

Complex Adaptive Taxi Service Scheduler

Complex Adaptive Taxi Service Scheduler

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

Lead Author: *Rick Adcock*

This article is based around a **London Taxi Service Case Study** (Rzevski and Skobelev, 2014). The case study focuses on the development of a Real-Time Complex Adaptive Scheduler for a London Taxi Service capable of managing the complexity of many hundreds of taxi journeys in an unpredictable and changing environment, while fitting into the goals and values of the Enterprise.

□

Contents

Background

Purpose

Challenges

Systems Engineering Practices

 Overview

 Knowledge Base

 Virtual World

 Connecting Virtual and Real Worlds

Lessons Learned

References

 Works Cited

 Primary References

 Additional References

Background

When this project was initiated, the company, the largest and the best-known minicab (taxi) operator in London had a fleet of more than 2,000 vehicles, each with a Global Positioning System (GPS) navigation system. The fleet comprised a variety of vehicles, including minivans and Sport Utility Vehicles (SUVs), some with equipment to match special customer requirements. Under usual circumstances, approximately 700 drivers worked concurrently, competing with each other for customers.

The company had a modern Enterprise Resource Planning (ERP) system and a call center with over 100 operators receiving orders concurrently. Some orders were received through the company website. A large team of skilled dispatchers allocated vehicles to customers.

Main characteristics of the taxi service were as follows:

- More than 13,000 orders per day
- Occasionally more than 1,500 orders per hour (1 order every 2.4 seconds)
- Unpredictable order arrival times and locations
- Various clients, e.g., personal, corporate, Very Important Persons (VIPs), a variety of discounted tariffs, special requirements suitable for the disabled, small children (child seats), transportation of pets, etc.
- Many freelance drivers who leased cars from the company and were allowed to start and finish their shifts at times that suited them, which may have differed from day to day
- Clients in central London were guaranteed pick up times within 15 minutes of order placement
- Fundamentally, the company tried to find the best economic match of vehicle to every client. However, dynamic exceptions to this basic requirement included:
 - Matching drivers going to and from home with passengers travelling in the same direction (to reduce drivers' idle runs); and
 - Giving priority to drivers with less work during a particular day (to increase drivers' satisfaction with working conditions)

No pre-planned taxi schedule was viable because any of

the following unpredictable “Disruptive Events” occurred every 2 to 10 seconds:

- Order arrival, change, or cancellation
- Changes in driver profile, status, or location
- Client no-show
- Vehicle failure
- Delays due to traffic congestion, or queues at airports, railway stations, etc.

Purpose

Rescheduling up to 700 independent entities travelling in London under unpredictable conditions that change every few seconds represented an exceedingly complex task, which was not feasible to accomplish using any known mathematical method.

Manual scheduling, as practiced, could not handle the frequent disruptive events. Many perturbations, such as unexpected delays, had to be ignored by the human dispatchers.

Therefore, the project’s objective was to provide effective, real-time, automated assistance to accommodate the disruptions that drove the scheduling. Thus, the project purpose became the development of a complex adaptive software system capable of managing the taxi operation complexity described above with the aim of substantially improving: (1) operational profitability; (2) customer service quality; and (3) driver working conditions.

The planned transformation was from a manual to semi-automated managed taxi operation that facilitated optional human dispatcher interactions with a complex adaptive system scheduler. A thorough analysis of contemporary practices showed that such a transformation has never been achieved before. To the best of the project team’s knowledge, there were no real-time schedulers of taxi operations in existence anywhere in the world.

Challenges

The team undertaking the development of a new real-time scheduler for this client had vast experience of designing and implementing complex adaptive software, and therefore no particular challenges were anticipated. The multi-agent technology, which underpinned the

system, was well understood by the team, and a methodology for managing complexity (Rzevski and Skobelev, 2014) of the task was in place.

