Emergent Intelligence in Large Scale Multi-Agent Systems

George Rzevski, Petr Skobelev

***Abstract*—** The paper describes a multi-agent system which is capable of achieving its goals under conditions of uncertainty and which exhibits emergent intelligent behaviour such as adaptation, learning and co-evolution with their environment. The intelligence of the scheduler emerges from the horizontal and vertical interaction of its constituent agents balancing their individual and group interests.

***Keywords*—** multi-agent systems, emergent intelligence, real-time planning, adaptive schedulers, transportation logistics.

I. Introduction

THE authors with their development teams have designed and implemented multi-agent systems [1] that exhibit emergent intelligence defined as the ability to find effective solutions to problems under conditions of uncertainty. One of these systems was designed specifically for road transportation logistics [2] (referred to as the Scheduler in the further text) and is described in some detail below.

The concept of emergent intelligence is currently widely discussed [3]. The term "emergent" denotes that intelligence is a property of a group of agents rather than of individual constituent agents. The thesis is that intelligent behaviour emerges from the interaction between agents. The idea is compatible with the notion that human intelligence emerges from the interaction of billions of neurons in our brains [4].

The evidence of the intelligent behaviour of the Scheduler was obtained (a) from Scheduler performance – it solves problems under conditions of uncertainty – and (b) from logs of agent interactions. The logs clearly show that the Scheduler is capable of autonomy, adaptation, learning and evolution - the behaviour usually associated with intelligence - and since none of their components (agents) are capable of such behaviour in isolation, the logical conclusion is that intelligence of the Scheduler is created by the *interaction* of agents.

The most interesting aspects of Scheduler behaviour are: *self-organization*, where local interactions between agents generate global structures which, in turn, affect behaviour of the very agents that built the global structure [5]; *learning*, where agents search for patterns of successful and unsuccessful decisions to improve the process of schedule construction [6] and *spontaneous acceleration* of agent interactions which could be described as autocatalytic chain reaction [7], not dissimilar to those observed in lightning, lasers and even atomic bombs.

The Scheduler exhibits behaviour that perfectly fits the theory of complex adaptive systems with concepts such as order and chaos, link strength, unstable equilibriums, attractors, bifurcations, catastrophes and nonlinearities.

These non-standard aspects of the behaviour of the Scheduler, which drastically differ from the deterministic behaviour of purely algorithmic codes, enable it to generate effectively road transportation schedules under volatile dynamic operational conditions with or without interacting with users.

Manuscript received March 4, 2007: Revised version received May 14, 2007.

G. Rzevski is Professor, Design and Complexity, the Open University, Milton Keynes, UK and Founder and Chief Scientist, Magenta Technology, London, UK (tel.: +44 20 8998 8538, email: george@rzevski.net, www.rzevski.net).

P. Skobelev is Director of Technology, Magenta Development, Samara, Russia (email: skobelev@magenta-technology.com)

II. Complexity of Scheduling

The scheduling requirements of the UK road transportation industry are very complex. The complexity is caused by (a) many possible solutions (a very large solution space), which rules out traditional combinatorial search algorithms; (b) a very high variety of transportation demands and resources, which makes formulation of an optimization function extremely difficult; and (c) uncertainty due to high dynamics and volatility of the operational environment, which makes optimization impractical – a single optimization run is typically an order of magnitude longer than a typical interval between two consecutive changes in operational conditions.

Let us consider the road transportation scheduling complexity in some detail. The scheduler is required to (a) handle transportation instructions (TI) from many different loading points to many different destinations (e.g. customer locations and cross docks, where cargoes are offloaded and consolidated); (b) consider many different routes by which orders can be delivered; (c) allocate cargos of many different sizes and weights to many different types of trucks with or without different types of trailers; (d) take into account preferences of owners, operators and drivers; (e) fit the schedule into numerous constraints imposed by warehouse working hours, driver work rules, safety regulations and enterprise policies, eg, on choosing between own fleet and third-party carriers; (f) take into account that different logistics companies participating in the supply chain have different critical constraints and different criteria for allowing the Scheduler to override certain constraints to achieve a more effective schedule; (g) ensure that the schedule offers opportunities for backhauls and consolidations; and (h) accommodate the frequency and variety of the unpredictable events such as: the arrival of new orders, cancellations, failures, bad weather conditions, road works and no-show of drivers or loading crews.

To enable enterprises to plan and re-plan continuously, reacting to events in real-time, scheduling is divided into "Planning / Commit / eXecute" (PCX) stages and reaches across a multi-day planning horizon.

In the *planning* stage orders are assigned to trucks and truck journeys are constructed. During this stage orders can be added or removed and the route planned for a truck can be changed as a result of subsequent events.

