number of agents constant. Since the agents are bounded in the suppliers with which they are acquainted (they can only contact local suppliers or suppliers recommended to them by other agents), there exists a critical density threshold above which customers cannot locate sufficient suppliers to satisfy their demand, and below which they nearly always can satisfy their demand. There was a parallel result that that the number of surviving brokers was very much larger below this threshold.

Other important results were that statistical signatures were found in the model output that are also typical of the patterns observed in empirical data. The authors described a power law distribution of brokers’ market share, clustered volatility in time-series data of individuals’ sales, and leptokurtosis.

It is easy to understand how this model could be useful in a real world context: if we suppose human users of the system are able to interact with broker agents in order to fulfil demands for different types of information. Rapidly changing information such

as stock prices, news articles, consumer goods prices, etc. is appropriate to this type of MAS specification. The advantages of such systems, developed along sociological principles, are that they have the properties of being flexible, scalable, and robust, and are therefore suitable to open systems where any number of users may participate.

Fioretti (2001) describes a model designed to explore the phenomenon of the survival of the industrial district for textile production in Prato, Italy. The author observed that it was surprising that a district with very high labour costs was able to compete with imported goods, because this is in contrast with the typical situation in textile production. He examined, via simulation with an agent-based model, one proposed explanation for this result: that competitive advantage was maintained through variety of finish i.e. 'feature flexibility'.

Fioretti designed a model composed of ten different types of agent (modelled on Pratese firms), each of which carries out one specific part of the production process. Buyers of the finished product contact 'middlemen' in the supply chain, who subcontract to other producers recursively and build production chains that will be able to complete the order. Product variability is measured by the variability of production chains that are constructed in this way. Simulation results reproduce very well the qualitative behaviour of the Prato industrial district: increasing variability from the end of the 1950's to the end of the 1970's, followed by a period of decreasing variability and industry concentration in the 1980's. The shift to feature flexibility in the 1980's and early 1990's is captured by another indicator: mobility of finisher-agents. Interestingly, it is shown that a 1:10 scale model does not capture all of these phenomena, and suggested that working with real sizes is crucial for empirical agent-based models.

This is an example of a well-validated model that is based on multiple data sources: upon the accounts of the historian Becattini, on research carried out on behalf of industry groups, and upon official statistics provided by the agencies. It is well validated on the macro data level but less well validated on the micro, for example the number of companies is not precisely known, and the structure of the acquaintance networks amongst them is not known.

## **Conclusion**

This chapter discussed the scientific underpinnings of social simulation and contrasted traditional approaches with the more recent Agent Based Social Simulation (ABSS). It was argued that the direct mapping of heterogeneous social actors to agents, and the ability to embed agents within a highly interactive environment, permits more flexibility in the kinds of sociological issues that can be explored through simulation. It was described how researchers working in the area of ABSS are concerned with modelling distributed, often large systems, involving much interaction amongst the components.

ABSS differs from traditional approaches in that the primary interest is in the untangling of processes rather than in finding general and analytic solutions for the model. It is focussed on exploring phenomena of emergence and complexity by taking a 'bottom up' approach to model design. Artificial systems built along sociological principles produce similarly complex and interesting behavioural patterns to those identified in real social systems. It is anticipated that this approach will ultimately provide a new and powerful set of techniques and tools for social simulation.

Key philosophical and practical issues of ABSS were discussed, from the concepts of social embeddedness and bounded rationality, to the differences between qualitative and quantitative approaches, and between fine- and coarse-grain levels of analysis. The qualitative approach to simulation involves a highly semantic style of programming, which although expressive in nature, is more difficult to perform computational operations with. Care must be taken not to conflate the semantic structure in the process of doing this computation because results then carry the danger of misinterpretation.

Formal models have the advantage of less ambiguous relationships between model and target system, but lack the information richness synonymous with qualitative research. The existence of a trade off between qualitative realism and level of formalism, which is necessary to keep the models understandable, was discussed. It

was argued that this thesis contributes to this discussion because it is placed at this crossing point of qualitative and formal approaches. Hence it brings to light some important methodological issues associated with the strand of representative ABSS.

A distinction was made between model experimentation and exploration - two important objectives of simulation - and the interplay between the two was discussed. Analysis of results focuses less on proving the properties of the model than on understanding the characteristic behaviour of the model and how this relates to the target system. The Strictly Declarative Programming Language (SDML) was introduced as a declarative, object oriented language designed for developing agent-based models of social and economic scenarios.

Two methods of validation were identified: one qualitative and one quantitative, and these compare very different aspects of model output with empirical data. One method compares the macro data and the other compares the micro data. Using both in conjunction, and independently, we have the cross-validation technique. Finally, the last section introduced some simulation models of the supply chain and intermediation as examples of the type of research taking place in this area.

Linking this chapter back to the work reviewed in chapter two, it has been argued here that traditional social simulation methods are less appropriate for the kind of research questions identified therein: questions highlighting the impact of technological change upon the supply chain. As these questions do involve large, distributed, complex systems exhibiting patterns of collective behaviour, it seems that the ABSS approach is a strong candidate for modelling them.

The following chapter describes how the case study investigation was devised to develop the enquiry. Preceding the description of the initial phases of the case study, the first part of chapter four reviews the literature on qualitative methodology, focusing on case studies carried out within an organisational context, as well as methodological approaches to information systems research. As will be seen, qualitative fieldwork will play an important role in informing the development of the agent-based model.