Optimizing Limousine Service with AI

Andy Hon Wai Chun
Chief Information Officer
City University of Hong Kong
Tat Chee Avenue, Kowloon Tong
Hong Kong SAR
andy.chun@cityu.edu.hk

Abstract

A common problem faced by expanding companies is the lack of skilled and experienced domain experts, especially planners and controllers. This can seriously slow down or impede growth. This paper describes how we worked with one of the largest travel agencies in Hong Kong to alleviate this problem by using AI to support decision-making and problem-solving so that their planners/controllers can be more productive in sustaining business growth while providing quality service. This paper describes a Web-based mission critical Fleet Management System (FMS) that supports the scheduling and management of a fleet of luxury limousines. Clientele is mainly business travelers. The use of AI allowed our client to increase their business volume and expand fleet size with the same team of planners/controllers while maintaining service quality. This paper also describes our experience in building modern AI systems leveraging on Web 2.0 open-source tools and libraries. Although we used a proven AI model and search algorithm, we believe our innovation is in striking the right balance and combination of AI with modern Web 2.0 techniques to achieve low-risk implementation and deployment success as well as concrete and measurable business benefits.

Introduction

The Fleet Management System (FMS) was created for the Airport Limousine Services Ltd. (ALS), a subsidiary of Swire Travel (Swire Travel 2010, ALS 2010). Established in 1948, Swire Travel one of the largest travel management companies in Hong Kong and widely recognized as the leader in service quality in the local travel industry. It was the first agent to be appointed to the Hong Kong Travel Industry Council (TIC) and has been registered as an IATA-approved travel agent for over 50 years (TIC 2010). Swire Travel is part of the worldwide Swire group, which has over 145 years of history (Swire Group 2010).

ALS operates a 24-hour luxury limousine and shuttle service. Their clientele is mainly from the business community.

Problem Domain

ALS operates a fleet of Mercedes Benz sedans and stretched limousines as well as multi-purpose vehicles (MPVs) and shuttle buses. ALS provides two main types of services – airport transfer and limousine on-hire services. The airport transfer service provides transfers to and/or from the airport with possible multiple stopovers or pickup points in between. For airport transfers, the company also provides a personalized meet-and-greet service and assists with VIP check-ins if needed. The limousine on-hire service, on the other hand, provides a vehicle and chauffeur to support any type of transportation needs, such as business appointments, plant site visits, hospitality for overseas corporate clients, and even weddings. Trips can be “cross border” to the Mainland of China using vehicles with dual Mainland China and Hong Kong license plates. Since their clientele consists mainly of business users, quality of service, especially in precise timing for pick-up/drop-off, is very important.

Business Needs

ALS is a young and fast-growing company with aggressive expansion plans. However, one of the major bottlenecks in expansion capability is the availability of skilled planners and controllers. Besides long and extensive training, the job can be hectic and stressful at times. Planners/ controllers need to take in information from many different channels, constantly communicate with limousine chauffeurs to record statuses and dispatch orders, and continuously perform scheduling/re-scheduling decisions while trying to balance various business/operational criteria to maximize profit and productivity while maintaining service quality. A fair amount of problem solving is needed as traffic can be congested and leading to delays, clients might be late, flights might be delayed, etc. ALS found it hard to train fast enough an adequate number of high quality planners/ controllers who are up to these challenges, to meet ALS’s aggressive growth plans. Hence they approach the University to see if AI technology can help solve some of these problems.

Using technology to streamline operations is not new to ALS. ALS has already been using many different technologies to reduce planner/controller workload so that they can handle more orders. Currently they have a Web-based online reservation system that is used both by their business clients as well as Swire Travel's hundreds of travel agents. Hotel orders, which are simpler, are recorded separately. A GPS system monitors the location of all vehicles on a map. A separate system retrieves airport flight statuses, arrival/departure times as well as arrival hall information so that limousine chauffeurs know exactly where to meet up with their clients for airport pick-up. Unfortunately, these systems were not integrated and the planners/controllers had to operate several computers and view different monitors to get a complete picture.

AI Project Objectives

The AI project objective for the Fleet Management System (FMS) is simple: use AI to help planners/controllers handle more orders and manage more vehicles while maintaining high service quality.

This high-level objective was then broken down into several finer and more concrete objectives. Firstly, in order for any AI system to work, access to or integration with all necessary data and information sources was needed. The first objective of the Fleet Management System (FMS) was to provide a set of screens to consolidate all relevant data/information into one place. This included creating software to access order information from clients, travel agents and hotels; flight information, statuses and arrival hall locations; limousine/chauffeur data; vehicle locations; and passenger statuses. Once all the data integration was done, this version of the FMS was released to the planners/controllers as a decision-making support system for manual operations.

The second objective of the system is to help planners/controllers easily visualize the state of their world in one simple screen. Using data from different sources, we were able to create a Web-based Gantt chart to help planners/controllers visualize vehicle utilization/availability over time. This bird's eye view allowed planners/controllers to make more informed and better decisions.

The third objective of the AI system is to be able to automate the vehicle dispatching and re-scheduling function while maintaining high service quality. This is done through constraints, heuristics and a search algorithm. One of the original concerns is the AI will not have the same experience as human planners/controllers when it comes to estimating travel times. Accuracy in determining travel times is tightly linked to service quality. This is another key objective of the system.