At some point there is a need to *commit* the truck. This will trigger communications to warehouses, driver shift planners, truck servicing etc to make ready the truck for its journey. During this phase changes to the truck schedule are undesirable because there would be knock on effects for the warehouse, driver assignment, etc.

The *execute* stage starts with the driver performing his pre-journey checks and continues until his debriefing at the end of his shift is completed. During this phase a high level of sophistication is needed to alter the truck schedule in transit.

To achieve competitive advantage it is necessary to take into account real-time conditions and makes the allocation decisions taking into account a detailed analysis of the current situation rather than following rigid rules. By calculating the profitability of each order, truck and journey, using a dynamic cost model, it is possible to produce a realistic "the best possible under circumstances" schedule rather than an unrealistic "optimal" schedule. For example, a standard rule-based scheduler would not allow a nearly empty truck to start a journey whilst a truck loaded by only 10% of its capacity may be very profitable if it is loaded with a special cargo.

III. Complexity of the Scheduler

A principle derived from our experience is that the tool used for solving complex problems must be at least as complex as the problem.

The Scheduler has successfully accomplished scheduling of one of the most difficult road transportation networks in the UK, as described in [2], containing 600 transportation locations, 250 trucks and 3 cross-docking locations and involving 4500 orders arriving at irregular intervals. The scheduling was done in real time involving dynamic re-routing and intermediate consolidation of loads. No classical optimization algorithm or constraint-based scheduler [8] could possibly cope with this problem. To the best of our knowledge such schedulers have not been previously proposed in the literature or implemented in practice. A scheduler that can cope with such a variety of operating conditions, handle uncertainty due to the occurrence of unpredictable events and still produce schedules that maximize the specified value (or minimize transportation costs) exhibits a truly intelligent behaviour.

The key elements of the Scheduler are Ontology and Scene Editors and Agent-based Scheduling Engine based on Virtual Market [9].

A. Ontology and Scenes

Ontology contains conceptual knowledge on transportation logistics (eg, trucks, orders, schedules, routes) represented as a semantic network of object classes, relations, attributes and the decision-making logic. Conceptual knowledge on scheduling is separated from the resource allocation mechanism, which greatly simplifies updates and increases the reuse of code. The process of formalizing domain knowledge helps in refining it and closing the gaps left due to the empirical nature of knowledge collection.

Based on problem domain knowledge, agents construct a Scene, which is a model of a specific real life situation of the transportation enterprise. A typical scene is a current resource allocation schedule depicting orders and resources and their attributes, including values, locations, availability etc. (eg, a small section of a scene may be described as «Truck 17» «Moves» «from point A» «To» «point «B»). As events (new orders, cancellations and delays) occur, agents interact to arrive at the decision how to change the current scene (eg, schedule) to accommodate the new event.

Computer-readable representation of domain knowledge and of problem situations enables agents to analyze the current situation and make decisions (eg, "Truck 17 can not be used"); check input data (eg, "point B does not exist in our business network"); and ask users questions.

B. Scheduling Engine

The Scheduling Engine contains all computational resources required to modify the current scene to accommodate the occurrence of an event, such as the arrival of a new order, a failure, a delay or a human error. Key components of the engine are agents and modified contract-net protocols. Magenta agents have multi-dimensional goals, balancing criteria such as Cost, Time, Risk and Service Level. What is the right balance can be decided by the agent or by the system users, as they interact with the system during construction/ reconstruction of a scene. The initial values of agent goals are specified by the user based on the initial situation analysis, taking into account historical data and forecasts. An agent can change its goals in response to an unexpected changing circumstances (eg, failure of a resource), which caused a weak link in the outcome of negotiations. In such cases agents use homeostatic behaviour (first improve the values of the criterion which is the lowest, ie, causes the weak link). Key features of agents in adaptive scheduler is that they can solve conflicts by value/cost trade-offs, which takes into account the amount of cash available in a specific situation before deciding if the increase in value or cost cutting is the priority.

C. Virtual Market

The key principle of multi-agent systems is that each agent pursues its own local goal and that the global goal emerges from the interaction of agents. To tune and speed up the emergence of the global goal there is a need for a carefully design organization and guidance of agent interaction, which is the role of the Virtual Market. The Virtual Market organizing interaction of agents in the Scheduler has several original features, which are described below.

1) Compensation versus Drop-and-Go Method

To achieve the best possible allocation of resources to orders in a volatile environment, agents representing orders are given certain amount of virtual money to enable them to pay for required resources. Charges are imposed on the acquisition of resources with a view to creating free market trading conditions. To speed up re-allocation caused by the occurrence of an unpredictable event, taxes are levied on each transaction, which decreases the number of incremental changes caused by an event. One of the two possible methods is used to regulate re-scheduling in real time.

