Boundedly *versus* Procedurally Rational Expectations

Scott Moss
Centre for Policy Modelling
Manchester Metropolitan University
Manchester, M1 3GH, UK.
http://www.cpm.mmu.ac.uk/~scott
s.moss@mmu.ac.uk

Contents

I don't see the point of not assuming that agents know the correct model of the economy. This is not economics.[1]

Economists who have relied on the rational expectations hypothesis are now seeking to demonstrate that rational expectiations equilibria can emerge in models with agents who are artificually intelligent. Agents' intelligence is represented by genetic algorithms. However, these algorithms misrepresent current understanding of human cognition as well as well known and long standing evidence from business history and the history of technology. A well validated representation of human cognition is implemented in SDML, a logic-based programming language which is optimized for representations of interactions among agents. Within that software environment, a model of a transition economy was developed with three production sectors and a household sector. The numerical outputs from that model are broadly in accord with the statistical evidence from the Russian economy. The model itself was developed explicitly to incorporate qualitatively specified characteristics of entrepreneurial behaviour in that economy. Unlike conventional economic models, transactions are negotiated and effected explicitly — there are no unspecified or

1. The whole, wonderfully revealing, report by a *Journal of Economic Dynamics and Control* reviewer on [11].

underspecified “markets”.

1 Rational and boundedly rational expectations

Rational and consistent expectations in macroeconometric models are justified on the grounds that it would be wrong to assume that econometric modellers were smarter than the agents who make the decisions that generate the time series data used to specify and estimate the model. If those agents are really as smart as the econometricians, and if the econometrician’s model of the economy is correct, then the agents will also have specified the correct model of the economy and it will be the same as that of the econometricians. One consequence of this line of reasoning is that, if all agents know the correct (econometrician’s) model of the economy, they therefore have the same model of the economy. By using this model to form their expectations, they must all have the same expectations and there is nothing analytically to distinguish one agent from another. The point can be put in two ways. Either there is a single representative agent or agents are homogeneous.

A number of modellers, mainly associated with the Santa Fe Institute, have been using algorithms from artificial intelligence and artificial life programs to represent heterogeneous agents. The clearest statement of the reasons for so doing is probably Sargent’s ([17], pp. 2-3):

> Rational expectations imposes two requirements on economic models: individual rationality, and mutual consistency of perceptions. When implemented numerically or econometrically, rational expectations models impute much *more* knowledge to the agents within the model (who use the *equilibrium* probability distributions in evaluating their Euler equations) than is possessed by an econometrician, who faces estimation and inference problems that the agents in the model have somehow solved. I interpret a proposal to build models with boundedly ‘rational agents’ as a call to retreat from the second piece of rational expectations (mutual consistence of expectations) by expelling rational agents from our model environments and replacing them with ‘artificially intelligent’ agents who behave like econometricians. These ‘econometricians’ theorize, estimate and adapt in attempting to learn about probability distributions which, under rational expectations, they already know.

The point of this exercise is to demonstrate than it is not necessary to impose mutual consistency of expectations on all agents in order to get simulation results which approximate the rational expectations outcomes. As Marcet and Sargent [10] put it, “the notion of a rational expectations equilibrium would be a more attractive one if there were plausible and undemanding learning schemes which would drive the system towards a rational expectations equilibrium. Even though not all models with heterogeneous agents yield approximately rational-expectations results, where they do, the features of the learning process are frequently (for example Timmermann, 1994) adduced as evidence that volatile prices and sales volumes are part of the convergence to rational-expectations equilibria.

Two recent papers yield some insight into the circumstances in which rational-expectations equilibria are likely to arise in computational models with heterogeneous agents. In one of these papers, Arthur, Holland, LeBaron, Palmer and Tayler [3] represent the traders in an artificial stock market as classifier systems which update their expectations by means of genetic algorithms. Each classifier is of the standard Holland type and represents a linear forecasting model with a limited domain. Over time, the population of classifiers converges to a piece-wise linear forecasting model. If learning takes place at a sufficiently slow rate (because the genetic algorithm is applied relatively infrequently — in their reported simulations, once in every 1,000 transactions), rational expectations equilibria always emerge as agents’ expectations converge to a common norm. Faster rates of learning (once in every 250 transactions) generate a market regime “in which psychological behaviour emerges, there are significant deviations from the r.e.e. benchmark, and statistical “signatures” of real financial markets are observed.

The other paper in this vein, Darley and Kaufman [6] find two similar regimes in dynamic non-cooperative games in which agents learn from more or less local interaction with their neighbours. States resembling rational-expectations equilibria (with shared agent perceptions) arise when prediction is easy (because each agent looks at the behaviour of a small number of neighbours so there is not much feedback but) not when prediction is difficult (because each agent notices the behaviour of a lot of other agents and, so there is a lot of feedback). When the model does not generate rational-expectations equilibria marketed by mutually consistent models or perceptions, then the output is meta-stable in the sense that it is marked by periods of stasis interspersed with periods of turbulence. Darley and Kaufman call these punctuated equilibria by analogy with evolutionary biology.

The suggestion here is that, in general, rational expectations equilibria can emerge from computational models in which heterogeneous agents develop models endogenously. The circumstances in which such equilibria emerge are likely to be characterized by slow rates of learning and low feedback. The faster agents develop their models, perhaps because of high feedback, the less likely is mutual consistency of perceptions as defined by commonality of predictive agent models.

While these results look interesting, they are generated by computational models with unknown analytical properties. Moreover, the learning procedures assumed for agents are arbitrary in the sense that there is no reason to believe that actual decision-makers learn in a manner which is described by those assumptions. Certainly, Herbert Simon, the inventor of the notion of bounded rationality rejects the Sargent view. Sent [18] describes the difference between the Sargent and Simon approaches to representations of cognition as one which turns on the descriptive accuracy of the representation of cognition as behaviour. Whereas Sargent justifies the use of genetic algorithms to represent cognition on their effective parallelism in computation, Simon argues that cognition depends on symbol processing. In effect, Sargent appeals to current views of the physiological basis of all mental activity while Simon appeals to experimental evidence about the epiphenomena of decision-making and learning as observed by experimental psychologists.

This difference is important for economists if either (a) the two approaches imply different theoretical and modelling structures yielding different relationships between actions such as policy measures on the one hand and the consequences of those actions or (b) one approach more usefully supports policy analyses than does the other. In the remainder of this paper, we consider only the second of these criteria.

2 Procedurally rational expectations

Although the prase “bounded rationality” was originated by Herbert Simon, it has been taken over by more conventional economists and redefined to cohere with mainstream economic theory. For Simon, bounded rationality implied limited information-processing and computational capacities. I do not know for certain who first redefined it, but certainly Williamson [21] claimed to encompass bounded rationality as limited access to information in his introduction of transactions cost economics. The difference is that, for Simon, bounded rationality entails the availability of more information than can be taken into account by decision-makers while for economists such as Williamson bounded rationality entails the paucity of information which is therefore a constraint in optimization procedures.

The effect of Williamson’s redefinition of bounded rationality was to leave in tact the underlying precept of mainstream economic theory that agent behaviour can be represented by some constrained optimization algorithm. The compulsion of economists to adhere to constrained optimization as the defining characteristic of human behaviour is manifest also in Sargent's specification of artificially intelligent agents. Genetic algorithms in the hands of Sargent or Arifovic to represent behaviour are different but they remain optimizing algorithms which arguably misspecify the nature of human cognition.

The argument that these implementations of genetic algorithms misspecify human cognition starts from the implication from cognitive science that cognition takes the form of exploitation of what we know and a highly directed exploration of our environment which is focused by our knowledge. This implications follows from the distinction in cognitive science between procedural and declarative knowledge.

Procedural knowledge is knowledge about how to do something and this knowledge is held by individuals in a way which does not allow it to be communicated directly to other individuals. Declarative knowledge is knowledge of what is true and can be communicated directly to other individuals. For example, an Englishman may have both procedural and declarative knowledge about the game of cricket. He can explain the rules of the game and describe or show a novice how to stand at the wicket or where to stand if he is to play off-stump or what to do if he is the wicket-keeper or the necessity of keeping the bowling arm straight at the elbow. All of this knowledge is declarative. To hit the ball successfully and place it where the batsman wants the ball to land or to spin-bowl so that the ball hits the ground and bounces so as to hit the wicket without coming into the range of the bat require abilities that can only be attained by practice. However well a person might know the rules and be able to describe the practices of cricket, that person will not be able actually to play cricket without acquiring substantial procedural knowledge.