Application Description

The Fleet Management System (FMS) is currently used by ALS planners/controllers as well as ALS travel agents. Planners/controllers use the system to perform vehicle dispatching. Travel agents use FMS to view vehicle details and assignment statuses. In the future, ALS plans to extend system use to chauffeurs to that they can directly update statuses and view assignment details, such as client addresses, phone numbers, etc. Figure 1 is a high-level use case diagram for FMS.

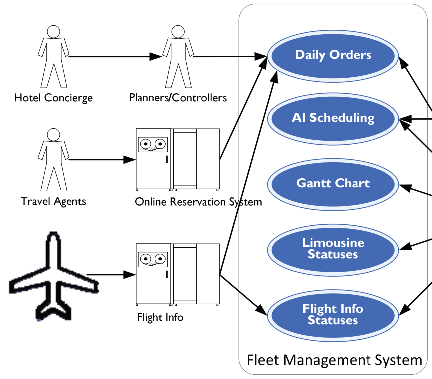


Figure 1. High-level Use case Diagram for FMS

The daily workflow is usually like this. After midnight, the planner would use FMS to retrieve daily orders for limousine services from the online reservation system. This includes orders from clients, travel agents and hotel concierges. Once retrieved, a Web-based spreadsheet displays details of orders. In addition, FMS would automatically connect airport transfer orders with real-time flight info/statuses. Orders with special needs, such as VIP, meet/assist, or stopovers will be highlighted with color flags.

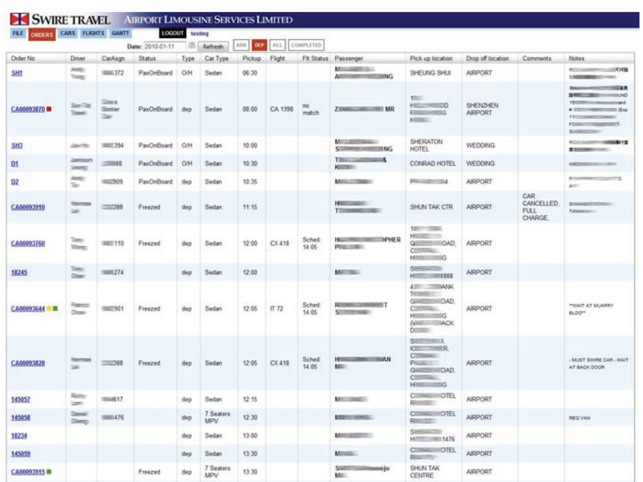


Figure 2. The FMS Orders Spreadsheet Screen

The planner may, at this time, also update availability for all limousines/vehicles; some may be scheduled for maintenance and will not be available. Once order and limousine data are updated, the planner would proceed to assign vehicles to orders using the AI engine. Orders would usually be assigned vehicles in time periods of a few hours each. In any case, VIP orders would usually be assigned first. As orders are completed, chauffeurs would report back on their statuses: when they reached the pick-up point, when the client is onboard, and when they reach the destination. All these statuses are updated into FMS. The FMS Gantt chart provides a bird's eye view of the entire day. Orders with special needs are also highlighted.

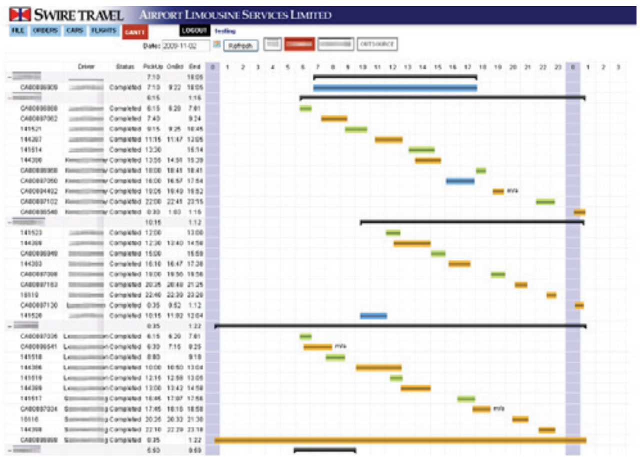
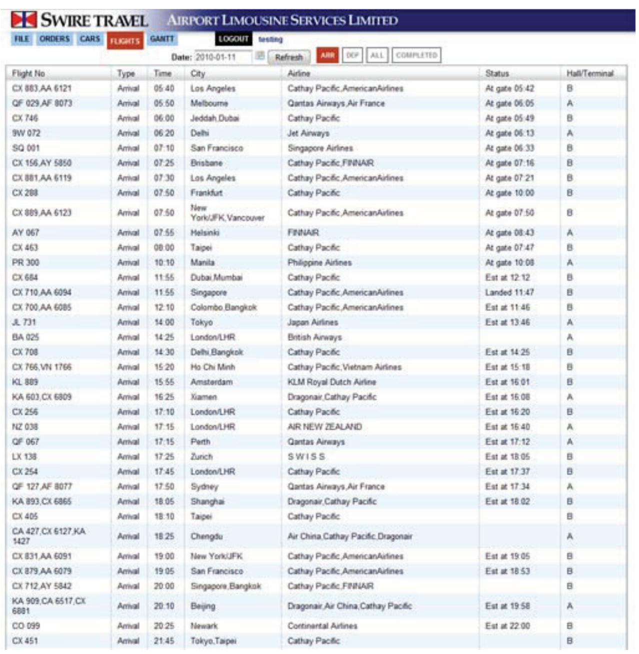


Figure 3. The FMS Gantt Chart Screen

All orders are automatically connected with flight information. A separate flight information screen is also available to show relevant fights that have passengers serviced by ALS or all flights of the day. The hall/terminal information helps the chauffeurs determine where to meet up with the passenger.