The Compensation Method [9] is used when there is a need for a thorough analysis of the current situation when an unexpected event occurs. The fundamental principle is that if a new order cannot find a suitable free resource it may make an offer to a previously engaged resource promising to pay a compensation for the annulment of its previous match. Such an offer may trigger a wave of negotiations, including, negotiations for the release of the resource from its previous allocation and the acquisition of a new resource by the abandoned order. The wave of matching and re-matching may extend to several previously agreed allocations, particularly to those that were only partially satisfactory. Virtual money available for the payment of compensations in this chain of negotiations comes from the budgets of those order or resource agents that ask for renegotiation. In exceptional cases where an order comes from a VIP customer, an additional sum of virtual money may be released by the enterprise agent to ensure that the privileged order will be fulfilled, even on expense of the overall enterprise value.

For certain applications there is a need to speedup the agent negotiations and for this purpose the method of compensation is replaced by Drop-and-Go Method, which allows newly arrived orders to grab a resource previously allocated to another order without compensation provided this action will increase the Enterprise Value. In situations characterized by frequent changes, the re-matching of orders and resources after the occurrence of any substantial change affecting the problem domain has considerable advantages over simpler incremental methods where orders are matched to resources on the first-come-first-served basis or via auctions [10, 11, 12].

2) Demand and Resource Pro-Activity

Pro-activity is one of the key conditions for effective teamwork. One can hardly imagine a productive team where everyone is passive and makes no contribution unless specifically asked.

Similarly, agent pro-activity turns to be very important in creating emergent intelligence. For example, when Truck Agents are not satisfied with their assignments, they can pro-actively seek other options by offering their services and proposing discounts to Order Agents. Agents of trucks that are almost fully loaded may recapture the initiative and pro-actively seek those orders, which would make the trucks fully loaded. The same applies to previously allocated orders that are not active for some reason (eg, orders that belong to a group). The selection of agents that will be given pro-activity is done by an intelligent dispatcher, which assesses which agent is in a position to achieve the greatest impact on the scheduling process at a particular point in time. When a resource successfully attracts orders that were previously allocated to other resources, this change initiates a ripple effect of renegotiations, which in turn increases the Enterprise Value.

Pro-activity can also be directed towards the external world. For example the Scheduler can propose to the operator to accelerate or postpone delivery of certain cargoes in order to increase the Enterprise Value. If an order due tomorrow can be profitably delivered today, then this option should be offered to the customer even if he doesn't expect an early delivery. Pro-active interaction with customers (approved by company managers), that takes into account the enterprise interests in the developing situation, is another feature of emergent intelligence of the Scheduler that results directly from characteristics of the applied multi-agent technology.

3) Constructive Destruction

If agents find that one or several parts of the schedule contain weak links, they may initiate the process of destruction and re-building of those parts of the schedule, or the whole schedule. The reconstruction may be triggered by changed goals of agents participating in negotiations, for example if priority given to the minimization of costs is replaced by the need to reduce the risk. To accomplish this task, a new group of agents is formed for a certain period of time. If the re-allocation does not produce an improvement, the previous schedule can be restored. This method is similar to the method of random disturbances used to improve decisions in classic numerical optimizations, but it is far subtler; it is based on the situation analysis and the identification of weak links in the current scene. Analysis is performed by the Enterprise Agent, which continuously monitors agent negotiations, finds out weak links and introduces changes that aim to increase the Enterprise Value during the process of scheduling.

4) The Role of the Enterprise Agent

The Enterprise Agent can offer credit or investments to agents of important clients or scarce resources to improve their position in the virtual market. Through interventions described in this section, the Enterprise Agent, as the representative of the global schedule (which was constructed by interactions of local agents) influences the performance of these same local agents – an important aspect of self-organization.

It is important to note that the Enterprise Agent has no power to order other agents what to do or how to do their jobs; it influences outcomes by adjusting criteria or by triggering agent renegotiation processes, exactly as in modern enterprises, where enlightened executives facilitate rather than instruct.

5) Constraint Stressing

In transportation logistics there are often constraints that can be easily stressed or even rejected, if no other option can be found. Consider an example where no truck is allowed to arrive to the warehouse after 1 pm; if, however, a truck according to the schedule is due to arrive at 1.05 pm and if this is the only option that significantly increases the Enterprise Value, it is worth trying to “stress” this constraint and allow the truck to complete this trip rather than leave the order unallocated.

The decision on constraint stressing may be supported through a review of agent negotiation logs. An agent can be created that is charged to find all rejections given to the Order Agent of this unallocated order, and to sort them by their “closeness” to the acceptance. In this example, 5 minutes may be considered as a relatively small deviation from the rule for the warehouse, and the system may decide to allow constraint stressing autonomously, or to ask the planning operator or warehouse manager for their approval. In this example the agent log serves as another global structure that is temporally created and exists not only to record decisions, but also to find and eliminate weak links in the system. This is a case where the system proposes to the user to review definitions of previous tasks, which were not solved under predetermined constraints.