Independently, this same distinction has been made by historians of business, the organization and technological change who demonstrated its relevance to these areas of economic activity. Edith Penrose [16] in her seminal (1959) analysis of the direction of the growth of the firm called the two types of knowledge objective and subjective. But her distinction between the two was couched in the same terms as Anderson's [2] discussion of the distinction between procedural and declarative knowledge. Similar distinctions — though not quite so explicit as in Penrose — are found in Chandler's (e.g. [5]) work on the development of organizational structures and Rosenberg's (1975, 198x) discussions of the determinants of the direction of technical change.

Because we cannot know everything, a reasonable assumption is that what declarative knowledge we do have comes from the activities in which we engage. How we use this declarative knowledge follows from our experience and, to the extent that experience is necessary to use declarative knowledge effectively, its use is governed by procedural knowledge. In other words, we start from what we know and develop new ideas and perceptions only by extending our experience.

Genetic algorithms, by contrast, search the whole of the environmental information space randomly and, if well constructed, evenly at the outset and concentrate increasingly on the parts of the information space that yield the best results. In the language of the field, genetic algorithms explore the search space and then exploit the subspaces described by classifiers that yield the greatest fitnesses. For cognitive scientists, human cognition takes the form of exploitation of what we know and a highly directed exploration which is focused by our procedural knowledge.

The difference could hardly be more important for how economists practice their discipline. Either agents are global optimizers and genetic algorithms can be used to represent that optimization in conditions of constrained information-processing and computational capacities or they can at best exploit their existing procedural and declarative knowledge in the hope of gaining some local (though possibly large) improvement in their circumstances. The substance of this difference is that the assumption of global optimization can evidently support the construction of models with no concern for the procedures by means of which agents actually go about collecting declarative knowledge and then developing their procedural knowledge where as the Simon approach requires the specification of precisely those aspects of cognitive behaviour.

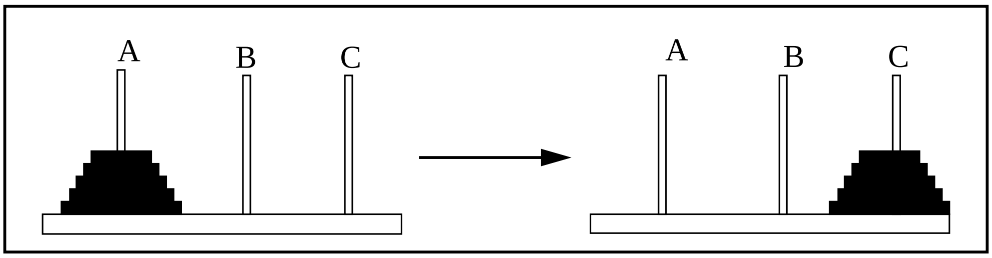
This brings us back to Simon's earlier distinction between substantive and procedural rationality.

Procedural rationality concerns the choice or development of procedures for taking decisions when the decision-maker has effectively limited capacities to process information and calculate appropriate outcomes. Certainly, procedural rationality entails satisficing. Our concern here is to find a representation of satisficing that will support models of decision-making in a macroeconomic environment.

The particular representation reported in this paper is drawn from several cognitive theories that have been implemented as computer software architectures designed to replicate data from psychological experiments. These architectures are Soar [9] and ACT-R [2]. Both of these architectures are based around the concept of a problem space architecture which itself is a tree structure of goals and sub-goals. The original specification of this goal and subgoal structure was by Newell and Simon (1972) as a planning algorithm. The sort of situation it might be used for in the Newell-Simon version was planning a trip from an office at MIT to an office in Berkeley. If the goal were to make that journey, a subgoal would be to fly from the nearest airport to MIT to the nearest airport to Berkeley. The subgoal of making that flight would be to get from the MIT office to the airport which would be undertaken by (say) car or taxi. To take the car would entail the subgoal of getting from the MIT office to the car by walking.

A classic problem on which to test artificial-intelligence algorithms is the Tower of Hanoi problem. This involves moving a set of discs of graded size from one peg to another, using a third peg as an intermediate step. The five-disc Tower of Hanoi problem is illustrated below. The discs can be moved one at a time and it is not permitted to place a larger on a smaller disc. The problem-space architecture for this problem, as specified by Simon [19] is to specify a subgoal of getting the top four discs onto peg B so that the largest disk and be placed on peg C and then to execute the next subgoal of moving the four discs on peg B to peg C. That move entails a subgoal of moving the top three discs to peg A so that the remaining disc cam be placed on the largest disc which is already on peg C. There is then a similar subgoal to get the three-disc tower onto peg A, and so on. Anderson [2] developed a program in ACT-R to learn to solve the Tower of Hanoi problem and compared the results of that program with the results of experiments with human subjects. He found that the students did indeed learn use a goal stack in the same way as ACT-R. The actual movements of the discs and the setting of goals and subgoals were accomplished in ACT-R by production (if-then) rules.

Figure 1: The Tower of Hanoi Problem



Three points about the ACT-R representation of cognition are relevant here. The first is that the results obtained from ACT-R programs can be compared with the results of psychological experiments to verify the accuracy of a program as a representation of cognitive behaviour in particular circumstances. Secondly, ACT-R is an encoding of an underlying theory of cognition. Thirdly, the representation of the problem space architecture as rules for moving up and down the goal tree and the rules for performing tasks to achieve each goal can, in principle, be obtained by the standard knowledge-elicitation techniques used for building expert systems.

Taking the first two of these points together, we have a means of encoding procedural knowledge about how agents learn which is informed and justified by a particular theoretical structure and discipline which is independent of the domain of application in economics or the management sciences. Discussions or arguments about the appropriateness of that particular encoding are not likely to be influenced by the results desired for economics models. The third point allows us to develop independent evidence to support a particular encoding of agents' procedural and declarative knowledge.

In the rest of this paper, I will report a pilot model of a transition economy. This model will be used to investigate the characteristics of procedures for learning and decision making which are validated in relation to cognitive science and verified in relation to economic time-series data. These procedures take for granted bounded rationality in the sense of Simon. These limitations preclude the assumption of optimizing behaviour. Encoding the process of goal formation, learning and declarative knowledge about the environment in a manner which corresponds to encodings in the cognitive sciences, we are able to determine whether procedures for forming expectations and perceptions about the environment are rational in the sense that action based on those perceptions is increasingly likely and, in any case, not less likely to further the attainment of agents' goals.

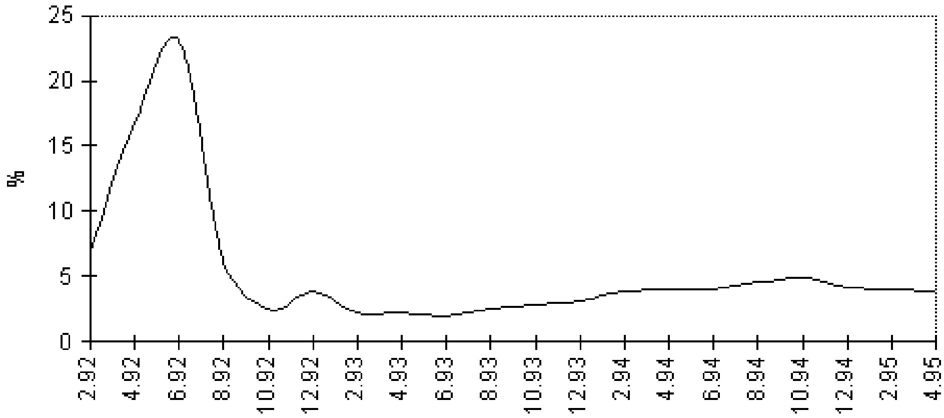
3 An emerging-market model

The model reported here was developed to capture certain stylized descriptions of the commercial environment faced by newly privatized and by state-owned enterprises in the Russian Federation.

