Multi-Agent Real Time Scheduling System for Taxi Companies

Andrey Glaschenko
Magenta Development
3rd Floor, 349 Novo-Sadovaya st., Samara, 443125, Russia.
Tel.: +7 846 342 5177
a.glaschenko@gmail.com

Anton Ivaschenko
Magenta Development
3rd Floor, 349 Novo-Sadovaya st., Samara, 443125, Russia.
Tel.: +7 846 342 5177
anton.ivashenko@gmail.com

George Rzevski
Rzevski Solutions
3 Ashbourne Close
London W5 3EF
Tel.: +44 20 8998 4662
george@rzevski.net

Petr Skobelev
Magenta Development
3rd Floor, 349 Novo-Sadovaya st., Samara, 443125, Russia.
Tel.: +7 846 342 5177
petr.skobelev@gmail.com

ABSTRACT
The paper gives an overview of a multi-agent real-time scheduling solution for taxi companies. It covers problem definition, describes a multi-agent approach to real-time scheduling, outlines the system architecture and gives performance matrices. Key design decisions are also discussed. The most important achievements are the ability of the system to re-schedule taxi service before confirming order acceptance to the client and the system ability to update schedules in intervals between two events, in real time. The multi-agent approach described in this paper is applicable to real-time scheduling and optimization of a wide range of business and social system.

Categories and Subject Descriptors
I.2.11 [**Computing Methodologies**]: Artificial Intelligence – *distributed artificial intelligence*

General Terms
Algorithms, Management, Performance, Design, Experimentation.

Keywords
Multi-agent systems, taxi, dynamic scheduling, events, real time

1. INTRODUCTION
The problem of real-time enterprise management is becoming important and significant in many industries and especially in transportation [1].

During the last 10 – 15 years we have witnessed a considerable growth of various transportation networks, as exemplified by freight services (TNT, Exel, Wincanton, etc.), global courier services (DHL, Fedex, UPS), as well as in the air and sea transportation. As stated in [1], there are two basic reasons for this trend:

- Large fleets are particularly suited for serving large clients because they meet user requirements better and provide higher market benefits
- Intense competition and increased fuel costs have led to bankruptcy of many small and medium companies and their absorption by more successful competitors

Cite as: Multi-Agent Real Time Scheduling System for Taxi Companies, Andrey Glaschenko, Anton Ivaschenko, George Rzevski, Petr Skobelev, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 29 – 36

The market for ERP systems for transportation companies is now mature and ERP solutions are being implemented in most large companies. However, the problem of the real-time management of complex transportation networks has not been satisfactory solved yet. Current solutions (like Descartes [2], Oracle [3] and others) are not effective because they cannot cope with complexity, dynamics and large scale of such networks.

Current solutions rarely consider new technologies like GPS/GLONASS navigation, electronic maps (Map24, MapPoint, GoogleMap, MapInfo and others), the Internet services that provide the up-to-date data on weather and traffic conditions, mobile communication devices such as cell phones with the ability to determine geographic coordinates, communicators, hand held devices and so on. Currently used optimization solutions do not take into consideration many transportation specifics, e.g., identifying the actual location of each vehicle at any time.

As a result, transportation companies either deal with archaic solutions, or delegate the task of transportation scheduling to highly qualified dispatchers, implementing only a limited range of operational requirements. An analysis of several transportation networks shows that a real-time management solution, which is capable of coping with company specifics, could lead to an increase of efficiency of up to 3 – 15%. The economy of scale in modern transportation networks is such that the cost of implementing a real-time management system, such as described in this paper, could be recovered in less than 6 month.

2. Problem definition
Magenta Corporation has successfully developed and implemented an intelligent taxi management system for Addison Lee, which is the largest corporate taxi operator in London.

The characteristics of this taxi service are as follows:

- A modern ERP system and a call centre with about 130 operators receiving orders concurrently; some orders are received through the company website
- A very large fleet of more than 2,000 vehicles (each with a GPS navigation system)
- A very large number of orders: more than 13,000 orders per day; the order flow occasionally exceeds the rate of 1,500 orders per hour; order arrival times and locations are unpredictable
- The order attributes are as follows: place of pick-up and drop; urgent or booked in advance (for a certain date and time); type of service (minivan, VIP, etc.); importance (a

number from 0 to 100 depending the client); special requirements (pet, need for child chair, etc.)