6) Balancing Interests of all Agents

The schedule quality is considered as a dynamic balance between interests of all independent players in the transportation system under consideration. In transportation logistics such players represent clients, orders, transportation instructions, trucks, journeys, driver shifts, cross-docks, etc. All of them can be characterized not only by constraints but also by goals and preferences and the amount of virtual money, which they are prepared to pay for constraint overriding. Note that goals and preferences may change at the individual level during the process of schedule creation.

The achieved balance may be modified by changing the enterprise strategy in response to changing situations. For example, in some situations it is necessary to transport cargoes quicker and cheaper taking into consideration the level of acceptable risk and individual constraints / preferences of cargo owners. In others, it may be required to transport as much cargoes as possible even if it decreases the enterprise profit in order to deliver the expected service level for a VIP customer.

The balance of interests is not the same as equilibrium. Like with all complex adaptive systems, the Scheduler is never in equilibrium (the state where everything is as it should be and there is no motivation for agents to act). In some cases the balance of interests may be reached only partially, a case when participants in the scheduling process have found an acceptable schedule although some participants are probably still not quite happy with the outcome. The Enterprise Agent or possibly an Operator may intervene in such situations. They can change the weighting between costs, risk, delivery time and service level and thus trigger a new round of local negotiations and a search for new options. The new outcome will have a different “quality” from the business point of view. In fact at any time the emerging schedule can be considered as a network in an unstable equilibrium, which accounts for high adaptability of multi-agent schedulers.

7) Communities of Agents

In many cases the speed and effectiveness of agent negotiations can be improved by clustering orders and resources into groups and assigning an agent to act on behalf of all group members. To underline the fact that agents forming a group are still autonomous these groups are called Communities of Agents. For example, several small orders may not be able to find a place on a big and expensive truck but if consolidated into one big order, they become of interest for carriers and their Community Agent is put in a position to negotiate a truck, which satisfies requirements of all members of the group.

Another situation is when several orders that have already been allocated to a truck find the allocation not quite satisfactory. Partially satisfied orders may elect to be grouped together so that their Community Agent may negotiate their transfer to a smaller truck, a solution that is satisfactory for all members of the group. For illustration, if order 1 needs a transfer of a cargo from A to B, and order 2 – from B to C (and the truck then needs to go back to A), the best option is for these two orders to form a group with order 3 from C to A for a backhaul.

Resources with the same or similar attributes and preferences may decide to form a community and start a search for the allocation options for the whole community. If a satisfactory group allocation is not possible, the Community Agent may ask certain orders to leave the group. Members of the group may be allowed or not to negotiate with their Community Agent although they can always reply to messages. Individual Agents that want to stay in the group may be asked to pay membership fee. Agents who do not approve of work of the Community Agent can demand dismissal of the community or leave the community to start a search for options by themselves. Communities of agents can form associations that represent more complex hierarchical or networking structures and agents can dynamically create new organizations in order to solve complex problems that they fail to solve individually. In every case communities are formed and disbanded autonomously. We consider this particular aspect of agent organization as the most significant contribution to the design of agent interaction.

The formation of communities of agents effectively transforms a flat virtual market into a dynamic multi-level structure in which communities may spontaneously spring into existence and after a while may disappear, depending on prevailing conditions. In addition to horizontal agent-to-agent transactions we have now also vertical transactions between Community Agents and community members, which can be bottom-up, as in the case when an agent decides to leave a community or top-down, as in the case when the Community Agent asks a member to leave the community.

There is a synergy between the concept of agent community and that of a Holon [13]. Communities, at least temporary, become unique and indivisible entities (Holons) with shared interests, attributes and constraints and common behaviour, performance and achievements. The agent acting on behalf of such a community has similar role as any agent in the Virtual Market and will address community members only if required, focusing on external to community interactions.

In principle, communities of agents can be considered as organisms characterized by the goal-driven behaviour (seek missing orders), self-organization (accepting new or expelling existing members to accommodate internally generated requirements for change), protection of boundaries (rejecting unwanted orders or protecting allocations under attack from external agents) and so on. In addition, each community may organize itself differently to suite its particular needs without ever forming traditional command and control hierarchies, preserving instead the freedom for agents to dynamically belong to several communities and to interact horizontally or vertically depending on prevailing needs.