Flight No	Type	Time	City	Airline	Status	Hall/Terminal
CX 883,AA 6121	Arrival	05:40	Los Angeles	Cathay Pacific,AmericanAirlines	At gate 05:42	B
QF 029,AF 8073	Arrival	05:50	Melbourne	Qantas Airways,Air France	At gate 06:05	A
CX 746	Arrival	06:00	Jeddah,Dubai	Cathay Pacific	At gate 05:49	B
9W 072	Arrival	06:20	Delhi	Jet Airways	At gate 06:13	A
SQ 001	Arrival	07:10	San Francisco	Singapore Airlines	At gate 06:33	B
CX 156,AY 5850	Arrival	07:25	Brisbane	Cathay Pacific,FINNAIR	At gate 07:16	B
CX 881,AA 6119	Arrival	07:30	Los Angeles	Cathay Pacific,AmericanAirlines	At gate 07:21	B
CX 288	Arrival	07:50	Frankfurt	Cathay Pacific	At gate 10:00	B
CX 889,AA 6123	Arrival	07:50	New York/JFK,Vancouver	Cathay Pacific,AmericanAirlines	At gate 07:50	B
AY 067	Arrival	07:55	Helsinki	FINNAIR	At gate 08:43	A
CX 463	Arrival	08:00	Taipei	Cathay Pacific	At gate 07:47	B
PR 300	Arrival	10:10	Manila	Philippine Airlines	At gate 10:08	A
CX 684	Arrival	11:55	Dubai,Mumbai	Cathay Pacific	Est at 12:12	B
CX 710,AA 6094	Arrival	11:55	Singapore	Cathay Pacific,AmericanAirlines	Landed 11:47	B
CX 700,AA 6085	Arrival	12:10	Colombo,Bangkok	Cathay Pacific,AmericanAirlines	Est at 11:46	B
JL 731	Arrival	14:00	Tokyo	Japan Airlines	Est at 13:46	A
BA 025	Arrival	14:25	London/LHR	British Airways		A
CX 708	Arrival	14:30	Delhi,Bangkok	Cathay Pacific	Est at 14:25	B
CX 766,VN 1766	Arrival	15:20	Ho Chi Minh	Cathay Pacific,Vietnam Airlines	Est at 15:18	B
KL 889	Arrival	15:55	Amsterdam	KLM Royal Dutch Airline	Est at 16:01	B
KA 603,CX 6809	Arrival	16:25	Xiamen	Dragonair,Cathay Pacific	Est at 16:08	A
CX 256	Arrival	17:10	London/LHR	Cathay Pacific	Est at 16:20	B
NZ 038	Arrival	17:15	London/LHR	AIR NEW ZEALAND	Est at 16:40	A
QF 067	Arrival	17:15	Perth	Qantas Airways	Est at 17:12	A
LX 138	Arrival	17:25	Zurich	S W I S S	Est at 18:05	B
CX 254	Arrival	17:45	London/LHR	Cathay Pacific	Est at 17:37	B
QF 127,AF 8077	Arrival	17:50	Sydney	Qantas Airways,Air France	Est at 17:34	A
KA 893,CX 6865	Arrival	18:05	Shanghai	Dragonair,Cathay Pacific	Est at 18:02	B
CX 405	Arrival	18:10	Taipei	Cathay Pacific		B
CA 427,CX 6127,KA 1427	Arrival	18:25	Chengdu	Air China,Cathay Pacific,Dragonair		A
CX 831,AA 6091	Arrival	19:00	New York/JFK	Cathay Pacific,AmericanAirlines	Est at 19:05	B
CX 879,AA 6079	Arrival	19:05	San Francisco	Cathay Pacific,AmericanAirlines	Est at 18:53	B
CX 712,AY 5842	Arrival	20:00	Singapore,Bangkok	Cathay Pacific,FINNAIR		B
KA 909,CA 6517,CX 6881	Arrival	20:10	Beijing	Dragonair,Air China,Cathay Pacific	Est at 19:58	A
CO 099	Arrival	20:25	Newark	Continental Airlines	Est at 22:00	B
CX 451	Arrival	21:45	Tokyo,Taipei	Cathay Pacific		B

Figure 4. The FMS Flight Info Screen

If new orders are received during the day, the AI engine can be invoked to provide proposals for vehicle assignments. Besides, their own fleet of limousines and vehicles, ALS may also out source some of the orders to third parties. One of the scheduling objectives is to minimize the amount of out-sourcing needed. Figure 5 shows a typical popup window with AI proposals. The proposals are prioritized. The highest priority "AI Suggestion" list contains vehicles that satisfy all the constraints and their current assignments. The "Available" list contains other feasible alternatives. The final list contains other vehicles that are not available. However, they may be reassigned if needed.