- A large variety of clients, e.g., personal, corporate, VIPs, with a variety of discounted tariffs, with special requirements for drivers, disabled, requiring child seats, requiring transportation of pets, etc.
- A large variety of vehicles, including minivans and cabs, some with special equipment to match special requirements of clients
- A large number of freelance drivers who lease cars from the company and are allowed to start and finish their shifts at times that suite them, which may differ from one day to another
- At any time around 700 drivers are working concurrently, competing with each other for clients
- Vehicles and drivers have the following attributes: type of vehicle; capability to complete special jobs; driver experience (novice or experienced); place where drivers live; current vehicle location (GPS coordinates); driver status ("unavailable", "break", "working", "free", "will be free in 5/10 minutes", "goes home").
- Guaranteed pick up of clients in the centre of London within 15 minutes from the time of placing an order
- Unpredictability of the traffic congestion in various parts of London causing delays and consequently the interruption of schedules
- Unpredictability of times spent in queues at airports and railway stations
- Occasional no-shows of clients and failure of vehicles
- Disruptive events include: the arrival of new orders; changes or cancellations of orders; changes in driver profiles (vehicle type, etc.); changes in driver status or location
- A number of exceptions to the general requirement to find the best economic match between a vehicle and a client, including: matching drivers that drive home after finishing their shifts with passengers travelling in the same direction (to reduce drivers' idle runs) and giving priority to drivers that during a particular day had less work than others (to increase drivers' satisfaction with working conditions); exceptions of this kind may be changed at any time

Scheduling of vehicles and drivers under such conditions represents an exceedingly complex process, which is not feasible to achieve with any known mathematical method. Before the multi-agent scheduler was implemented the company used manual scheduling by a large number of very skilled dispatchers.

3. Existing approaches

In general, this problem falls into the category of a classic assignment problem [4]. However standard solutions of scheduling and optimization (that are usually being applied in transportation logistics and could be considered as a solution for taxi scheduling) have a number of limitations.

1. Theoretical computational complexity of a solution, depending on the chosen algorithm, is between O(n3) and O(n4). Since locations and status of drivers changes every 30 seconds, such dynamics would require perpetual rescheduling. This, combined with the high volume of orders and frequency of disruptive events, makes complexity of taxi management unacceptable.

2. Constraints of taxi management are changing depending on the current situation. For example, to complete an order the driver is usually chosen according to a basic criterion (e.g. distance), but when several drivers are close to the pick-up location, the preference is given to the driver who was idle for a longer time. Similar conditional preferences are taken into account also when a driver is chosen from a queue (a parking place where drivers are waiting, e.g. at the airport).

Because of the limitations of classical methods, heuristic methods [5, 6, 7] are usually used to solve logistic management problems, for example, the technology of savings [8]. Most of the current software packages combine several heuristics to achieve the best result. The following meta-heuristic algorithms ("heuristics" is usually interpreted as a set of rules that determine which option is the best, and "meta-heuristics" determine a strategy to choose the heuristics) have recently become widespread.

- Genetic algorithms
- Ant algorithms
- Tabu search
- Search with adaptive memory

Meta-heuristics achieves better results than classical heuristics and the choice of a concrete meta-heuristics is determined by a problem domain and developers preferences. A detailed quantitative comparison of the effectiveness of different heuristics and meta-heuristics approaches for scheduling is given in [5].

Despite a considerable progress in this area during the last few years it is an imperative to develop new scheduling methods and technology. The basic reason for the research into new algorithms and technologies for scheduling of taxis is complexity of the problem, as exemplified by the following requirements: (a) to consider individual features, preferences and constraints of drivers, vehicles and clients (VIP and ordinary); (b) to re-schedule the service whenever a disruptive event occurs, in real time; (c) to cope with difficult road situation (distant points of loading, traffic jams, and many unpredictable events) and (d) to maintain high service levels (e.g., by guarantying pick-up of clients within 15 minutes from the time of placing the order).

4. Value indication

The use of multi-agent technology for real-time scheduling can definitely improve matching of orders to drivers and vehicles and, in addition, it may lead to a review of the whole service including examining of resources and work practices. For both of these reasons multi-agent scheduling technology is now in high demand.