There is an important difference between communities, as implemented in the Scheduler, and coalitions. A Community Agent may, if it is expedient, temporary make decisions on behalf of the community without any consultation with members expecting that corrections may be necessary when the circumstances allow consultations. For example, if the journey agent decides to change shifts and the shift agent to change trucks, agents unhappy with this decision may leave the new journey at any time and thus "correct" previous decision made without general consultation. Decisions without consultations inevitably improve the speed of the scheduling process and often do not invoke corrections, like in the example where a Community Agent decides to place a whole community of orders on an unexpectedly available suitable truck and thus increase the Enterprise Value without needlessly wasting time on prolonged consultations.

It is important to note that the type of negotiations taking place at all levels in the Virtual Market is basically the same, which considerably simplifies the design and coding of the Scheduler (transportation instructions join journey's community in the same way as journeys – driver shift community, or driver shift – truck community).

IV. EMERGENT BEHAVIOR OF THE SCHEDULER

The Scheduler is designed to store the log of all agent activities. The log provide evidence of how a schedule improves in a stepwise manner as agents send to each other tentative proposals, counter proposals, modified proposals and arrive at the final decision in a trial-and-error process. The analysis of this log shows that the Scheduler exhibits the following types of emergent behaviours:

A. Self-Organization

As events that affect the schedule occur, agents react by modifying previously agreed demand-resource matches to meet new requirements. This re-matching represents self-organization. Agents autonomously (without being instructed) act to achieve their goals pursuing a trial-and-error strategy.

B. The Emergence of Order from Chaos

As orders arrive and resources are allocated to orders, the strength of links, which are formed between orders and resources, varies depending on the satisfaction with the match. With time more and more week links get broken and replaced with new stronger links and thus, in time, the order emerges from the initial chaos of disconnected objects.

C. Operation far from Equilibrium

The construction of the schedule initially generates many week links between orders and resources and consequently the schedule is unstable and easily modified. As the process continues and the strength of links increases it becomes more and more difficult to modify the schedule as though the process has locked into an attractor. After some time if there are no new orders the schedule will start degrading because of the outflow of energy (virtual money) due to taxes paid by agents to support links. The tax money can be re-invested in building a new schedule (fully or partially) – to make sure that the final schedule is not in a local optimum caused by a particular sequence of orders.

D. Butterfly Effect

Occasionally the smallest change in external conditions (for example, the arrival of a new small and insignificant order) causes large changes in the schedule. The butterfly effect is controlled by the uneven distribution of virtual money to orders favouring large orders. To predict such points of bifurcations special "virtual orders" can be used which can play role of "sensors" forecasting future dramatic changes of schedule.

E. Oscillations

The same or similar patterns of links between demands and resources dynamically appear and disappear in various parts of the schedule. This process can happen when the schedule is on the edge of two attractors and agents cannot decide which option is better. Special sensors can be introduced to stop or slow down this process when needed.

F. Evolution

As real-time, event-driven scheduling progresses the schedule is being perpetually modified. This is an irreversible process of adaptation to ever changing conditions (the arrivals of new orders, failures, delays and bad weather) and therefore – Evolution. Like in every evolution, there is evidence of the increase in complexity of the schedule as the process continues based on the availability of virtual money or, conversely, the collapse of the schedule caused by the lack of virtual money. The step-wise progress is typical for every evolution known to us, from the evolution of language to the paradigm shifts in the development of science.

G. Pattern Recognition and Learning

One of the advance features of the Scheduler is the ability to recognize regularities, ie, patterns in data. Such patterns represent knowledge hidden in data. As an illustration, knowledge about the effectiveness of scheduling decisions can be obtained from patterns contained in data on past performances. Similarly, knowledge about markets can be obtained from patterns hidden in data on transportation demands and supplies. Any regularity, ie, pattern, in the behaviour of a non-deterministic logistic network, reduces the scheduling solution space and can therefore save a considerable effort in searches.

The pattern recognition method used in the Scheduler is based on patented multi-agent clustering in real time [6], where the term cluster denotes a set of similar records (e.g., volume of deposits and age of bank clients or the amount of money that is monthly drawn from the account). Whilst in traditional data mining algorithms the structure of a cluster is rigidly selected for all clusters in advance (eg, parameters of orders for oil and date of their arrival), in the Scheduler, the structure emerges from the clustering process and different cluster can have different structures. The system thus can discover completely unexpected patterns. This does not prevent the users if they know exactly which patterns they are searching for, to specify their requirements.

The clustering works as follows: an agent is assigned to each record; as soon as created, Record Agents immediately begin to send messages to each other, searching for similar records with a view to forming clusters; when a number of Record Agents agree to form a cluster, a Cluster Agent is created, whose task is to attract further records to the newly created cluster; Record Agents and Cluster Agents continue their negotiations until clustering of all record is complete.