Systems Engineering Practices

Overview

The complexity of the taxi service ruled out all conventional systems engineering practices. The real-time adaptive scheduler for the client's taxi service was developed using multi-agent software technology. STOPPED HERE The scheduler design consisted of the following major components (Rzevski and Skobelev, 2014):

1. A Knowledge Base containing domain-useful information relevant to the client's taxi service
2. A Multi-agent Virtual World which models the Real World of the taxi service and is capable of managing its complexity
3. Communication channels between the Virtual and Real Worlds which enable the Virtual World management of the Real World with or without human intervention.

The system was designed to behave as follows. In reaction to every disruptive event, Order Agents, assigned to every received order, and Driver Agents, assigned to every working driver, negotiate the most suitable Order-Driver match through the exchange of messages. Before starting negotiations, these software agents consult the Knowledge Base for the current negotiation rules. Once the best possible match (under prevailing circumstances) is agreed upon, the result is communicated to Drivers, who are free to accept or reject the task (Glaschenko et al. 2009). This process is depicted simply in the figure below.

After a successful prototype implementation, a basic version of the complex adaptive scheduler was developed as described below.

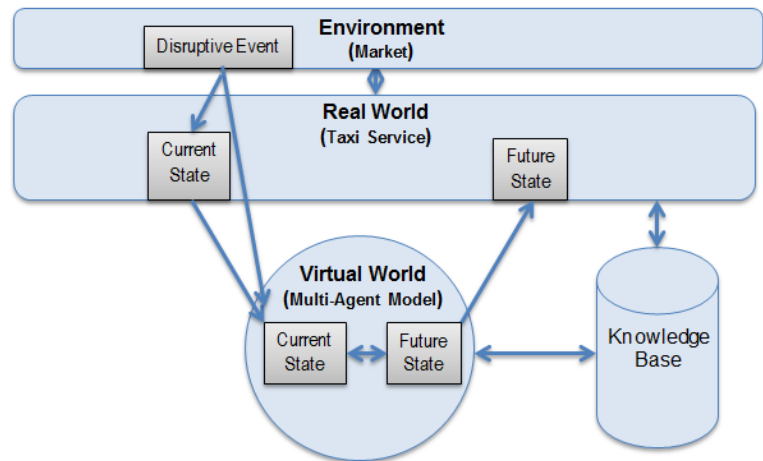


Figure 1. Essence of the event driven, real-time, adaptive scheduling of a taxi service. This material is reproduced with permission of G. Rzevski and publisher WitPress. All other rights are reserved by the copyright owner.

Knowledge Base

The Knowledge Base consisted of: (1) Ontology, containing conceptual knowledge as a semantic network; and (2) Values, in standard databases.

The basic Ontology contained two Object Classes: Order and Driver. Order attributes were:

- Location of pick-up and drop-off
- Pick-up(urgent or booked in advance for a certain date and time)
- Type of service (standard car, minivan, VIP, etc.)
- Importance of service (a number from 0 to 100 depending upon the client)
- Special requirements (pet, child chair, etc.)

Driver attributes were:

- Type of vehicle
- Capability to complete special jobs
- Driver experience (novice or experienced)
- Domicile of driver
- Current vehicle location (GPS coordinates)
- Driver status (unavailable, break, working, free, will be free in 5/10 minutes, home transit)

Factual data on Object Instances (Individual Orders and Drivers/Vehicles statuses) were stored in client's databases, including Scenes (i.e., instantaneous models

of the taxi service yielding every vehicle location and driver availability).

Virtual World

In the basic version of the scheduler, the allocation of taxis to customers was done by the negotiation between Order Agents, assigned to customers, and Driver Agents, assigned to taxi drivers. Order Agents were active: they compiled lists of available vehicles and initiated negotiations with Driver Agents. In this first version of the system, Driver Agents were designed to be only reactive: they only replied to requests from Order Agents and implemented the option selected by an Order Agent.

In the extended version, hereafter described, Order Agents and Driver Agents competed with each other or co-operated, depending on what was best for the whole enterprise. In addition to Order and Driver Agents, this version used some new types of agents, namely: External Events Agents, Regional Loading Agents, and Orders Allocation Agents.