Examples of taxi scheduling are given in Figures 1 – 3. If a new order has several requirements (see Figure 1), like "booking of a cab in 1 hour with child seat and payment by credit card", the order agent will request from driver agents all feasible options, receive bids, evaluate them and select the most appropriate match.

To assure fair and proportional distribution of jobs for drivers, which is very important for taxi companies to maintain high job satisfaction levels, some of the Driver Agents may receive additional bonuses.

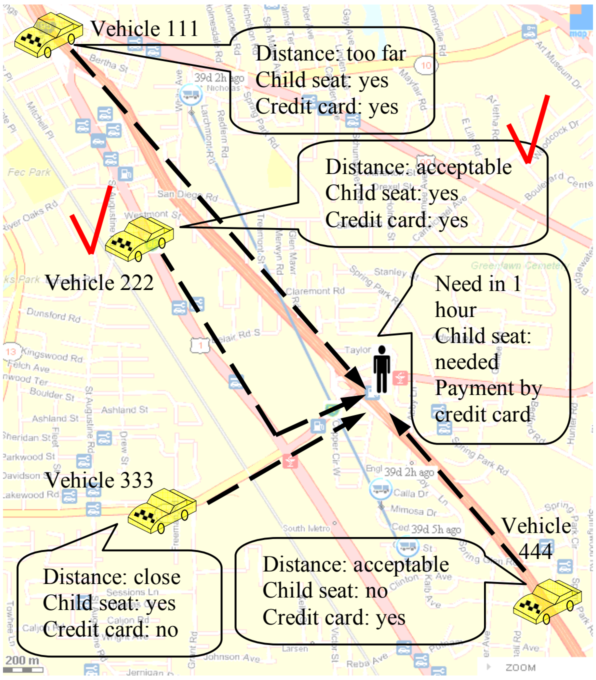

Figure 1. Selecting a vehicle

The second example (see Figure 2) illustrates functioning in real time. When a new event occurs (like traffic jam) the driver can request a "safety net" and, if another driver happens to be free near the Order A (e.g., due to a refusal of a previous order) he may agree to take over the job. This can be also a matter of agent negotiation – the reviewing of options can be transferred from the direct discussions between drivers and dispatchers using radio communication devices, to the multi-agent world inside the system.

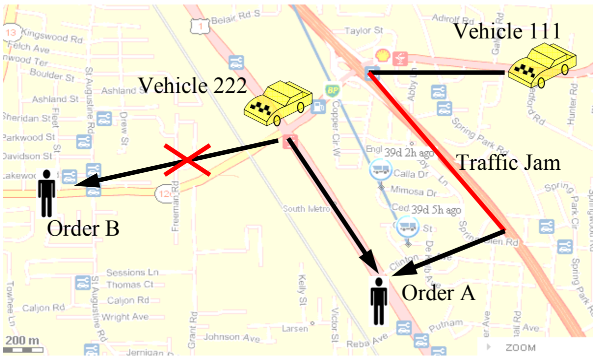

Figure 2. Reacting to events in real time

Agents can use flexible decision-making criteria instead of direct priorities, which is valuable when there is a need to deal with different categories of clients. For example if a VIP order arrives and there is only one cab that completely corresponds to the specified requirements and if that cab is already assigned to another job, the system will nevertheless allocate the VIP order to this vehicle and initiate re-scheduling of the order previously allocated to this vehicle.

Figure 3 illustrates fleet utilization improvement. If a drivers informs the system that he will be soon free (e.g. he is on the way to drop off his passenger, which will take no longer than 15 minutes) the system will take into account this information and schedule the driver for his next job, instead of using another, less suitable option.