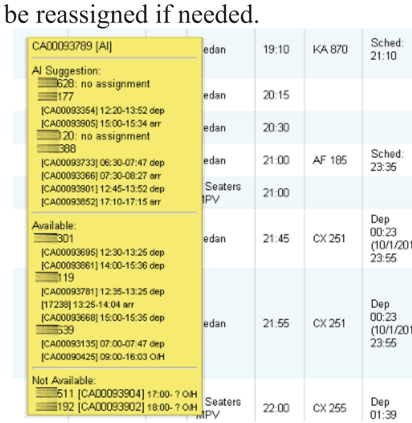


Figure 5. The AI Proposal Popup

System Architecture

The Fleet Management System (FMS) was designed with a modern Web 2.0 architecture. Client-side is a JavaScript-based rich internet application (RIA) (Wikipedia 2010a). The server-side was coded in Python and uses a model-view-controller (MVC) architecture (Burbeck 1992) and exposing all its services as xml/json RESTful Web services (Costello 2010, Rodriguez 2010, Wikipedia 2010b, Tyagi 2006).

The MVC architecture was provided by our own lightweight Python Web application framework. The AI constraint-based scheduling algorithm was coded in Python on top of our own multi-language AI platform (Chun 1999, 2000, 2004, 2005, 2007). This same Web framework and AI platform was used for a workforce management system that supported some of the Olympics activities in Hong Kong (Chun 2008). The MVC model provides separation of concerns and simplifies maintenance.

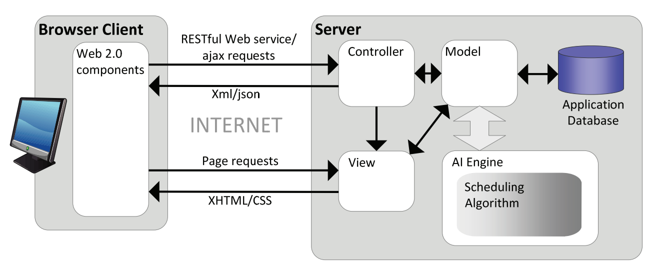


Figure 6. The FMS Software Architecture

The following highlights some of the tools/libraries we used for the client-side development.

Web 2.0 Technologies

Besides the basic Web framework and our AI platform, the development of the FMS benefited greatly from readily available Web 2.0 open-source tools and libraries. These tools and libraries allowed us to focus on solving the AI problem instead of designing user interfaces and interactivities. Because these tools are widely used, software development risks are greatly reduced. Furthermore, because these tools/libraries usually follow established interactivity best practices, most users can easily learn how to use these Web 2.0 components without much training.

To fully take advantage of these tools, all our server-side services were coded as RESTful Web services so that they can easily return xml/json data to drive these client-side tools and libraries. By decoupling the client-side and server-side with xml/json (JSON 2010) as the main interchange, we further simplify debugging/testing.

For any AI scheduling system, the main client-side user interface components are the spreadsheet, gantt chart and forms. For the client-side, we made extensive use of the Yahoo! UI (YUI) Library (YUI 2010). This library permitted us to easily create browser-independent application as well as provide a wide variety of basic utilities such as Ajax and xml/json handling. We also made use of several highly-flexible YUI widgets (i.e. web components). For example, the YUI DataTable widget allowed us to easily display data retrieved from server-side through Ajax in a spreadsheet format that can easily be manipulated by the user. The spreadsheet already provided sorting and mouse-handling capabilities. Spreadsheets are used in many FMS screens.

Besides spreadsheets, we also made use of a Web-based Gantt chart component called jsgantt (Jsgantt 2010) that conveniently displays xml data in a Gantt chart format. Items in the Gantt chart can also be manipulated by the user. To code various online forms with modern Web 2.0 styling, we made use of facilities provided by wufoo (Wufoo 2010).

Ease of Development

This approach of using an MVC Web framework to decouple the view and model/logic (including AI); exposing all services as xml/json RESTful services; and leveraging open source Web 2.0 components for rich interactivity allows us to quickly create modern AI applications that are easy to develop, easy to use and easy to maintain.

Uses of AI Technology

There has been decades of work on different variations of the vehicle scheduling and routing problem (Ceder 2002, Wren 1998). This includes scheduling vehicles to pickup and/or deliver goods, mapping shortest routes for vehicles to visit all destination points, dispatching vehicles from one or more depots, etc. The constraints are usually the destinations, location of depots, travel distances, travel times, sizes of payload, capacity of vehicles, etc. Many researchers model this problem as a type of travelling salesman problem. Some researchers offer solutions using constraint-programming (Tsang 1993, Shaw 1998, Backer et.al. 2000), linear programming (Foster 1976), or expert systems (Waters 1990).

The limousine scheduling problem is a bit different from the traditional vehicle scheduling problem (VSP). Firstly, for VSP, the pickup/delivery times are usually flexible and can be scheduled. For limousines, the pickup times are fixed. In fact, being on time is of utmost importance. The limousine scheduling problem is more dynamic; orders get cancelled and added throughout the day as well as the need to anticipate traffic delays.

In Laurent and Hao's work (Laurent and Hao 2007), they proposed a combined vehicle and driver scheduling algorithm using a combination of constraint-programming and simulated annealing. For their problem, driver scheduling was an issue. In our case, drivers are already associated with a team and/or vehicle. In Laurent and Hao's work, their limousines were dispatched to service roughly one job per day within the Paris area, whereas our client needed to dispatch each limousine to service many jobs per day within the metropolitan Hong Kong area.