Agents were designed to 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 arrived and there was only one driver that fully corresponded to the specified requirements and if that driver was already assigned to another job, the system would nevertheless allocate the VIP order to this driver and initiate re-scheduling of the previously agreed matches, if required.

The system first attempted to maximize company profit. Then, other criteria that are important for the business were considered, such as the service level and driver working conditions. For example, when choosing from two approximately equal options the system allocated the order to the driver who had not received orders for a longer time, thus ensuring relatively fair distribution of orders.

This virtual agent-based scheduling system was designed to work effectively with human dispatchers. In a situation where one dispatcher takes a 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 virtual agents can spot this schedule anomaly and recommend dispatchers change their decisions to be more effective.

To enable improved performance, the taxi allocation system functioned in short cycles rather than as an immediate reaction to every event. Between the cycles, the system collected the events and placed them in a queue. During each cycle, the events from the queue were processed one by one and appropriate agents, in turn, were given control by a designated human system dispatcher. Each event thus initiated a chain of negotiations among virtual agents. When all events were processed and the system dispatcher was satisfied that the best possible schedule was produced for that cycle, the schedule perturbation was implemented in the real world, and the system fell asleep (was idled) until a new event arrived causing the initiation of the next cycle.

To decrease the dimensions of the decision space, a pre-matching mechanism was used, which determined the suitability of Order-Driver matching. This mechanism cuts off unpromising options.

The Order-Driver pairs were evaluated before the final decision was made. An evaluation mark was given to each option and good options were remembered so that the evaluations did not need to be repeated later. The evaluation mark was determined using a multi-criteria model and calculated as a sum of all criteria values multiplied by their (variable) weights.

The following criteria were used for option evaluation: distance to the order, predicted delay of the pick-up, if any, preferences of the driver, driver 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 to or from home).

Scheduling workflow included the following steps:

1. New order arrives and joins the event queue
2. Possibility of order scheduling is checked
3. A software agent is assigned to the order
4. All drivers that can complete this order are included in pre-matching
5. Evaluation of all Order-Driver pairs is done according to agreed criteria
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, if any

7. The Driver Agent receives the information on the reallocation costs by sending a request to its current Order Agent
8. If the revised decision is better than the previous one, it is applied
9. Step 6 continues for all candidate drivers, for whom the initial evaluation (without transfers) was better than the current evaluation
10. If no further changes occur during the cycle, the event processing is considered finished

In order to achieve the best possible solution, the system continued to search for improvements in previously agreed Order-Driver matches until the last moment when it had to issue the instruction to a driver to fulfil an order (commitment time). During this time interval, the Driver was considered to be available for new allocations but only if the new allocation improved specified performance indicators.

When required, Driver Agents attempted “to come to an agreement” with each other about proposed re-allocation of orders. Occasionally, compensation was offered to the Driver Agent who lost a good client in order to improve overall value of the business, and Driver Agent satisfaction, in particular. Very often the re-scheduling of allocated resources caused a wave of negotiations aimed at the resolution of conflicts between new and old orders. The length of the re-scheduling chain was limited only by the time required for a taxi to reach a customer in a busy city such as London, which normally was sufficient for several changes of the schedule.

To summarize, the system built a schedule and perpetually reviewed it, attempting to improve key performance indicators as long as the time for essential re-scheduling was still available.

The Commitment Time was dynamically calculated for each order, taking into account the priority and service type of the order and some other parameters. The introduction of the dynamic Commitment Time resulted in the increase of the fleet effectiveness by reducing the average task completion time per driver.

An option was introduced for the system 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 expected order-flow was extrapolated, enabling the system to generate short-term (30 minutes) forecasts, which were normally

reasonably correct. Based on the forecast, the system sent text messages to unoccupied drivers with recommendations to stay in, or move to, the region where an increased order flow was expected. This feature enabled an improved distribution of the fleet, reducing response times and idle miles and increasing the number of pick-ups.