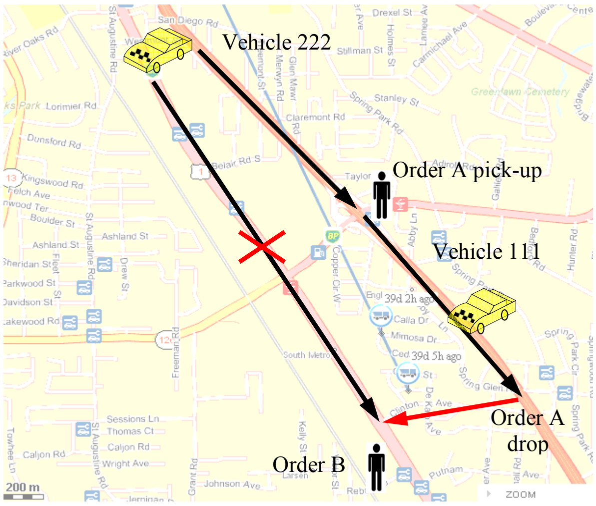

Figure 3. Improving fleet utilization

The number of pick-ups can be increased by scheduling several vehicles to gather in one place (in some cities there can be several such profitable places). If there is a probability that a VIP order may arrive at a distance from this gathering point, the system may recommend to a driver to move closer to the likely order point, offering him in return a guaranteed next order. This functionality can be supported by a pattern recognition algorithm that can forecast the location of the next order using agent-based data mining,

Agent-based scheduling system can work effectively in conjunction with human dispatchers. In a situation where one dispatcher takes new order and schedules a vehicle to come from north to south to pick up a client, and another dispatcher independently schedules another vehicle to go from south to north for another order, the agents can spot this anomaly in the schedule and recommend to dispatchers to change their decisions.

5. Adaptive scheduling approach

In order to improve results described in [9-11] a new approach to scheduling was developed and named "adaptive scheduling". The key feature of the new approach is that when a disruptive event occurs (e.g., arrival of a new order or a change of a driver's status), the whole schedule is not reworked from scratch; only parts of the schedule affected by the event are adapted (modified).

In order to achieve a solution as near as possible to the optimum in a short period of time between two events, the system introduces a delay between the allocation of a vehicle to an order and the point in time at which it sends pick-up instructions to the driver of that vehicle. During this time interval the vehicle is considered to be available for new allocations, but only if, in spite of the required re-scheduling, the re-allocation will lead to the improvements of performance indicators. During this process vehicle agents attempt "to come to an agreement" with each other, which means that they would consider previously allocated orders and give them opportunity to be re-scheduled. The re-scheduling of allocated resources leads to a wave of negotiations aimed at the resolution of conflicts between new and old orders. The length of the re-scheduling chain is limited only by the time required to reach a client in the busy city of London, which normally is sufficient for several changes of the schedule. Thus, the system builds a schedule and perpetually reviews it, as long as there is a possibility to improve key performance indicators, and the decision-making time is still available.

Order agents and resource agents compete with each other or co-operate, depending on requirements. In the case described above it can be construed that an agent is sacrificing its allocation to provide an opportunity for another agent to gain an allocation, and thus improve the performance for the whole service.

Multi-agent technology is in some respect based on object-oriented approach [10], but the key idea of this technology is to provide a mechanism for distributed decision-making, which is essential when dealing with complex problems. Instead of a single, centralized decision-making unit, the technology supports many decision-making nodes that are capable of co-operating, or competing with each other, by exchanging messages or by working on the same data space. Agents differ from objects in the autonomy of their behaviour – each agent has its own goal and an ability to communicate with other agents with a view to achieving a decision consensus.

The distributed paradigm enables complex situations to be represented in the form most natural for users. It also enables software to be partitioned into relatively self-contained parts. It is compatible with the basic trend in modern optimization techniques, most of which are based now on a family of simultaneously competing and/or cooperating optimization algorithms. New approach opens new opportunities for practically unlimited extension of system functionality, e.g., by including a larger number and higher variety of algorithms, and by providing a basis for future development.

Orders and resources (clients and drivers/vehicles) are represented in the system as agents. In addition, some new types of agents are introduced, like external events processor, region loading manager and orders allocation manager. The order agents are the most active: they compile lists of available vehicles and initiate negotiations with vehicle agents. Vehicle agents (driver agents) in the first version of the system are reactive agent: they only reply to requests from order agents and implement the selected option.

System functions in cycles, between which the system collects the events and places them in a queue (see Fig. 4). In each cycle, the events from the queue are processed, one by one, and agents are, in turn, given control by the dispatcher. Each event initiates a chain of negotiations between agents. When all events are processed, the system falls asleep until a new event arrives.