Our limousine scheduling problem is related to taxi scheduling (Horn 2002). However, resources for taxi dispatching is a bit homogeneous and with less demand for special requests or value-added services. Borndörfer et.al. (Borndörfer et. Al 1999), outlines a problem to schedule vehicles in a dial-a-ride system for handicapped people in Berlin – Telebus. The system operates 1,000 to 1,500 trips per day. They solved this massive problem by partitioning the trips and clustering them into smaller sets that can be scheduled using a branch-and-cut algorithm. Bus vehicle and driver scheduling is also related to our work (Huisman and Wagelmans 2006, Ismail and Ang 2005). However, the bus problem is mainly driven by a rigid timetable and fixed stops.

We also performed a survey of commercially available limousine software (LMS 2010, CorporateCarOnline 2010, Limo-Aide 2010, Trip Tracker 2010, LimoNexus 2010). As far as we can tell, most of them provided only screens for a human planner to input schedules. Most relied on traditional client-server model and were not as flexible or user-friendly. However, most of these software packages do provide a fully integrated system that ranges from online reservation, scheduling, dispatching, to billing.

AI Model

For our FMS project, we modeled the vehicle scheduling problem as a constraint-satisfaction problem (CSP); whereby a solution is found through the satisfaction of a number of constraints or criteria (Tsang 1993). Our constraints were coded using our own AI platform (Chun 2008). This platform contains a set of libraries/utilities as well as a set of common AI algorithms, such as a rule engine, data mining algorithms, genetic algorithm, neural network, etc. In addition, our AI platform supports creating Python business objects based on the HR-XML standard (HR-XML 2007) – a library of XML schemas developed by the HR-XML Consortium, Inc.

For our limousine scheduling problem, a schedule is generated by repeated application of rules which generated labeling for the search algorithm (Apt 2001). The labels represent evaluation scores based on rule execution results and heuristics. The problem can be represented as:

• a set of trips, defined by pickup time, pickup location, destination, stopover points, car type (capacity), and services, and

• a set of vehicles, defined by car type (capacity), and features (such as cross border license plates).

The Scheduling Constraints

Constraints in FMS are divided into hard and soft constraints. Hard constraints are those that can never be violated. Soft constraints represent preferences that can be violated if necessary. The following are the key hard constraints considered by FMS:

- **car type** – the type of vehicle, such as Mercedes Benz sedans, stretched limousines, shuttle buses, cross border vehicles, etc. Once booked, it can only be changed with the approval from the client. Implicitly encoded with car type is capacity and car features (such as cross border license plates).
- **availability** – the availability of a vehicle. This includes whether the vehicle is available for the day (not in shop for service) or whether it is already assigned to service another order. Time needed to travel between orders is also factored into availability.

The following are the key soft constraints:

- **proximity** – if the job is to pick up a client at the airport, obviously a car already at the airport would be preferred. However, if no other car is available, a car from the city will be assigned to go empty to the airport.
- **team** – vehicles are divided into teams to service either general orders or individual hotel partners. A vehicle can be reassigned to service orders from another team if needed.

As we mentioned, unlike traditional vehicle scheduling problem, our limousine scheduling problem requires precision in timing. In order to achieve more accurate driving time predictions, we created two additional AI modules, which are described below.

Auto Correcting

Since orders are potentially entered by any of the hundreds of travel agents as well as thousands of clients, the information entered may contain typing errors. To use AI for scheduling, it must first process addresses accurately. We have collected data over a year and a half of use to create an AI pattern-matching module that accurately guesses destinations. This includes a combination of spelling correction as well as Hong Kong-specific locations (such as abbreviations, informal names, and alternative names). The module handles cities, roads, hotel names, landmarks, and well-known buildings.

Travel Time Prediction

Another key concern was in the accuracy of the AI drive time predictions. Our approach was to create a self-adapting AI module that automatically adjusts travel time based on actual historical values. The entire Hong Kong was first divided into roughly 70 regions/areas. An initial single drive time matrix was created manually to bootstrap the system. This initial matrix was then duplicated for weekday/weekend drive times as well as drive times during different periods of the day. All these resulting drive time matrixes are then self-modified over time as actual data are received. An averaging over a fixed time window is used to capture seasonal variations and to smooth out any unusual patterns. In addition, a separate database is maintained to map certain popular destinations, such as hotels or buildings, to known regions/areas.

Driver Assignment

Most vehicles are operated in two shifts. Drivers are assigned based on their experience with handling that type of vehicle and historical data on experience with a particular vehicle. In the next phase of this project, the AI system will be extended to perform duty rostering as well.

The Scheduling Algorithm

Figure 7 shows the process flow of the FMS scheduling algorithm. Each morning, most of the orders will be loaded. As the day progresses, additional last-minute orders will be received, mostly from hotels. Before the morning shift, the AI scheduling algorithm will be used to schedule within a certain time window of orders. Drivers will be dispatched over phone.