Due to the self-organizing capability of agent swarms, the process of clustering is very flexible and can be performed in real time, in which case whenever a new record arrives the swarm reconsiders previously agreed clusters and decides on the best fit for the new arrival (the process is analogous to that of scheduling in real time, as explained earlier in this paper). Structures of clusters may include sub-clusters (eg, a cluster of orders "delivery to Europe" may contain several sub-clusters of orders for different type of oil and for different weeks in September). Structures of clusters are likely to change with the arrival of new records, if clustering is done in a dynamic data environment.

Discovery of a cluster implies the existence of rules connecting records that belong to the cluster, and these rules can be used as an empirical generalization, eg, orders for certain type of oil come every year from Europe in September. Such rules appear and disappear dynamically and can be used for decision-making only under conditions prevailing during data collection. For example, rules derived from data collected when the logistic environment was distorted due to unforeseen factors such as local armed conflicts at shipping ports or a singular jump in oil prices, cannot be used in situations where these factors are absent. Nevertheless, more often than not these rules are very useful discoveries that can be used as forecasts for marketing and sales purposes as well as generalizations for tentative decision making in scheduling. Typical examples of clusters in transportation logistics are orders grouped by their parameters (original location and destination location, geographical areas, volume of delivery, repeating sequence of orders from different customers, shapes of "good" journeys, truck types, etc.). The following pattern "As a rule, long-distance trips are executed by Third Party Carriers (TPC)" generates the strategy "We immediately plan long-distance orders to be fulfilled by TPC and only when the schedule is nearly completed we check if it is possible to improve it by assigning own transportation resources ". Knowledge about such patterns makes possible to specialize individual agents (eg, small and close-distance Order Agents behaves in a different way then big and long-distance Order Agents). Use of this information in real time allows the users/agents to significantly improve quality and effectiveness of scheduling decisions.

H. Spontaneous Acceleration

Spontaneous acceleration can occur without any apparent cause, akin to autocatalytic processes observed by Prigogine [9]. The acceleration usually leads to the accumulation of energy (virtual money) resulting in a kind of explosion or catastrophe (radical changes in the schedule).

Let us consider how this occurs in some detail. A simplified scene of the Virtual Market is shown in Fig. 1. The scene depicts a community of several trucks, each in turn containing a community of journeys, each of which containing a community of cargos. Each link between demands and resources is labelled with two figures denoting the perceived values of the link by both nodes connected by the link.

We can see that the order 1 is satisfied with the allocation of resources (white colour). However some of the resources, those darker or very dark, are for various reasons less satisfied or not satisfied, respectively (it may be that the return journeys of some trucks are idle and trucks allocated to some cargos are inadequate, etc). However there is no possibility to improve the current situation and scheduling process is slowing to a standstill. Let us assume that the next event is the arrival of the order 2, which provides for less satisfied and not satisfied agents a new opportunity to improve their allocation. Negotiations will immediately begin between cargos of order 2 and a suitable truck, which may agree to accept new cargos and if necessary create new journeys. At the same time there will in parallel start many negotiation processes pro-actively initiated by resources that aim to improve their allocations. This increased activity of agents sensing new opportunities combined with their dissatisfaction with the decision by the truck agent to accept new cargos and create new journeys, may result in a ripple effect of changes to the schedule, accelerating the rate of change and causing a full collapse of the previously agreed schedule and its immediate re-building in a new manner. The schedule thus passes through a slowdown, accelerated activity, collapse into chaos and re-birth.

It is important to note that as the schedule is closer to chaos, it becomes more sensitive to changes and easier to modify, justifying the expression that the performance of complex systems is the most effective "at the edge of chaos".

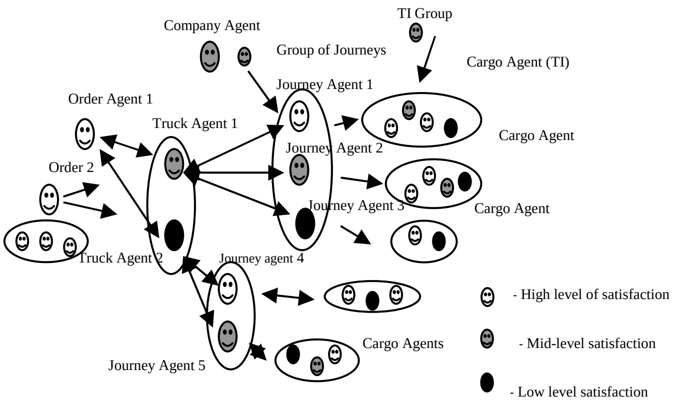

Fig. 1. An example of auto-catalytic reactions in the Scheduler

Other interesting observations include the analogy between the number of messages arriving at an imagined space unit of a scene per unit time and the temperature of that space unit. Indeed, the greater the density of messages and the larger the number of conflicts that require to be resolved per space unit, the higher is the temperature of this spot and, most likely, the longer it will take to resolve all outstanding issues. Considering that virtual money is equivalent to energy, we can talk about "thermodynamics" of the Virtual Market and use thermodynamic methods for identifying problematic regions of large schedules with a view to partition them and process them by different swarms on the same or different servers.