Figure 4 illustrates a situation where two orders (Order 1 and Order 2) are allocated to two vehicles (Resource 1 and Resource 2). When a new order arrives, it triggers into existence the new order agent, which starts negotiations with agents of vehicles that are located within the specified distance. As illustrated, these vehicles include Resource 3 that is free at that time, and Resource 1 that is busy. The new order agent starts negotiations with Resources 1 and 3. Resource 2 is too far away and does not participate in negotiations.

In response to a request from the new order agent, the agents of Resources 1 and 3 will each submit a bid containing an estimate of delivery cost and time. The new order agent will compare bids and choose the best offer according to the most relevant criterion at that time.

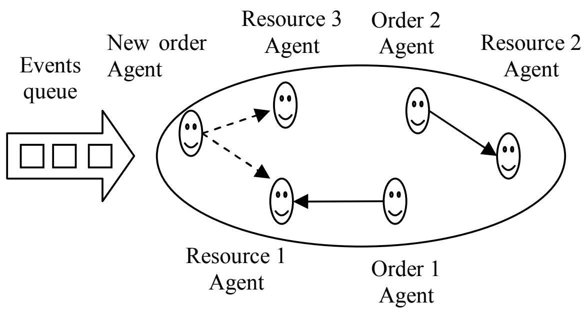

Figure 4. Virtual world of taxis (broken lines show the first message, arrows show matching)

Options are evaluated using the cyclic transfer algorithm [12] with the values of b < 5 (search depth) and k = 1 (number of orders transferred). The experimental results have shown that a search depth of 4 achieves optimal results in 99% of cases. When allocating concurrently incoming 100 orders (which exceeds the number required by problem domain) the algorithm requires about 0.1 seconds. In reality the average search depth of cyclic transfers is approximately 1,5, increasing in rush hours and decreasing to the value of 1 at times when the number of drivers exceeds the number of orders.

Table 1. Scheduling results for different search depth

Search depth	Scheduling time	Evaluation mark
1	–	198.98
2	–	200.28
3	–	198.98
4	0.1 sec	198.87
5	1.3 sec	198.87
6	14.87 sec	198.87
7	532.87 sec	198.87

To decrease the dimensions of the decision space, a pre-matching mechanism is used, which determines the suitability of matching of orders to drivers. This mechanism cuts off unpromising options.

Microeconomics is used to evaluate and compare the order-driver pairs that were created at the previous step. An evaluation mark is given to each option and good options are remembered so that the evaluations do not need to be repeated later. The evaluation mark is determined using multi-criteria model and calculated as a sum of all criteria values multiplied by their weights.

The following criteria are used for option evaluation:

- Distance to the order
- Predicted delay of the pick up, if any

- Preferences of the driver (priority given to the drivers that did not have orders for a long time)
- Drivers experience
- Distance of the driver to overloaded area (to utilize drivers from outlying districts)
- Service level conformity
- Importance and priority of the order
- Driver's place in a queue (if he is waiting at an airport)
- Driver's home address (if he is looking for an order on the way home)

Scheduling workflow includes the following steps:

1. New order comes to the event queue
2. Possibility of order scheduling is checked
3. The order receives an agent
4. All drivers that can complete this order are included in pre-matching (to cut off unpromising options)
5. Evaluation of all driver-order pair is done according to criteria based on microeconomics
6. The order agent requests order completion costs from selected driver agents. This cost includes the cost of transferring the order from the previously allocated driver. The driver receives the information on the reallocation costs by sending a request to its current order
7. If the revised decision is better than the previous one, it is applied
8. Process described in step 6 continues for all candidate drivers, for whom the initial evaluation (without transfers) was better than the current evaluation
9. If no changes occur during the cycle, the event processing is considered finished
10. For each order the decision-making time is determined. This time is dynamically calculated and depends upon the priority and service type of the order and some other parameters. The introduction of the dynamic decision-making time resulted in the increase of the fleet effectiveness by reducing the average task completion time per driver
11. When decision-making time expires, information about the order is sent to a driver and the order is removed from the scheduler.

When new order appears the system automatically finds the best vehicle and preliminary books a resource. On average, it takes 9 minutes to provide a vehicle for an urgent order. Orders with specified time of pick up are immediately allocated but the system continues re-allocating these orders as new suitable resources arrive and does not commit the final decision until the point in time when the driver must start a journey to complete the order. The system can change an allocation several times.