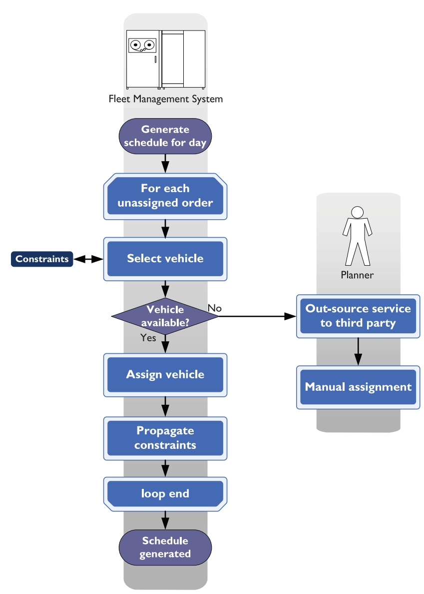


Figure 7. Process flow diagram for AI scheduling

The scheduling algorithm works like this. Each order contains a variable representing the vehicle to be assigned. These variables are then selected according to a heuristic.

The heuristic gives VIP orders the highest priority. After a variable is selected, labels are generated and matched against the scheduling constraints. In the event that no suitable vehicle can be found, the order will be flagged and an out-sourcing company will be contacted to provide the service. ALS maintains a list of several potential out-source companies to work with. Minimizing the number of out-sourced orders is of course an important business objective. In any case, all VIP orders must be handled with their own cars.

Application Use and Payoff

The day-to-day use of the Fleet Management System began 1 Jan 2009, initially for manual assignment; the AI-part of the system was deployed early June 2009. The AI scheduling feature of the system is used only by the planners and controllers. The viewing and inputting of orders can be accessed by several hundred travel agents within Swire Travel. In the future, the system will allow mobile access to chauffeurs for assignment retrieval and status updates.

Application Payoff

The key benefits from using FMS were:

- **Enable business growth** – Using FMS, ALS was able to handle a 100% increase in the number of weekly orders in the 2009 calendar year without needing any increase in number of planners/controllers. Figure 8 shows the number of orders per week, the number of vehicles used to service these orders, and the number of out-sourced orders. (The vertical scale is hidden to protect confidential commercial data.) Although the chart does not show the state of the manual operations prior to Phase 1, we believe the data would be similar to Phase 1. Phase 1 is mainly to put operations online. The benefits of AI do not come in until Phase 2.

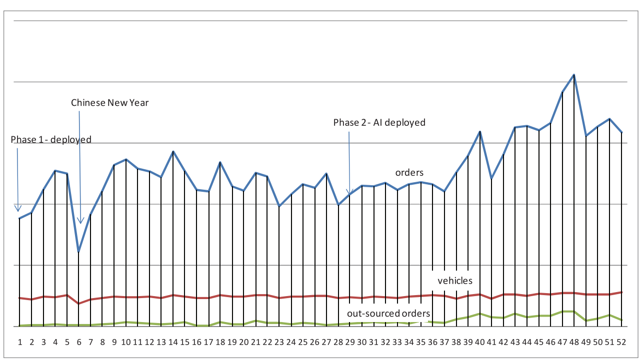


Figure 8. Plot showing how AI supported business growth

- **Maximize resource utilization** – Despite the over 100% increase in orders, the number of vehicles and drivers deployed remained fairly constant, as Figure 8 indicates. We believe our AI system helped ALS optimize how their existing resources were deployed to meet increasing demands. Figure 9 shows the average number of orders serviced per vehicle per day after the Phase 2 AI deployment. Roughly 2 more orders were handled per vehicle each day, i.e. 40% improvement in utilization.
- **Minimize out-sourcing** – Despite the slight increase in out-sourcing due to dramatic increase in business volume, the number of out-sourced orders was still maintained to be within only a few percentages of total orders throughout the year (see Figure 8).
- **Bird's eye view of operations** – FMS consolidates all information related to orders, vehicles, drivers, and flights in one place and displays them in Gantt charts and spreadsheets. Planners/controllers can easily visualize operation statuses as well as assignment statuses and workload for each vehicle for the entire day.

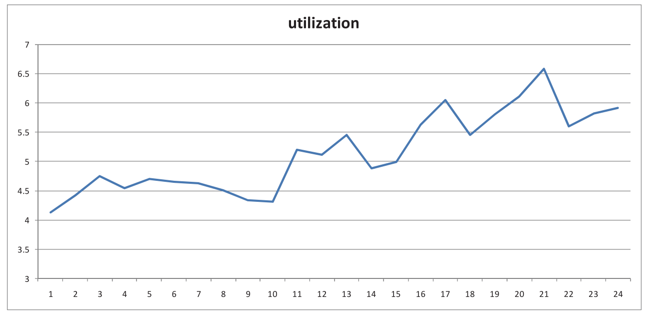


Figure 9. Average num orders serviced per vehicle

- **Respond quickly to changes** – With the bird's-eye view and the AI proposal mechanism, ALS is better equipped to handle sudden changes, such as last-minute orders, and delays.

Application Development and Deployment

The design and development of the Fleet Management System (FMS) was divided into two phases – Phase 1: Web-based FMS and Phase 2: AI Automatic Car Dispatching. Phase 1 provided the core Web-based system with all the screens and features needed to support manual scheduling and vehicle dispatching. Phase 2 adds to this AI vehicle scheduling support and a Web-based Gantt chart.

This project began in April 2008 and was performed by the CityU Professional Services Limited (CPS 2010), a non-profit subsidiary of the City University of Hong Kong (CityU 2010).