In cases when forecasts envisaged a probability of a VIP order arrival at a significant distance from the point where drivers were advised to congregate, the system would recommend that a proximate driver to move closer to the likely order point, offering him/her a guaranteed next order in exchange for compliance. This was an important feature because there were usually enough proximate drivers to complete available orders in areas that were not overloaded, and productivity of work in overloaded areas determined the actual fleet effectiveness. The system was also designed with an option to temporarily amend criteria for the allocation of orders to drivers (for example, to extend the area where drivers are allowed to search for orders) to enable drivers to reach critical locations without being intercepted by less important orders from nearby locations.

The forecasting functionality was supported by an agent-based dynamic data mining system, which was, in fact, another complex adaptive system cooperating with the complex adaptive scheduler.

In later versions the system was designed to detect and identify drivers that cheat, i.e., deliberately providing the scheduler with false information to gain personal advantages. Recorded cases include attempts to:

- Reduce their ultimate waiting time by reporting that they were already waiting in an airport queue when, in reality, they may still have been tens of miles from the airport
- Get an earlier next order by indicating “free in 10 minutes” at or near the beginning of a long assignment
- Receive orders in their home direction by indicating “going home” several times during a day.

To reduce cheating, Driver Agents were designed to monitor drivers’ schedules and ignore their messages, when judged inappropriate.

The final version of the complex adaptive taxi service

scheduler negotiated only with agents that were affected by a disruptive event and then modified only affected parts of the schedule. This capability was a key feature that improved overall effectiveness.

Connecting Virtual and Real Worlds

The Virtual World, which is a model of reality and resides in the scheduler, is connected with the Real World of customers, dispatchers, and drivers, as follows.

As customers ring the call center or visit the website to place, modify or cancel an order, dispatchers enter the pertinent information into the system. Drivers communicate with the system using GPS, mobile phones, or specialized handheld devices, conveying information on their location, direction of travel, availability, etc., and, in turn, receive instructions to pick up customers.

Lessons Learned

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 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 %.

Time required to repay investments was 2 months from the beginning of the operation and maintenance phase. During the first month of operation the fleet utilization effectiveness was increased by 5 - 7 %, which represents potential additional revenue of up to 5 million dollars per year. Such realized additional income has benefited both the company and the taxi drivers. According to available statistics, driver wages have increased by 9 % since 2008, and there is a possibility for an overall fleet growth.

Delayed pick-ups were reduced by a factor of 3, which considerably improved customer service. Urgent order average response time (from booking until taxi pick-up arrival) decreased to 9 minutes, which is the best time among all taxi services in London. For high priority orders, the response time is 5 - 7 minutes or less.

Response time reductions are especially noticeable in overloaded areas.

Implementation of “on the way home” orders, an improved allocation mechanism, when compared with a previous system, gives 3 - 4 thousand miles reduction in daily fleet run, greatly benefiting both drivers and the city’s ecology.

Further developments targeting business effectiveness improvements may include an analysis of vehicle movements to determine actual vehicle velocities that could improve courier service by increasing the number of orders per courier.

References

Works Cited

Rzevski, G., Skobelev, P. 2014. *Managing Complexity*. WIT Press, New Forest, Boston. ISBN 978-1-84564-936-4.

Glaschenko, A., Ivaschenko, A., Rzevski, G., Skobelev, P. “Multi-Agent Real Time Scheduling System for Taxi Companies”. 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. ISBN: 978-0-9817381-6-1, pp. 29-35.

Primary References

Rzevski, G., Skobelev, P. 2014. *Managing Complexity*. WIT Press, New Forest, Boston. ISBN 978-1-84564-936-4.

Additional References

None.

< Previous Article | Parent Article | Next Article >

SEBoK v. 2.5, released 15 October 2021

Retrieved from

"https://www.sebokwiki.org/w/index.php?title=Complex_Adaptive_Taxi_Service_Scheduler&oldid=62416"

This page was last edited on 11 October 2021, at 08:01.