The system first attempts to maximize company profit. Then, other criteria that are important for the business are considered, such as the service level. Also, when choosing from two approximately equal options the system sends the order to that driver who did not receive orders for a long time, thus ensuring relatively fair distribution of orders.

The scheduling method described above underwent certain evolution during the process of implementation.

Firstly, a possibility was introduced to dynamically determine the overloaded areas of the city and to attract taxi vehicles to these areas. The order flow at certain times explodes leaving the affected area without available drivers, the situation that can last for several hours. This is an important feature because productivity of work in overloaded areas determines the actual fleet effectiveness - in areas that are not overloaded there are usually enough drivers to complete available orders.

In the normal mode of scheduling, drivers who are directed to overloaded areas may be intercepted by newly arrived orders. For example, a driver who drives towards the centre of the city, where there are many outstanding orders, and who is still at a point where these orders are out of his visibility area (the maximum distance where orders allocation is allowed), may be allocated another order, which is within his visibility area, even if it is not as urgent as one in the city centre. As a result, some drivers may never get to the problematic area unless the system, in cases of emergency, temporary amends criteria for the allocation. This feature has been implemented and as a result drivers are now congregating in critical locations without interception.

In addition, the scheduler raises a flag in the operational platform according to which drivers are temporally restricted to accepting urgent orders in the critical area even from clients with a low priority. If necessary, the system may temporary extend the area where drivers are allowed to search for orders. As soon as the loading improves, the limitations are annulled.

Secondly, if drivers are allowed to act purely according to personal preferences, a surplus of drivers may accumulate in some regions and insufficient numbers in others. To prevent such situations an option was introduced to distribute the fleet according to the order flow forecasts. Having information about the current order flow and distribution of orders in the past, the order flow may be extrapolated, enabling the system to generate short-term (30 minutes) forecasts, which are normally reasonably correct. Based on the forecast, the system sends to unoccupied drivers text messages with recommendations to stay in the current region or to move to the region where a better order flow is expected. This feature enables an optimal distribution of the fleet, reducing response times and idle miles.

Thirdly, the system tries to identify drivers' attempts to cheat. All interactions of the system with drivers are automated, so from time to time drivers attempt to gain personal advantages by giving deliberately false information. The following cases have been recorded:

- Drivers attempt to reduce waiting time by reporting that they are waiting in an airport queue when, in reality, they may be tens of miles far from the airport
- Drivers attempt to get an earlier order by pressing key "free in 10 minutes" although they may be at the beginning of their assignment
- Drivers attempt to receive orders in a particular direction several times during a day by pressing key "goes home" several times

To reduce cheating driver agents monitor driver schedules and, when appropriate, ignore their requests.

During the system implementation other changes aimed at improving the effectiveness of the scheduling process were introduced, e.g., a mechanism for the allocation of orders, which are preliminary booked by drivers, to be completed on the way from home on the next day.

The technology used in this project enables the introduction of improvements, such as those described above, during the system operation, thus ensuring a perpetual evolution and performance enhancement. The independence of constituent algorithms facilitates step-wise development without causing undue increase in system complexity.

The multi-agent technology enables the application of all facilities of object-oriented design to schedule optimization in cases where it is difficult, or impossible, to use more formalized algorithms such as ant colonies or genetic algorithms. The proposed approach increases the productivity of design, development, support and evolution of the system and decreases overall costs and time required for the implementation of real-time scheduling software.

6. Application architecture and UI

Scheduling tool is a "virtual client" for the company operational platform. It functions on a client computer getting all information required for decision making and for autonomous allocation of resources through an application server.

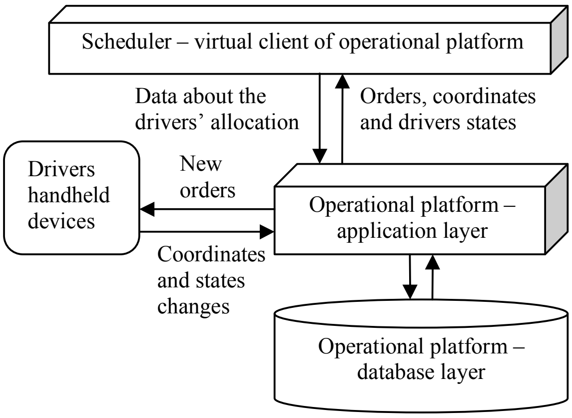

Figure 5. Interaction of scheduler with operational platform

The system functions 24 hours a day and consists of several modules interconnected by the event queue. It works in several threads and changes system configuration when required. For example, for testing purposes, the system may use loaders of XML scenarios or historical data. By temporally replacing "Time Provider" module, the system operates in real or model time, or changes the current time step by step, which helps the analysis of scheduling results.