Application development was performed using a modern agile iterative approach and user-centered design. The result of first iteration was released in July 2008 for user feedback. After two more iterative refinements, Phase 1 of the project was completed in Dec 2008 and was deployed 1 Jan 2009. Phase 1 provided core capabilities such as user/role management, database to store vehicle and driver information, integration with booking data (from hotel, online web-based booking, and travel agents), integration with flight data (from airport), manual scheduling features (vehicle/driver assignments), data export features, and screens to display orders, vehicle location/statuses, driver assignments, and flight schedules/statuses.

The Phase 2 AI work began in March 2009. The first release was deployed early Jun 2009. After refinement, the final release was deployed in late July 2009. Phase 2 provided trip travel time estimation, AI scheduling capabilities, and a new Gantt chart screen.

The project team consists of a project leader from the customer who coordinated the requirements from the planners and other stake-holders. The AI development was led by Dr. Andy Chun, who also performed the design and was the lead coder, a system administrator who handled the systems configuration and routine maintenance work, and software engineers who performed testing.

The FMS development leveraged extensively on open source libraries and routines for JavaScript and Python. In addition, we used our existing Python Web frameworks and AI libraries.

The main design and development tasks were to integrate various data sources, create the various screens and functions, and to code the AI constraints, heuristics and scheduling algorithm. We also had to make sure that the FMS match the operational needs and workflow of the planners and that exported data was sufficient for billing and management reporting purposes.

Deployment

The deployment of the mission critical FMS was broken down into 2 stages – 6 months of manual approach before AI was deployed. The first "manual" stage deployment was on 1 Jan 2009. The FMS was fully used to support daily operations for all vehicle/driver assignments. The system is integrated with several data sources. However, scheduling was manual. Prior to deployment, the system had been up and running for 6 months for user testing/training and to ensure all data inputs were correct.

The second "AI" stage deployment was in July 2009. This version roughly coincided roughly with new business expansion and new vehicle purchases. The use of AI helped the company expand business without additional planner/controller.

Maintenance

The frontline support/maintenance for the FMS is provided by Swire Travel's own IT team. Since the FMS is a Web-based solution, many of the traditional maintenance tasks, such as exporting data, are provided in "self-service" mode, e.g. through downloading of MS Excel files through the Web. All site-specific configuration data are maintained as XML or MS Excel files. For example, the limousine inventory and driver details are all stored as MS Excel files and uploaded whenever changed. For the AI part, no on-going maintenance is needed since the rules and constraints governing vehicle assignment are relatively static.

Future

In addition to the original Phase 1 and 2, a separate Phase 3 is being planned to extend the system for AI driver rostering. In addition, there is plan to open up FMS access to drivers so that they can directly retrieve client details (such as addresses, phones, etc) and update location and statuses using mobile devices.

Conclusion

This paper explained how we took advantage of modern Web 2.0 architecture and open-source tools/libraries to successfully create a highly user-friendly and interactive AI system. We also shared our experiences in using AI to alleviate a planner/controller staff shortage problem that was impeding business growth. We also highlighted the domain knowledge relating to vehicle dispatching for the limousine industry. In terms of business benefits, our AI system was able to help our client achieve 100% growth in number of weekly orders while maintaining fleet size and without needing additional human planners/controllers or a major increase in out-sourcing. Resource utilization improved by 40%.

References

ALS. 2010. Airport Limousine Services Web Site. Available at: http://www.airportlimousine.com.hk/home.aspx

Apt, K.R., Monfroy, E. 2001, "Constraint programming viewed as rule-based programming," *Theory and Practice of Logic Programming*, v.1 n.6, p.713-750, November 2001.

Backer, B. D., Furnon, V., Shaw, P., Kilby, P., and Prosser, P. 2000. "Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristics." In the *Journal of Heuristics*, Vol 6, Issue 4 (Sep. 2000), pp. 501-523.

Borndörfer, R., Grötschel, M., Klostermeier, F., Küttner, C. 1999. "Telebus Berlin: Vehicle Scheduling in a Dial-a-Ride System," In the *Proceedings of the 7th International Workshop on Computer-Aided Transit Scheduling*, Berlin, pp. 391–422.

Burbeck, S. 1992. "Application Programming in Smalltalk-80: How to use Model-View-Controller (MVC)." University of Illinois in Urbana-Champaign (UIUC) Smalltalk Archive. Available at st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html

Ceder, A., 2002. "Urban Transit Scheduling: Framework, Review and Examples." In the *Journal of Urban Planning and Development*, 128 (4), 225–244.

Chun, H.W., Chan H.C., Tsang M.F., and Yeung W.M., 1999, "HKIA SAS: A Constraint-Based Airport Stand Allocation System Developed with Software Components," In *Proceedings of the Eleventh Conference on Innovative Applications of Artificial Intelligence*, pp. 786-793, Orlando, July, 1999.

Chun, H.W., Chan H.C., Lam P.S., Tsang M.F., Wong J. and Yeung W.M., 2000. "Nurse Rostering at the Hospital Authority of Hong Kong," In *Proceedings of the Twelfth Conference on Innovative Applications of Artificial Intelligence*, Austin, August, 2000.