One such fact of particular importance is the sharp increase in inter enterprise arrears which was not being repaid during the first half of 1992 — an episode known as "the arrears crisis". As shown in Figure 1, the volume of arrears peaked in the summer of 1992 at some 23 per cent of Russian GNP. The volume of arrears soon subsided to an average of about 5% for the period after 1992. Some experts find this figure to be not out of line with what is a normal amount of overdue trade credit by international standards [1]. A straightforward statistical comparison, however, can be misleading because enterprise arrears in Russia represent a different type of economic relations. In many cases the credit is forced, bears negative real interest and has no fixed repayment period or agreed repayment schedule. To a degree, these peculiar features were conditioned by the shortage of working capital, a drastic fall in demand and other economic consequences of a government's attempt to put an end to the regime of soft budget constraints by lifting price control and removing state subsidies to producers.Arrears in the industrial sectors as shares of GDP 1992-1995

Figure 2: Arrears in the industrial sectors as shares of GDP 1992-1995

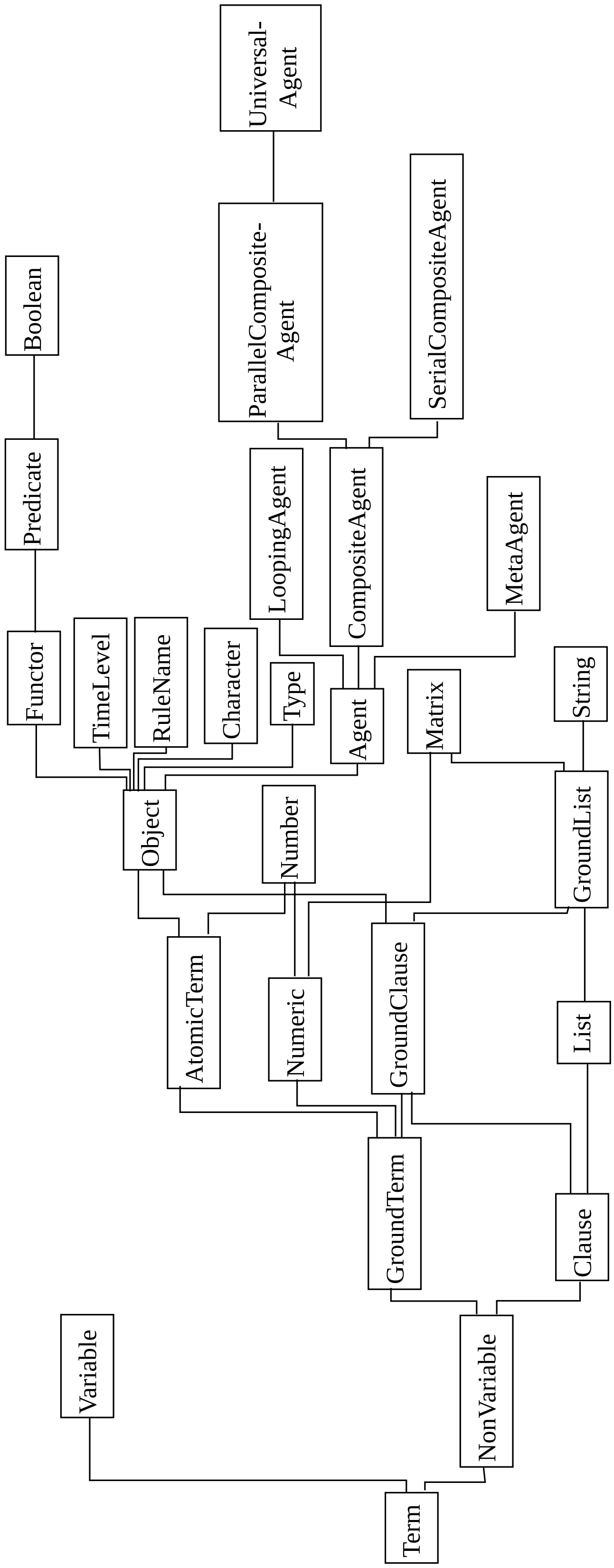
Source: Finansy promyshlennosti Rossii, The Monthly Bulletin of the Working Centre on Economic Reform of the Government of the Russian Federation, no.1, June 1995, p.2.



Moss and Kuznetsova [13] argued that the evolution of the arrears crisis provides a clear example of how enterprises cope with situations characterised by significant uncertainty. In the Russian case, enterprises were forced to adopt a survival strategy giving priority to existing, recognized constraints. There was no possibility to maximize anything in the framework of those constraints (Aukutsionek, 1995). Indeed, the scale of the accumulated debt and its persistence suggest that enterprises that debt reduction was not a high priority. For one thing, both the liquid assets of enterprises and their debts have been growing simultaneously [4]. The debt became an element of their survival strategy being instrumental in prolonging the existence of a business environment they were accustomed to, i.e., the one governed by soft budget constraints. Because the accumulated bad debt grew out of proportion on the national scale and became commonplace in all industrial sectors, this important business indicator ceased to be seen as a symptom of poor management efficiency. By the autumn of 1992, 95% of enterprises had bad debts enough to be proclaimed bankrupt on legal grounds [22]. Debt performance had become separated from the business performance of a firm.

The Russian arrears crisis provides us with sufficiently clear stylized facts that we can assess whether the outputs from simulations conform to those facts or not. The stylized facts we want to capture, in addition to the arrears increases, are a high average and widely fluctuating rate of price inflation and volatile but trendless outputs. The point is to capture these stylized facts using a credible and validated specification of the processes agents use to build up their own models of their environments.

Figure 3: : The Basic SDML Type Hierarchy



3.1 The modelling language

The model reported here was implemented in SDML, a strictly declarative modelling language which corresponds to strongly gounded autoepistemic logic (SGAL)(*cf.* [7], [12]). This means that any model which runs under SDML is formally sound and consistent with reprect to the axioms and rules of inference of SGAL. Consequently, models written in SDML can entail qualitative as well as numerical relationships without loss of formal clarity and rigour. This is an important issue which is taken up in some detail in Section 4.1 below.

SDML has a number of object-oriented features which make it particularly useful for modelling cognition along the lines described in Section 2. The particular object oriented features supporting the model reported here are the type (or class) hierarchy and the container hierarchy.

The type hierarchy is similar to the class hierarchy of (say) C++ but where C++ has simple inheritance (each class inherits the methods and instances of one superclass), SDML has multiple inheritance. The basic inheritance class is reproduced as Figure 3 from [cmot ref].

The user adds further subtypes, in particular, subtypes of Object, Agent and SDML's predefined Agent subtypes. The type Agent is distinguished from Object in that it has rulebases associated with it. The number of such rulebases varies with the time levels defined by the user. Time levels are discussed in several contexts below.

Agent is the principal type of interest here. Models are specified in terms of instances of agents but these will not normally be instances of the type Agent but of a user-defined subtype of Agent or one of its predefined subtypes (or of more than one of these). Clause definitions and rules are specified in types and are inherited from them by their instances. In this way, the rules for a number of identical agents can be defined in a shared type. Similarly, agents which are not identical may nevertheless share certain rules by means of a common supertype.

Abstract supertypes such as ParallelCompositeAgent or LoopingAgent add particular functionality to agents. The instances of every subtype of CompositeAgent can contain other agents (its subagents). Instances of subtypes of ParallelCompositeAgent contain subagents the rulebases of which fire in parallel. Instances inheriting from type SerialCompositeAgent fire their rulebases in a previously specified order. LoopingAgents loop over time periods and there can be an arbitrary number of such time levels.

Finally, any agent can contain an instance of a subtype of type MetaAgent. Meta agents can assert statements to, and retrieve statements from, their containers' rulebases in the same way that all agents assert to and retrieve from databases. The main difference is that meta agents can only write rules to and read rules from their containers' rulebases whereas they and all other agents can read and write satements conforming to any previously defined clause definition when these statements are held on databases. Meta agents are used to devise the sort of agent routines discussed by Nelson and Winter [14] but to do so as a result of some representation of cognitive activity. these routines represent procedures which are the best the agents and their meta agents have so far found in their attempt to meet their current aspirations.