An example of a main system screen is given in Figure 7 that shows orders coming from the call centre and website, vehicles allocation results, etc. For a selected order, the Map can be opened to shown the route of the vehicle and its current location (see Figure 8).

7. Results

The system began its operation and maintenance phase in March 2008, only 6 months from the beginning of the project.

Results were extremely good: 98.5 % of all orders were allocated automatically without dispatcher's assistance; the number of lost orders was reduced to 3.5 (by up to 2 %); the number of vehicles idle runs was reduced by 22.5 %. Each vehicle was able to complete two additional orders per week spending the same time and consuming the same amount of fuel, which increased the yield of each vehicle by 5 – 7 %.

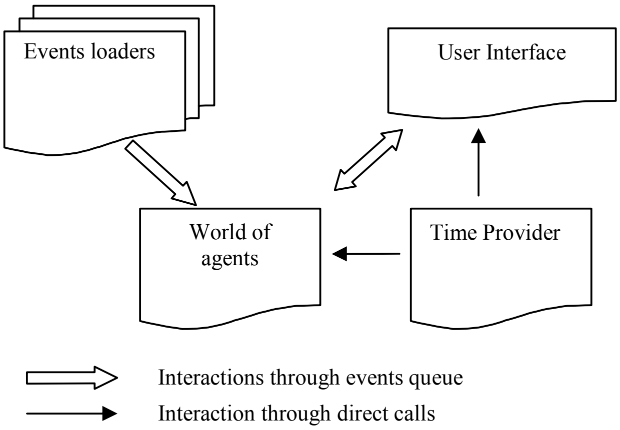

Figure 6. Scheduler architecture

Time required to repay investments was 2 months from the beginning of the operational and maintenance phase. During the first month of operation the fleet utilization effectiveness was increased by 5 – 7 %, which, in absolute numbers, gives the same fleet an opportunity to complete additional orders bringing 5 million dollars per year. The additional income is being distributed between company and drivers. According to available statistics, since 2008 driver wages have increased by 9 %, and there is a possibility for an overall fleet growth.

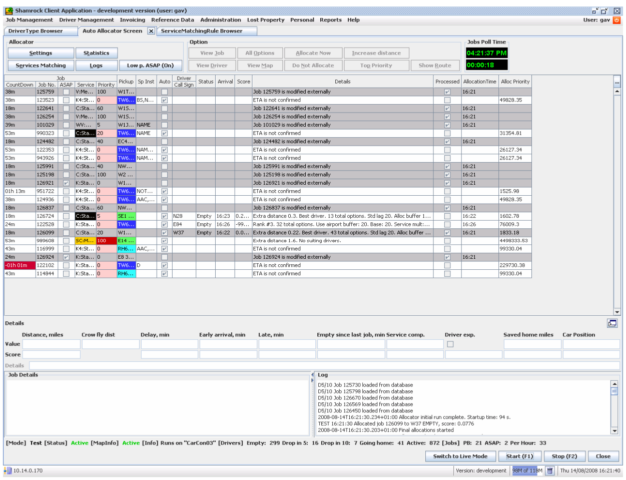

Figure 7. Main scheduling screen

Further improvements include a reduction of delayed pick-ups by 3 times, which brought a considerable improvement of service levels for the customers. Urgent order average response time (from booking till arrival for pick up) is now 9 minutes, which is the best time in London. For high priority orders the response time is not greater than 5 – 7 minutes. A reduction in response time is especially noticeable in overloaded areas.

Search for orders on the way home and improved optimization mechanism, when compared with a previous system, gives 3 – 4 thousand miles reduction in daily fleet run, which is of benefit to both drivers and city ecology.