Chun, H.W. and Yeung, Wai Ming, 2004. Rule-based Approach to the Validation of Subway Engineering Work Allocation Plans, In the *Proceeding of the International Conference on Computing, Communications and Control Technologies*, August 14-17, 2004, Austin, Texas, USA.

Chun, H.W., Yeung, W.M., Lam, P.S., Lai, D., Keefe, R., Lam, J., and Chan, H., 2005. "Scheduling Engineering Works for the MTR Corporation in Hong Kong," In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence*, Pittsburgh, July, 2005.

Chun, H.W. 2007. "Using AI for e-Government Automatic Assessment of Immigration Application Forms," In *Proceedings of the 19th Conference on Innovative Applications of Artificial Intelligence*, Vancouver, July, 2007.

Chun, H.W. 2008. "Using AI for Olympic Equestrian Event Preparation," In *Proceedings of the 20th Conference on Innovative Applications of Artificial Intelligence*, Chicago, July, 2008.

CityU. 2010. The City University of Hong Kong Web Site. Available at: www.cityu.edu.hk

CorporateCarOnline. 2010. CorporateCarOnline Web Site. Available at: http://www.limosinesoftware.com/

Costello, R.L. 2010. "Building Web Services the REST Way" Available at: http://www.xfront.com/REST-Web-Services.html

CPS. 2010. The CityU Professional Services Limited Web Site. Available at: www.cityu.edu.hk/cityu/dpt-admin/cps.htm

Foster, B.A. and Ryan, D. M. 1976. "An Integer Programming Approach to the Vehicle Scheduling Problem," In *Operational Research Quarterly*, Vol. 27, No. 2, Part 1 (1976), pp. 367-384.

Horn, Mark E. T., 2002. "Fleet Scheduling and Dispatching for Demand-Responsive Passenger Services," In the *Transportation Research,* Part C, 10 (1), pp.35–63.

HR-XML. 2007. The HR-XML 2.5 Release XML schemas. Available at www.hr-xml.org/hr-xml/wms/hr-xml-1-org/index.php?id={E00DA03B685A0DD18FB6A08AF0923DE0|139|2}

Huisman, D., Wagelmans, Albert P.M., 2006. "A solution approach for dynamic vehicle and crew scheduling," In the *European Journal of Operational Research*, Volume 172, Issue 2, 16 July 2006, pp. 453-471.

Ismail, Z and Ang, P.S. 2005. "Integer Programming Approach In Bus Scheduling And Collection Optimization," In the *Jurnal Teknologi*, 43 (C). pp. 1-14.

Jsgantt. 2010. Jsgantt site at code.google. Available at: http://code.google.com/p/jsgantt/

Laurent, B. and Hao, J. 2007. "Simultaneous vehicle and driver scheduling: A case study in a limousine rental company," In the Computers and Industrial Engineering. Vol. 53, Issue 3 (Oct. 2007), pp.542-558.

LMS. 2010. Limousine Management Systems Web Site. Available at http://www.lmsgold.com/

Limo-Aide. 2010. Comp-Easy Software LLC Web Site. Available at: http://www.compez.com/Software/Limo_Aide_tm_/limo_aide_tm_.html

LimoNexus. 2010. BlueNexus Web Site. Available at: http://www.bluenexus.com/limonexus.html

Rodriguez, A. 2010. "RESTful Web services: The basics." Available at: http://www.ibm.com/developerworks/webservices/library/ws-restful/index.html

Shaw, P. 1998. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems." In *Proceedings of the 4th international Conference on Principles and Practice of Constraint Programming* (October 26 - 30, 1998). M. J. Maher and J. Puget, Eds. Lecture Notes In Computer Science, Vol. 1520. Springer-Verlag, London, pp.417-431.

Swire Group. 2010. Swire Group Web Site. Available at: http://www.swire.com/eng/global/home.htm

Swire Travel. 2010. Swire Travel Web Site. Available at: http://www.swiretravel.com/en/

TIC. 2010. Travel Industry Council of Hong Kong Website. Available at: http://www.tichk.org/public/website/en/index.html

Trip Tracker. 2010. Trip Tracker Web Site. Available at: http://www.triptracker.com/

Tsang, E. 1993. *Foundations of Constraint Satisfaction*. Academic Press.

Tyagi, S. 2006. "Restful Web Services," In Sun Developer Network Web Site. Available at: http://java.sun.com/developer/technicalArticles/WebServices/restful/

JSON. 2010. JavaScript Object Notation Web Site. Available at: http://www.json.org/

Waters, C.D.J. 1990. "Expert Systems for Vehicle Scheduling," In *The Journal of the Operational Research Society*, Vol. 41, No. 6, pp. 505-515, 1990.

Wikipedia. 2010a. "Rich Internet Application," In the Wikipedia Web Site. Available at: http://en.wikipedia.org/wiki/Rich_Internet_application

Wikipedia. 2010b. "Representational State Transfer," In the Wikipedia Web Site. Available at: http://en.wikipedia.org/wiki/Representational_State_Transfer

Wren, A., 1998. "Heuristics Ancient and Modern: Transport Scheduling Through the Ages," In the *Journal of Heuristics*, 4 (1), 87–100.

Wufoo. 2010. Wufoo Website. Available at: http://wufoo.com/

YUI. 2010. Yahoo! UI Library Website. Available at: http://developer.yahoo.com/yui/