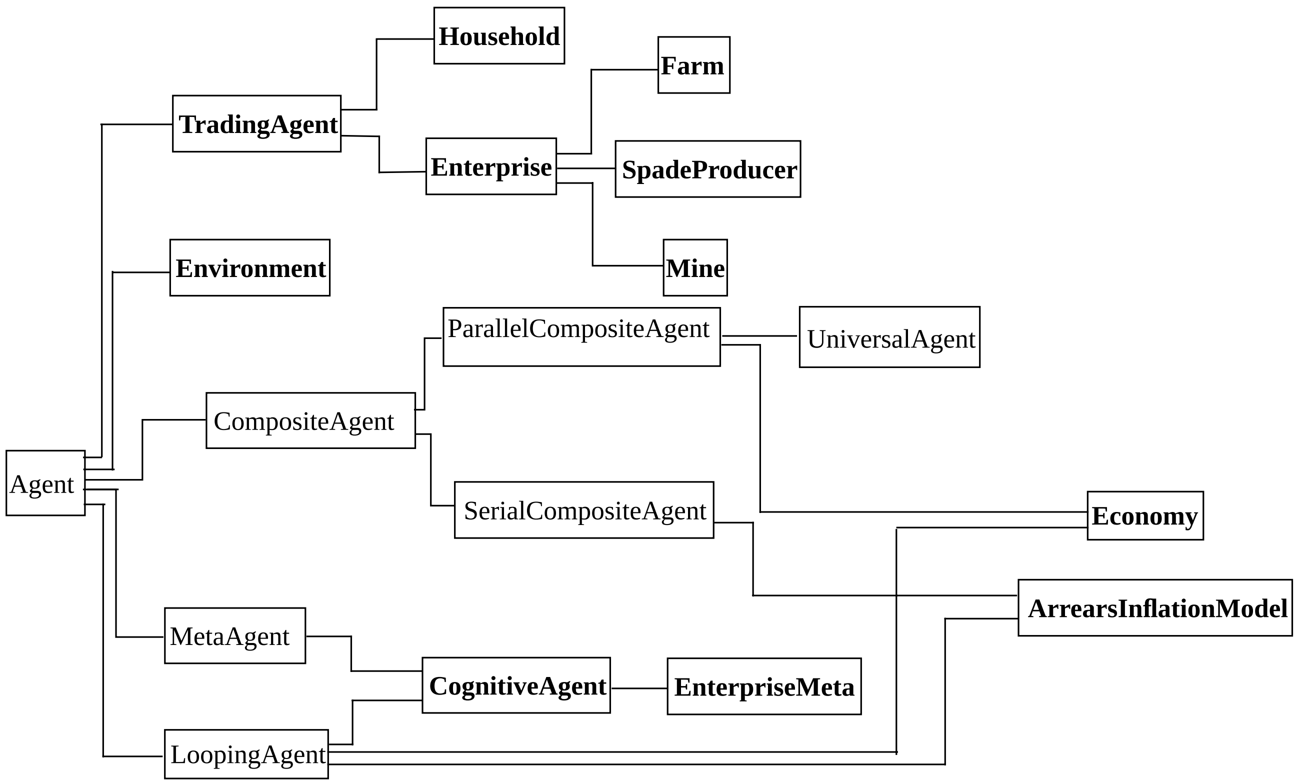
So what we want to do is to use the type and container hierarchies to put together a model in which enterprises learn about each other in the course of finding enterprise routines that support their goals and aspirations.

3.2 Model structure

The type hierarchy defined for the Russian transition model is reproduced in Figure 4 and the container hierarchy in Figures 5 and 6. The type symbols in boldface in Figure 4 are user defined.

All of the common features of agents that engage in transactions are implemented in the TradingAgent type. In the main, type TradingAgent defines clauses which are used to effect transactions. These clauses include “orderPlacedWith <ProductGroup> <TradingAgent> <Number>” and “orderReceivedFrom <ProductGroup> <TradingAgent> <Number>”. The symbol <*Type*> indicates an instance of that type. so if a household (say *household-18*) decides to purchase 12 units of corn from a farm (say *farm-3*), it would write “orderReceivedFrom corn household-18 12” to the database of *farm-3*. At the same time, it would write to its own database “orderPlacedWith corn farm-312”. Similarly, if *farm-3* decided to accept the order, it would write to the database of *household-18* “purchasedFrom corn farm-3 1.305 12” where the price is 1.395 and the quantity 12 units of corn. In order to remember the sale itself, *farm-3* would write to its own database “saleTo corn household-181.305 12”. In this way the transaction would have been proposed and agreed by all of the different types of agent using the same language. However, households and enterprises behave differently in a number of ways in the model so that those clause definitions and rules which govern the behaviour of each of these subtypes of trading agents separately and implemented in their respective types.

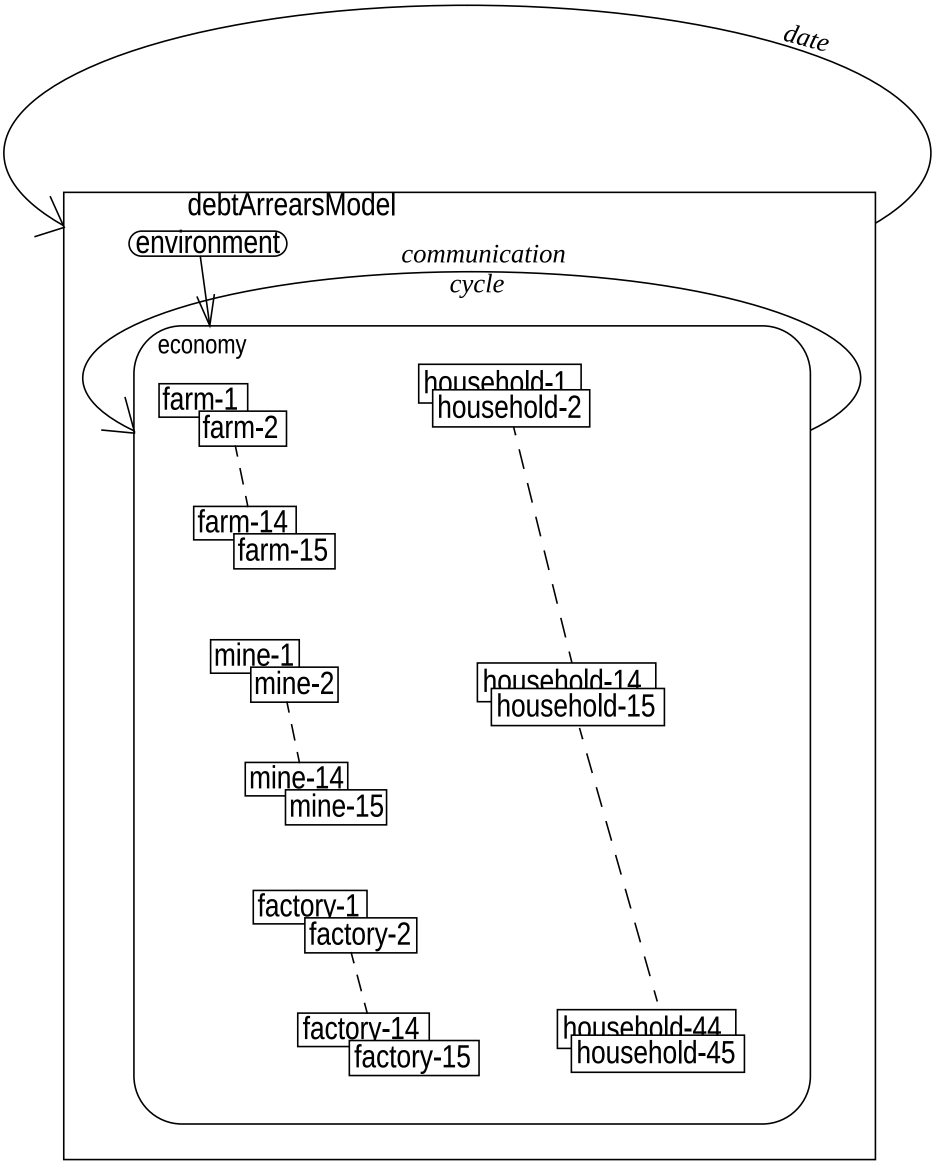
Figure 4: Hierarchy of agent types



In order to understand the nature of the subtypes of CompositeAgent and of LoopingAgent, it is necessary also to consider the container hierarchy of the model.

The outermost container is debtArrearsModel, an instance of type ArrearsInflationModel. The type ArrearsInflationModel is a subtype of LoopingAgent from which it inherits all of the rules and clause definitions required to enable the model to loop over the time levels defined for instances of ArrearsInflationModel in the computational model. Type ArrearsInflationModel is also a subtype of type SerialCompositeAgent from which it inherits the rules and clause definitions to support subagents which fire their rules in a sequence determined by the agent. In this case, the agent environment fires its rules before the agent economy. The agent environment fires its rules once at each date. It is a simple agent the function of which is to introduce representations of environmental changes such as natural disasters into the simulations.