The uncertainty present in the multi-agent system enables it to create emergent behaviour but also causes some real problems for the system designers, which could be summarized as follows.

- The behaviour of the system is unpredictable in detail although the system always arrives at a balanced solution under circumstances
- The reaction of the system to events may vary widely from rapid to slow (when a big reconstruction of the schedule is required)
- It is almost impossible to follow cause-effect chains of ripple effects in the presence of a stream of input events
- The irreversible evolution of the schedule causes problems when attempts are made to roll back
- The dependence of results on time confuses the analysis of system behaviour

Nevertheless the operation of the Scheduler at the edge of chaos is so much more effective under conditions of a volatile and highly dynamic global market in comparison with purely algorithmic and rule-based schedulers that it is worth putting up with certain difficulties.

It is revealing to observe an artificial system, designed primarily to produce rather unexciting albeit complex road transportation schedules, behaving similarly to so many important natural and living systems, including social systems and human mind, in which major breakthroughs are achieved by nonlinear reactions at unpredictable moments of time.

V. Emergent Intelligence of the Scheduler

The system is clearly behaving as a complex adaptive system, a swarm [14], or a team, rather than a computer program. There is no global algorithm to follow (although there are many local ones); there are individual agent goals and guidelines but not step-by-step instructions for the swarm how to achieve the global goals. Each agent pursues its individual goals and as a result of their interaction they collectively achieve global goals – the schedule.

The system is capable of autonomously and rapidly reacting to unpredictable events by re-scheduling parts of the overall schedule that were affected by these events. Reactions to the same event at different times are different, depending on the situation at the time of the occurrence of the event. The system usually finds a feasible way of accommodating an event, provided that the solution space exists.

With hindsight any of these actions can be justified given prevailing conditions but none of them were performed following instructions nor could they have been predicted before they were actually undertaken by the system.

The system autonomously undertakes rather unexpected actions to achieve its goal under conditions of uncertainty created by disruptive events. For example,

- It may find a simple modification that satisfies the new conditions or, to the contrary, it may destroy the previously constructed schedule and rebuild it from scratch
- It may form and disband increasingly complex communities of agents as powerful global structures which can act autonomously and affect the behaviour of agents
- Agents may wait for messages and then respond or they may pro-actively offer their services to other agents
- Agents may compete with each other or co-operate
- A spontaneous acceleration of negotiations may occur in horizontal (agent to agent) and vertical (agent to community agent, etc) interactions, which we consider as a fundamental basis of emergent intelligence.

As a result we have a situation as follows: the system satisfies on of the well-known definition of intelligence, as the capability of achieving its goals under conditions of uncertainty – but yet no single component of the system is intelligent. It is obvious then that the solution to this paradox must be found in component interaction.

As we look at the log of agent interaction we see that the solution to every problem emerges step by step. The proposal of the first pro-active agent is always improved by reactions to this proposal from other agents. The final decision on the allocation of resources to demands is a result of as many as several hundreds conjectures and refutations, to use Karl Popper's terminology [15]. According to Karl Popper, science advances by a trial-and-error process, as follows. A hypothesis is first proposed which is then tested and results of tests are incorporated into the improved hypothesis, which is again tested. The process is repeated until the hypothesis becomes stable, at which point it could be considered as a new theory. Popper described this process as a sequence of "Conjectures and Refutations". The process is similar to Hegel's dialectics but it proceeds in two rather than three steps: "conjectures and refutations" rather than "thesis, antithesis and synthesis". It is no coincidence that our agent swarms arrive at solutions in exactly the same way: by agent proposals improved by counter-proposals in a stepwise manner until the further improvements are not practical or the system runs out of time.

VI. CONCLUSION

Based on observations of a working large scale multi-agent system, the Scheduler, it is reasonable to arrive at the conclusion that a guided interaction of a large number of relatively simple agents produces behaviour, which for all intents and purposes can be defined as intelligent.

The key factor for multi-agent system effectiveness is the organization and guidance of agent interaction. Some uncertainty must be left in the system for it to generate emergent behaviour and yet this uncertainty must not be unbounded. Too much of uncertainty seems to result in unfocussed agent behaviour. We conclude therefore that the critical component of the Scheduler described above is its Virtual Market.

The outstanding research questions are many and include measuring the speed of reactions triggered by events, time required for the schedule to settle down after the ripple, the identification of attractors in the state space of the system and prediction of the conditions under which the system will reach one of the attractors or shift from one to another. The key question is how to guide the interaction of agents, to slow down or accelerate the occurrence of catastrophes and trigger the system to reach a desirable attractor.