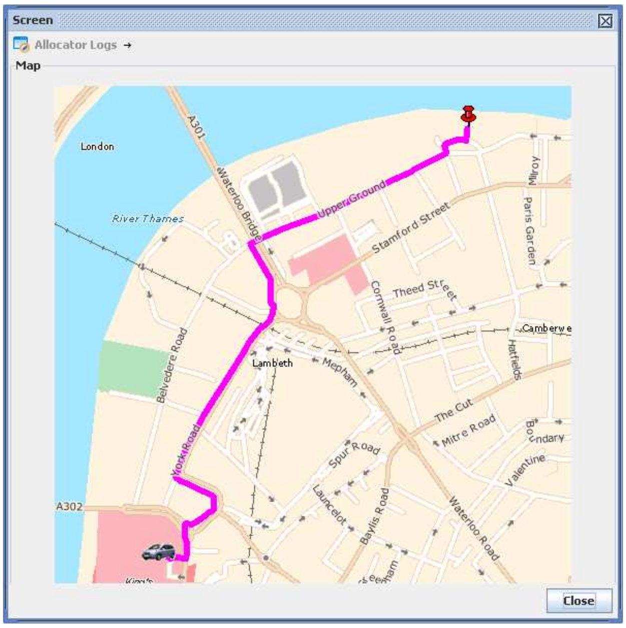

Figure 8. Using map (Map Info) for routing

Further developments will include an analysis of vehicle movements to determine actual speed of vehicles, a partial optimization of the business taking into account courier deliveries (the basic advantage of courier service is having several orders per one courier) as well as other changes targeting the improvements of business effectiveness.

The Addison Lee scheduling project was referenced by a Russian TV news programme (the trailer can be downloaded from [13]) and, as a result, the taxi company became a finalist of The Orange Best Use of Technology in Business Award 2008 [14].

8. CONCLUSION

A very complex problem of managing the largest taxi service in London was solved using a new multi-agent adaptive scheduling method. A considerable economy of resources has been achieved for the client.

The method is innovative, scalable, compatible with modern trends in optimization and, above all, proven to be effective in a large-scale, practical, commercial application. The use of distributed decision-making and the employment of a variety of interacting self-contained algorithms ensures its applicability to a variety of business problems.

9. REFERENCES

[1] CapGemini Transportation Report 2007 DOI= http://www.capgemini.com/resources/thought_leadership/transportation-report-2007.

[2] Descartes DOI= http://www.descartes.com/solutions/dms/daily_planning.html

[3] Oracle DOI= http://www.oracle.com/applications/scm/transportation-management/index.html.

[4] Shuhman A.E., Shuhman E.V. 2006 Empiric analysis of methods to solve the assignment problem. Materials of scientific conference "Challenges of XII century and education". Orenburg, 2006, 227-230.

[5] The VRP Web DOI= http://neo.lcc.uma.es/radi-aeb/WebVRP.

[6] Stefan Vos. 2000 Meta-heuristics: The state of the Art. Local Search for Planning and Scheduling. Edited by A. Nareyek, ECAI 2000 Workshop (Springer-Verlag, Germany, August 21, 2000)

[7] Handbook of Scheduling: Algorithms, Models and Performance Analysis. Edited by J. Y-T. Leung, Chapman & Hall, CRC Computer and Information Science Series, 2004.

[8] Clarke G., Wright J. 1964 Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12 # 4, 568-581.

[9] Skobelev P.O. 2002 Open multi-agent systems for decision making support. Avtometriya, 2002, # 6, 45-61.

[10] Vittikh V.A., Skobelev P.O. 2003 The multi-agent models of interaction in demand-resource networks. Automatica and Telemechanica, 2003, # 1, 177-185.

[11] Andreev V.V., Vittikh V.A., Batischev S.V., Ivkushkin K.V., Minakov I.A., Rzevski G.A., Safronov A.V., Skobelev P.O. 2003 Methods and tools of developing multi-agent systems for decisions making support, News of the Russian Academy of Sciences. Theory and systems of control, 2003, # 1, 126-137.

[12] Thompson P. M., Psaraftis H. N. 1993 Cyclic Transfer Algorithms for the Multivehicle Routing and Scheduling Problems, Operations Research 41, 935-946.

[13] Magenta Technology, DOI= www.magenta-technology.com.

[14] The Orange Best Use of Technology in Business Award 2008 DOI= http://www.nationalbusinessawards.co.uk/2008finalists/index.cfm?ccs=651.

This page is intentionally left blank.