Figure 5: The macro container structure



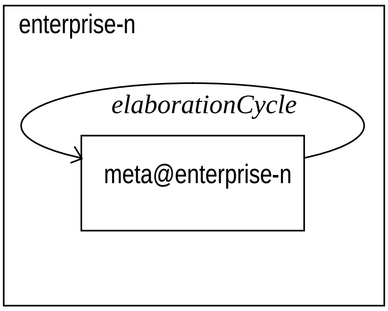
The agent economy is an instance of type Economy which itself is a subtype of ParallelCompositeAgent and of LoopingAgent. As an instance of ParallelCompositeAgent, it contains agents which fire their rules effectively in parallel. This is an important feature in a model which represents transactions as the outcomes of communication among agents and where communication is represented by the assertion of clauses to the databases of other agents or some other database which is common to both. Since the agents are active at the same time, we have to represent the fact that a communication can be received only after it is sent. This means, in terms of any computational model, that an agent can retrieve any message written by another agent only at a time period subsequent to the period in which it was written. So, in order to effect a transaction, agent-1 will assert an order for goods to the database of agent-2 at time *t;* agent-2 will retrieve that assertion at time *t+1* and assert its acceptance of the offer to the database of agent-1. This acceptance will be retrieved by agent-1 and time *t+2* and the agreement to transact has been concluded.

In this model, transactions take place each date but the negotiations required to effect them we now see will take more than a single time period. For this reason, several communicationCycles are allowed each date. The clauses shared by instances of TradingAgent include, for example, purchasedFrom and soldTo each of which has arguments to identify the product, the price, the amount and the trading partner.

Instances of type Enterprise represent cognitive agents. In this model, cognition takes the form of the building of mental models which are used to create rules for guiding the decisions of the enterprise. These decision rules are what Nelson and Winter [14] called “routines”.

The building and assessment of the mental models are undertaken by subagents of the enterprises. These subagents are instances of type EnterpriseMeta which is a subtype of MetaAgent. Meta-agents can use the *rulebases* of their containers and other subagents of their containers as *databases*. In this model, each instance inheriting from type Enterprise contains a meta-agent of type EnterpriseMeta. All instances of EnterpriseMeta build mental models in a procedure derived from computational cognitive science. This involves identifying goals and subgoals and the tasks needed to achieve those goals over a number of time periods called elaboration cycles. One of the tasks to be completed in this process is the writing of decision rules to the rulebase of the containing enterprise. For this reason, type EnterpriseMeta is itself a subtype of LoopingAgent as well as MetaAgent and it loops over time level *elaborationCycle*. The articulation of instance of type Enterprise, expanded from Figure 1, is given in Figure 6.

Figure 6: The enterprise container structure



3.3 The model setup

In order to capture the main, essential elements of the position of Russian enterprises, we assume the decision variables of the enterprises to be planned output, output price, the wage rate, the offer of employment, orders placed with suppliers and payments to creditors. In addition, the enterprises choose the suppliers with whom they place their orders and the customers whose orders they will fill in whole or in part. Each enterprise notes at each date whether its suppliers have filled the orders placed with them and it customers have paid for the goods previously supplied to them. These notes take the form of endorsements attached to the enterprises records of its customers and suppliers. Orders are allocated among known suppliers in proportion to their records of reliability. Sales are allocated first to orders from known customers with the best records of payment. In effect, each enterprise builds up models of the enterprises with which it trades.

These endorsements are also used by enterprises to formulate views about which other firms are the most successful. It is natural to assume that those suppliers which are best at supplying orders and those customers which pay most quickly are also the strongest enterprises. On this basis, enterprises take into account any observable information they have about these trading partners and assume that their behaviour is highly functional. In the model reported here, the only information which one agent can observe about an enterprise is the output prices it sets, its employment of labour and information arising from the transactions in which they engage (supplies, orders and payments). Thus, if one enterprise observes its best trading partners lowering (or raising) prices, it will assume that to lower (or raise) prices increases the values of goal variables and, so, will conjecture a model to that effect.

The goals of the enterprises are sales volume and cash. There is no attempt at optimisation of these values but, rather, the agents seek strategies which will increase the value of one or the other of these goals. In the present set-up, neither is given pride of place. In the event that changing the value of one decision variable is expected to increase the value of one goal value and diminish the other, then the action which is considered most likely to have the anticipated outcome will dominate the decision. If the agent has more confidence that the goal value diminution will occur then he will change the decision variable value to reduce or prevent the diminution. If he has more confidence that the other goal value will be increased, then he will change the value of the decision variable in the appropriate direction.

The pre-defined intermediate variables observed by the enterprises are their own purchases, their own sales, their current stocks of real goods (inputs and unsold outputs), their current financial asset holdings (only cash, so far), and the wage bill (the product of the wage rate paid and employment by the enterprise).

The wage bill is the only intermediate variable to be calculated from other variables observed by the enterprise. It gets this special treatment because we assume that it is paid in the same period as the employment it covers. This assumption itself seemed appropriate because of the relatively insignificant incidence of delayed wage payments and also because, in inflationary conditions, the impact of delayed wage payments is much the same as offering a lower wage rate. A rather more elaborate setup would be required directly to capture the effects of wage arrears. If such a model were thought likely to be useful, it would be a straightforward extension of the model reported here.

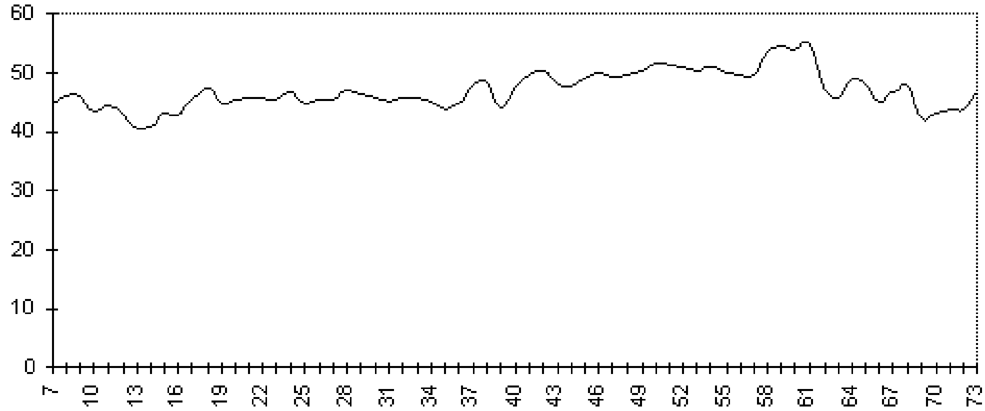
In general terms, the set-up reported here was devised only to capture a coarse-grained account of the development of Russian enterprises and to demonstrate how our modelling techniques perform on problems relating to the emergence of new markets and market institutions.

3.4 Results

Our first simulation set-up included a specification of the input-output relations and the inclusion as model variables of inter-enterprise debts as well as payments and the usual economic variables of employment, prices, wage rates, production decisions, actual outputs and sales. The Russian experience is of steady, even rising, employment, rapidly rising prices, growing debt and declining output and sales.

We have run the set-up for 73 periods. What we get is not too dissimilar to what we observe. Employment, for example, does not show any dramatic changes at all although, reflecting variations in production activities over the whole of the simulation, it has not been stable. The time path of employment is given in Figure 7.

Figure 7: Total employment (simulated)



In production we observed a collapse after an initial surge (due probably to initial simulation conditions) but starting from date 7 production trends varied from sector to sector. The production of corn remained remarkably stable because corn has at least one relatively stable source of demand in the form of households. The production of both spades and iron shows considerable oscillation but it develops accordingly with the picture of unit sales which demonstrates certain cyclical pattern. Series for production is shown in Figure 4 and for unit sales in Figure 8.