The results of these investigations will help significantly to achieve better than humans quality and performance of scheduling not only in transportation and all other logistics applications but also in many other complex domains.

REFERENCES

[1] G. Rzevski, J. Himoff, P. Skobelev: Magenta Technology: A Family of Multi-Agent Intelligent Schedulers. Workshop on Software Agents in Information Systems and Industrial Applications, SAISIA. (February 2006) Fraunhofer IITB

[2] J. Himoff, G. Rzevski, P. Skobelev: Magenta Technology: Multi-Agent Logistics i-Scheduler for Road Transportation – Proceedings of the Fifth International Conference on Autonomous Agents and Multi Agent Systems, AAMAS 2006, - Japan (May 2006)

[3] J. Deguet, L. Magnin, Y. Demazeau: Emergence and Software Development, Based on a Survey of Emergence Definitions, Proc. of Workshop: Emergent Intelligence on Networked Agents, AAMAS 2006, Japan (May 2006) 143-151

[4] Gerald Edelman: Bright Air. Brilliant Fire. On the Matter of the Mind. Allen Lane the Penguin Press, London (1992)

[5] G. Kuppers: Self-organization. The Emergence of Order. From local interactions to global structures http://www.uni-bielefeld.de/iwt/sein/paper no2, PDF (July 1999)

[6] G. Rzevski, I. Minakov, P. Skobelev: Data Mining Patent No: GB 2 411 015 A (17.08.2005)

[7] G. Nicolis, I. Prigogine: Exploring Complexity, W H Freeman (1939)

[8] R. Dechter: Constraint Processing Morgan Kaufmann (2003)

[9] G. Rzevski, P. Skobelev: Agent Method and Computer System For Negotiating in a Virtual Environment, Patent No: WO 03/067432 A1 (14.08.2003)

[10] K. Fisher, J.P. Muller, M. Pischel: Cooperative transportation scheduling, an application domain for DA. Journal of Applied Artificial Intelligence 10 (1) (1996)

[11] Y. Chevaleyre, U. Endriss, S. Estivie, N. Maudet: Welfare engineering in practice: On the variety of multi-agent resource allocation problems. Engineering Societies in the Agents World V, Springer-Verlag (2005) 335 - 347

[12] V. Conitzer, T.W. Sandholm, P. Santi: Combinatorial Auctions with k-wise dependent valuations – Proceedings of 20-th National Conference on Artificial Intelligence (AAAI-2005) AAAI Press, 248 – 254

[13] Dr. Jonathan Thompson: Ant Colony Optimisation. School of Mathematics, Cardiff University, SWORDS (2004)

[14] H. van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, P. Peeters: Reference architecture for holonic manufacturing systems: PROSA – Computers in Industry, 37 (3) (1998) 255 – 274

[15] Karl Popper. Conjectures and Refutations: The Growth of Scientific Knowledge. Rutledge & Kegan Paul Ltd, London (1963)

G. Rzevski is an academic, entrepreneur and consultant. He is Professor Emeritus, Department of Design and Innovation at The Open University, Milton Keynes; Visiting Professor of Multi-Agent Systems at Cologne University of Applied Sciences and Visiting Scholar at Westminster Business School, where he is involved in a number of advanced research projects in the fields of Complexity and Multi-Agent Systems. In 1999 Professor Rzevski founded Magenta Corporation Ltd, London, an international company developing and marketing multi-agent systems for applications such as logistics, e-commerce, text understanding and knowledge discovery. The company employs two hundred programmers and exports software around the world. He is also Chairman of Rzevski Solutions Ltd, London, providing advice and solutions for a wide variety of organizational and Information Technology problems to private companies and administrations.

As a tribute to his successful research career, the Open University established recently a new "The George Rzevski Complexity Laboratory". At The Open University George pioneered undergraduate education in intelligent mechatronics launching a course in which every student experimented with his/her personal intelligent robot. Professor Rzevski is author of more than a hundred scientific publications and co-author of the Magenta Technology patents.

P. Skobelev is Doctor of Science and software technology Director, Magenta Development Ltd, Samara, Russia. From 1983 to1991 he was the chief developer of real-time measurement and control systems, image processing and pattern recognition systems, intelligent simulation systems, expert systems and databases for Samara Branch of Physical Institute of Russian Academy of Science. He has a post-graduate degree in Functional Programming Languages for Artificial Intelligence Applications.
Since Perestroika, starting from scratch, he built several successful software development companies that delivered software in Russia, UK, Germany and USA. In 1991 he met Prof. George Rzevski and they started first collaborative research and design projects in the area of multi-agent systems. As a result of these projects Magenta Corporation was established and funded by UK investors in 2000. He is co-author of the Magenta Technology patents and more than 50 papers on multi-agent systems and applications.