Figure 8: Sectoral production (simulated)

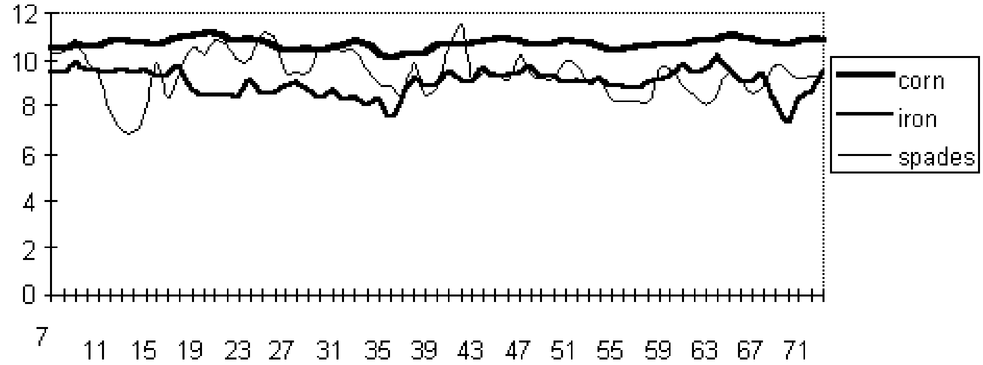
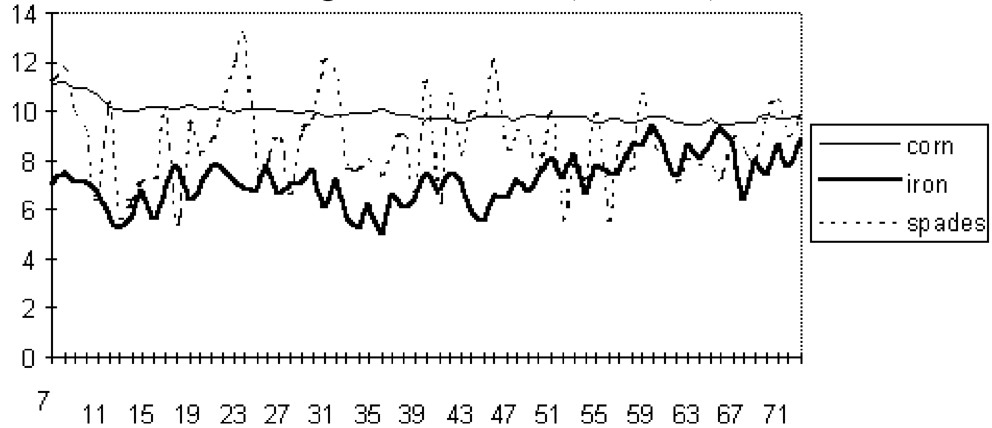
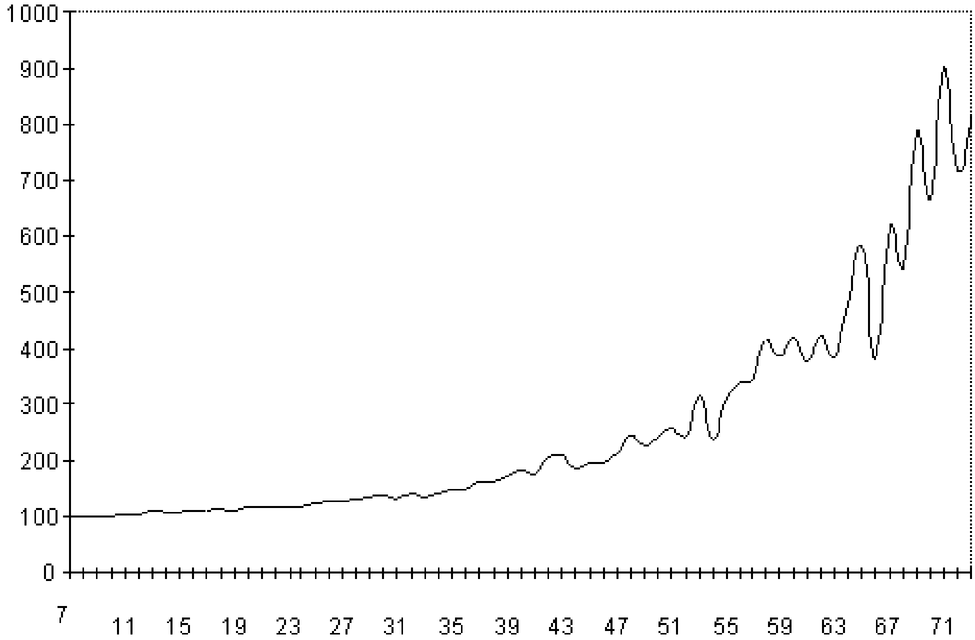


Figure 9: Unit sales (simulated)



A key result in the development of the simulation setup was the price series. The series, reproduced as Figure 10, now reflects the stylized facts of the Russian experience. However, earlier models generated trendless, though moderately volatile, price series. The achievement of the price series of Figure 10 was obtained by specifying credible information sources.

Figure 10: Paasche price index (simulated)



3.5 Procedural rationality as modelling

The goals of the enterprises are sales volume and cash. The numerical controls are price, wage rate, employment, input demands and debt payments. In addition, each enterprise has to determine the allocation of its orders for inputs and the allocation of its debt repayments.

There is no attempt at optimisation of the goal values but, rather, the agents seek strategies which will increase the value of one or the other of these goals. In the present set-up, neither is given pride of place. In the event that changing the value of one decision variable is expected to increase the value of one goal value and diminish the other, then the action which is considered most likely to have the anticipated outcome will dominate the decision. If the agent has more confidence that the goal value diminution will occur then he will change the decision variable value to reduce or prevent the diminution. If he has more confidence that the other goal value will be increased, then he will change the value of the decision variable in the appropriate direction.

Agent cognition is represented by a process of model building which has more in common with the cognitive sciences that with Holland classifiers as used by Sargent, Arifovic and others in the economics community. The modelling strategy of each enterprise in the simulation model is to formulate its own models relating selected control variables to at least one goal variable. Initially, following Moss (1995), control variables and goal variables were combined at random and the control variable was either increased or decreased with equal probability. There was a standard generate-and-test procedure so that enterprises kept those of their models that yielded improved goal variable values and abandoned those that did not. Agents kept track of how good their models were by a process of endorsement first suggested by Cohen (1985) and implemented by Moss (1995). In general terms, models were endorsed as successful at a date when they were used to formulate an action undertaken in the same period that a goal variable value was improved. They were endorsed as being unsuccessful when the goal value was not improved. Models that were usually successful or usually were so endorsed. This would happen when four of the five most recent applications of a model earned the successful or unsuccessful endorsements, respectively. Collections of endorsements mapped into numerical values with positive endorsements increasing overall endorsement values and negative endorsements reducing overall endorsement values. The probability of applying a model is proportional to its overall endorsement value. Models with negative overall endorsement values are abandoned.

Enterprises also endorsed other agents as being reliable or unreliable suppliers, reliable or unreliable debtors and so on. The resulting overall endorsement values attaching to these other agents were used to determine suppliers and, when orders exceeded stocks of outputs, the order in which to allocate outputs to customers. In this way, stable trading relationships were established among agents in the model.

An important feature of this model was the absence of any *deus ex machina*. There was no auctioneer, no bulletin board on which to post prices, supplies or demands for all agents to see. All information was communicated directly by one agent to another. Learning and expectations about the behaviour of other agents and the system as a whole resulted from an explicit cognitive mechanism which gains its credibility from disciplines outside economics. Procedural rationality entails the selection of information when more is available than an agent can process. It also involves identifying relations from that data without relying on it being correct in some fundamental sense. If the relationships suggest actions which improve on an agent’s present position, then they constitute a good model. Otherwise, the model needs to be abandoned or revised. An important means of understanding for humans is the forming of analogies. The analogy in the simulation environment reported here took the form of applying behaviour by agents who appear to be successful to one’s own behaviour. If, for example, I know that the farmer to whom I sell my spades pays his bills immediately and is raising the price of corn, then that farmer must be doing well enough to pay his bills so I will try raising the price of by spades in order to do well myself.

What did not work (i.e., did not conform to the stylized facts) in the simulation experiments that led to this model was random generation and testing. Holland classifiers rely on a generate-and-test algorithm, too, but on a much larger scale since a population of models would be generated, mutated, crossed-over and tested at each date rather than just the one model selected by the agent in our simulations. We know that such models can yield outputs which converge towards rational expectations equilibria. This issue of the relationship between simulation outputs and rational expectations equilibrium is clearly of some importance to economists. It is, therefore, addressed in the following section.

4 Model validation and verification

The validation of a computer program is the process of applying formal methods to ensure that a program design will achieve what is expected of it in appropriate conditions and, in particular, that it will not get into a confused or illegal state. If a program runs without error in any computer programming language, then that program is consistent and sound relative to that language. That is, the program does not generate or entail mutually contradictory statements and it does not generate statements which the language does not support. Consequently, program validation entails ascertaining that the program is consistent and sound relative to a formal statement of the properties of the programming language.

In this section, I argue that validation should be an important issue in the specification of economic models and, in particular, of economic cognition. Two aspects of validation are considered: validation with respect to logical formalisms and validation with respect to cognitive theories.

4.1 Logical validation

The virtue of validating the consistency and soundness of a model relative to a logical formalism is that it removes ambiguity from the specified relationships comprising the model. The particular formalisms that economists rely on are mathematical systems which are well suited to optimizing functions subject to well specified constraints. Such mathematical bases are inappropriate for the model reported in Section 3 because of its strong reliance on qualitatively defined variables such as endorsements and because there is no element of optimization in the model. Nonetheless, the model is consistent and sound relative to at least one logical formalism (and, though unproved, probably many such formalisms).

The logical formalism under which the transition model is consistent and sound is a fragment of strongly grounded autoepistemic logic (FOSGAL)[8]. The proof of the consistency and soundness of the model relative to FOSGAL runs as follows:

If a programming language corresponds to a logical formalism, then any program viewed as a set of statements or sentences which runs in that language will necessarily be sound and consistent relative to that logical formalism. One such language, implemented precisely to capture this feature of programs, is SDML11 which corresponds to FOSGAL. This particular logical formalism of SDML has emerged as one which supports the kind of multi-agent, strictly declarative modelling favoured by the SDML user community. It is by no means the only possible or appropriate logical formalism for modelling organizations. Indeed, the choice of logical formalisms for different classes of problems is already a fruitful field of enquiry in the artificial intelligence literature. A natural extension of the work described in this paper is the explication of the properties of appropriate logics to underpin a language of discourse for the management and economic sciences.

In SDML, and therefore in the model reported here, each agent is defined on a rulebase and database for each period of time. Every rule in the rulebases and every clause asserted to the databases is sound and consistent relative to strongly grounded auto-epistemic logic. If any were not, then the model would not run and inconsistency error would be reported.

FOSGAL emerged as a good logical basis for modelling agent behaviour and interaction because of its encoding of negative knowledge. Such encodings are well recognized to be difficult simply because it is not in practice possible to store all the facts that are not true. For this reason, SDML follows the conventional practice of storing only positive knowledge and dealing with negation by allowing rules which have ‘not inferred’ operators in their antecedents.

SDML was designed principally as a forward chaining language because drawing inferences from a set of beliefs and facts and then remembering those inferences by asserting them to a database seemed a more natural representation of agent reasoning than backward chaining in which the implication is stated and then the antecedents evaluated to see if the assertion can be justified.[2] In order to perform forward-chaining efficiently, SDML will fire rules but keep track of any assumptions it had to make on the way. This helps to minimise any back-tracking to try alternative assumptions in the more commonly occurring situations. Thus SDML sometimes needs to make inferences from its own *lack of inference* of certain facts, just as in SGAL one can infer from one’s own lack of belief.

2. Backward chaining is also supported by SDML. Its main uses are in recursion, especially on lists, and for using one clause of obvious meaning to replace several clauses in forward-chaining rules.

4.2 Theoretical validation

The position we have now reached is that a model written in SDML is consistent and sound relative to a particular formal logic which is well suited to the development of computational models involving interaction among agents which takes the form of direct communication of information. Genetic-algorithm representations of agents are doubtless consistent and sound relative to the mathematics underlying Holland classifiers. As far as that goes, we have two, alternative representations of cognition which have equal claim to logical rigour. How then do we choose between them?

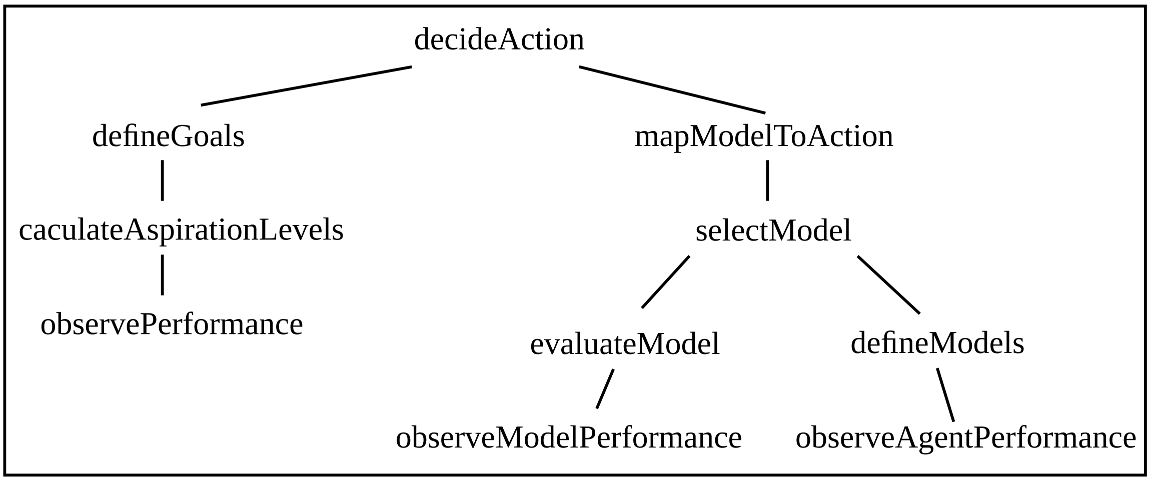
It seems likely that the appeal of genetic algorithms to economists such as Sargent is that they are optimizing algorithms. Not all problems are well conditioned to be optimized using genetic algorithms in general and Holland classifiers in particular. I imagine that this feature of GAs adds to their allure for someone who, like Sargent, wants to argue that the *possibility* of a failure actually to optimize gives the agents in the model an attractively human fallibility previously lacking in the agents of economic theory. I am not, however, aware of any independent argument that Holland classifiers are accurate representations of human cognition. Their main virtue seems to be that, under some circumstances at lease, computational models representing agents as Holland-type classifiers yield outputs that converge over simulated time towards rational expectations equilibria.

To find that a result can be achieved in a variety of different ways and, in particular, using a variety of different approaches is a standard means in the natural and mathematical sciences of building confidence in the result. That GAs can be used to generate rational expectations equilibria in computational models thus enhances confidence in the importance of the rational expectations hypothesis. If the attempt to generate the same result with different techniques sometimes leads to the intended result and sometimes does not, then the failure can itself lead deepen our understanding of the phenomena under investigation. One possibility is that a number of different approaches to the investigation yield a similar set of conditions under which the original result can be expected. In this way, we begin to identify the conditions of application of, in this case, the rational expectations hypothesis.

One such development looks to be emerging from the papers by Arthur *et.al.* and by Darley and Kaufman cited earlier. These papers follow Sargent in relying on Holland classifiers to represent key aspects of cognition. A complementary approach in the investigation of the conditions in which we might expect rational expectations equilibria to arise is suggested by computational cognitive science. Both Soar and ACT-R offer points of departure for this approach. Although we do not yet have complete results, we can exhibit the key features of and some early results from an implementation of key ideas from cognitive science into the transition model reported above.

The core concept drawn from both Soar and ACT-R (and originally from Newell and Simon [15]) is the problem-space architecture (PSA). The PSA rests on the experimentally verified idea that, in undertaking complex decisions, we engage in a process of sub-goaling. What is involved here is readily seen from the PSA devised for a new version of the model reported in Section 3.

Figure 11: Transition enterprise problem-space architecture



The label of each problem space in Figure 11 indicates a task to be completed. The top-level task (decideAction) which must be completed each date is to decide what actions to take. Examples of the actions to be decided upon are price levels, wage rate, planned output, input demands and how much of current cash resources to pay out. Actions are predicated upon goals so that, before deciding on an action, the agent must determine what the action is meant to achieve. In this model, aspiration levels are set for the various goal variables and this is done on the basis of the enterprise's observation of its own performance. Once the goals are defined, it is necessary to translate perceptions of the environment into some action. Since these perceptions are represented by mental models of the agents, an appropriate model must be selected. The selection will be from existing models and new models. The existing models are evaluated as they are used and the evaluations are remembered as endorsements. New models are defined on the basis of observations of the behaviour of the most successful (best endorsed) of the enterprise's trading partners. Consequently, the performance of these trading partners must be observed and evaluated.

Declarative knowledge is composed of facts which can be retrieved from one database of another by the agent and some relationships encoded as mental models. Procedural knowledge is composed of the means of mapping models into rules of action. There is also some common knowledge such as the impossibility of increasing outputs without increasing inputs which is declarative in the sense that agents could in principle communicate such knowledge to one another. Such knowledge is encoded as models which cannot be eliminated from the databases of an agent.

Models (apart from the common-knowledge models) and the memory of trading relations can be retrieved by agents with a probability related to their importance and the length of time since they were last retrieved. The scheme used to determine the probability of retrieval at need is taken from Anderson [2]. In particular, the odds in favour of retrieval are $\sum_{j=1}^{n} a t_j^{-d}$ where the t_j are the lags since the jth prior retrieval of the endorsement. d is a positive parameter determining the half-life of the influence of a prior retrieval on current retrieval. The value of a is determined in the simulation model by the endorsement values of models and the agents.

One virtue of this formula is its consistency with experimental data about memory. The choice of the d and a parameters is not determined by that experimental literature. Setting those values is an empirical issue to be considered presently. The precise specification of the PSA is also an empirical issue. In terms of validation, however, it seems much more robust to develop representations of cognition which, like this one, can be assessed on a basis which is independent of the application rather than to assess an arbitrary representation of cognition on the basis of its convergence to a rational expectations equilibrium.

4.3 Verification

While the PSA specified in Figure 11 conforms to similar models developed for social simulations (see, for example, [23] or [12]), we have not yet sought independent evidence of its descriptive accuracy. The source of such evidence would naturally come from knowledge elicitation exercises with enterprise managers in the Russian Federation. Such an exercise would also inform our assumptions about goals and goal-conflict resolution.

Parameterizing the model is computationally expensive and requires reliable data series. Previous models produced by the Centre for Policy Modelling have been parameterized using genetic programming algorithms. These extend genetic algorithms using Holland classifiers by representing data as tree structures which are subject to mutation and cross-over at points where usable programs result. Edmonds and Moss (1997) report this technique in detail. The fitness functions are typically related inversely to root mean-squared errors and are also biased in favour of parameters which look reasonable to domain experts. In this way, we not only get results from application to hold-out data sets that are at least as good as ordinary regression methods applied to the same data but also results which reflect at the same time domain expertise.

5 Conclusion

A necessary but not sufficient condition for simulation models to converge towards a rational expectations equilibrium is that agent perceptions converge so that mean expectations are close to model outcomes. The Sargent research programme has as its objective the demonstration that non-homogeneous agents who search the whole possibility space will converge on perceptions which are effectively consistent with the outputs of the system model. The transition models described above generate convergence in behaviour — specifically in price-setting behaviour — by local exploitation of declarative knowledge. These models were not implemented with rational expectations equilibria in mind. They were implemented to demonstrate that models which are well validated with respect to both formal logics and independent theoretical structures relating to learning and decision-making also lead to verifiably accurate model outputs.

I conjecture that any cognitive behaviour which entails the convergence of individuals' perceptions and where their perceptions are changed when systematically wrong will also entail convergence towards expectations which are not wrong in any biased way. If so, then ultimate convergence towards a rational-expectations equilibrium or something similar is a weak condition in the absence of structural change in the economy. However, the process of learning by global search and the process of learning by local exploitation might well yield very different results in relation to the effects of economic and social policies which are intended to effect particular structural changes. Before implementing policies based on one of these approaches or the other, it would make sense to look for reasons to believe in one or the other. Validation with respect for formalisms — mathematics or predicate or propositional logics — ensures rigour in the sense of a lack of ambiguity in specifications and implementations of models. Validation with respect to independent, experimentally or otherwise verifiable theories gives us confidence that our own models are better than just data-mining. I can see no reason to have confidence in the encoding of human cognition as genetic algorithms simply because the resulting models converge to something which is not too different from conventional rational-expectations-type results. This scepticism stems from the absence of any independent reason to believe that agents learn by global search and the plethora of evidence that they learn by local exploitation of declarative knowledge conditioned by their procedural knowledge. Since procedural knowledge can only be acquired by experience, the identification and use of declarative knowledge must always be based on the experience of what the agent has done and this is itself an inherently local processing of declarative knowledge.

Experience indicates that economists are unlikely to be influenced by the very different style of process-centred disciplines such as cognitive science. Nonetheless, we have shown the feasibility of implementing models using concepts which have arisen independently in analyses of business history, the economic history of technical change and in the cognitive sciences to explain with empirical verification how learning and decision-making actually take place.

References

[1] Alfandari G., Schaffer M. (1995) "On 'Arrears' in Russia", Paper presented at the Joint Conference of the World Bank and the Ministry of Economy of the Russian Federation, St.Petersburg, June 12-13.

[2] Anderson, J.R. (1993), *Rules of the Mind* (Hillsdale NJ: Lawrence Erlbaum Associates).

[3] Arthur, W.B.,J. Holland, B. LeBaron, R. Palmer and P. Tayler (1996), “Asset Pricing under Endogenous Expectations in an Artificial Stock Market”, University of Wisconsin-Madison, Social Systems Research Institute 9625.

[4] Belianova E. (1995) "Motivation and Behaviour of Russian Enterprises" [in Russian], *Voprosy ekonomiki*, no.6, pp.15-21.

[5] Chandler, A. (1975), *The Visible Hand*, (Cambridge MA: Harvard University Press).

[6] Darley, V. and S. Kauffman, “Natural Rationality” in W. B. Arthur, et. al. (eds), *Economy as an Evolving Complex Evolving System II* (Reading, MA: Addison-Wesley).

[7] Edmonds, B., S. Moss and S. Wallis (1996), “Logic, reasoning and a programming language for simulating economic and business processes with artificially intelligent agents” in Ein-Dor, Phillip (ed.) *Artificial Intelligence in Economics and Management* (Boston: Kluwer Academic Publishers), pp. 221-230,

[8] Konolige, K. (1988), “On the relation between default and autoepistemic logic”, *Artificial Intelligence*, v. 35, pp.343-382.

[9] Laird, J.E., A. Newell and P.S. Rosenbloom (1987), “Soar: An architecture for general intelligence”, *Artificial Intelligence*, v. 33, pp. 1-64.

[10] Marcet, A. and T. Sargent (1992), “The Convergence of Vector Autoregressions to Rational Expectations Equilibrium” in Vercelli, A. *et. al.* (eds), *Macroeconomics: A Strategic Survey* (Oxford: Oxford University Press).

[11] Moss and Edmonds (1997), “Modelling economic learning as modelling”, *Cybernetics and Systems (forthcoming).*

[12] Moss, Gaylard, Wallis and Edmonds (forthcoming), SDML: A multi-agent language for organizational modelling, *Computational and Mathematical Organization Theory.*

[13] Moss, Scott and Olga Kuznetsova; Modelling the Process of Market Emergence. In Owsinski, J. W. and Nahorski, Z. (eds.), *Modelling and Analysing Economies in Transition*,,(Warsaw: MODEST) pp. 125-138, 1996.

[14] Nelson, R.. and S. Winter 1982, *An Evolutionary Theory of Economic Change* (Cambridge, MA: Harvard University Press).

[15] Newell, A. and H. Simon (1972), *Human Problem Solving*, (Englewood Cliffs: Prentice-Hall).

[16] Penrose, E. (1959), *The Theory of the Growth of the Firm* (Oxford: Basil Blackwell).

[17] Sargent, T. (1973), *Bounded Rationality in Macroeconomics* (Oxford: Clarendon Press).

[18] Sent, E.-M. (1997) , “Sargent versus Simon: Bounded Rationality Unbound”, *Cambridge Journal of Economics*, .21(3), pp.323-338.

[19] Simon, H. (1975), “The functional Equivalent of problem-solving skills”, *Cognitive Psychology*, v. 8, pp. 268-288.

[20] Tambe, M. and P.S. Rosenbloom, (1996) “Architectures for Agents that Track Other Agents in Multi-agent Worlds”, *Intelligent Agents, II, Springer Verlag Lecture Notes in Artificial Intelligence* (LNAI 1037).

[21] Williamson, O.E. (1975), *Markets and Hierarchies* (New York: Macmillan).

[22] Yavlinsky G. (1994) "The Lessons of the Economic Reform" [in Russian], *Oktiabr'*, no.9, pp.136-149.

[23] Ye, M. and K.E. Carley (1995), “Radar Soar: towards and artificial organization composed of intelligent agents”, *Journal of Mathematical Sociology*, v. 20, pp. 219-246.