

A RULE BASED INTELLIGENT SYSTEM FOR ASSISTING BUS FLEET SCHEDULING

Dimitris Chalvatzaras
SoFar
Zakynthos 29100

Sophia Daskalaki
University of Patras, Department of Electrical & Computer Engineering
Rio, Patra 26500, Greece

ABSTRACT

This paper presents BusPlanner, an innovative web based Bus Fleet Management System. The goal of the system is to produce operable and cost effective timetables for touristic buses with predefined routes using Artificial Intelligence tools. The system takes into account well defined factors such as passenger demands, travel routes, travel costs, but also others that are rather ambiguous and fuzzy. The system provides two main advanced features.

The first refers to an Artificially Intelligent (AI) engine that can produce timetables in real time while taking into account mean delays and geographical positions of the traveling buses. The AI engine has been implemented as a rule based production system. Moreover, rule scripts can be altered while the engine is online making optimization of schedules faster and the scheduling patterns even more cost-efficient. The engine is connected with GPS trackers and actively receives feedback from the administrator concerning the progress of the bus trips. This information allows for real time rescheduling, thus maximizing the efficiency of the bus fleet. The second feature refers to an interactive timeline where the user may alter or create a timetable while getting feedback from the AI rules that are able to evaluate the created solution in real time.

BusPlanner has been used for more than a year by the bus company IonianTransport, managing more than 50 buses and serving more than 300 custom trip routes every day. During this period several evaluation experiments have been performed proving that the system can produce robust and cost effective timetables for the buses. At the same time it proved that its usage can raise significantly the productivity of the scheduler of the company as it assists his/her work for the production of timetables with minimum effort.

Key words: *Bus Scheduling, Artificial Intelligence, Rule Based Planning*

INTRODUCTION

In many touristic places private bus companies offering hiring services every day have to deal with a large number of trips that have to be scheduled on time periods that vary day-by-day. In addition during each day they face all sort of delays that may arise from plane delays, passengers not been ready on time for pick up, etc. In contrast to public bus transportation, where they have to execute fixed trips no matter what, private buses cannot just skip a delayed passenger. These conditions demand for constant rescheduling of the bus schedules and alterations to the timetable of the whole fleet in order to overcome delays, meet the needs of upcoming trips, keep all passengers satisfied and at the same time keep cost at a minimum level.

BusPlanner is designed and implemented as a web based system for supporting private bus fleet management and time scheduling. The system combines state of the art AI technologies with innovative user interface and aims to ease human decisions during the bus scheduling process. The key feature of the system is that it successfully balances the control between the Artificially Intelligent (AI) optimization engine and the scheduling administrator during the timetable building process and moreover putting the user on top of the processing procedure. The design of the system focused on providing all the necessary visual tools that may assist the user during the manual process of timetabling and on placing optimization on the background. The AI optimization tool functions mainly as an assistant that is able to provide live feedback for possible conflicts, cost estimation, etc. Nevertheless, if the user feels safe enough he/she can give full control to the AI Engine over the process of timetable production/alteration.

This paper focuses mainly on the AI features of the BusPlanner scheduling system. Also it outlines the good practices for user interaction with the AI functionalities through the gathered experience from a case study with the bus company Ionian Transport. The presented system does not deal with the automated routing problem, since the specific field of transport has to serve routes that are specified by the clients.

During the last few decades scheduling problems have been well addressed such as in Desrochers et al. (1990), Cardoena et al. (2010), and Ernst et al. (2004). At the same time many different AI techniques and algorithms have been introduced that are able to solve a great number of these types of problems. In academic research there is an extended number of articles where similar approaches have been applied to the classroom timetabling problem such as heuristic search (e.g. Osman, 1993; Burke and Costa, 1994), distributed constraint-based search (Chun and Chan, 1999), evolutionary algorithms (e.g. Newall, 1999), ant colony algorithms (Socha et al., 2003), graph based algorithms (e.g. Burke et al., 2001).

In literature our case scenario scheduling problem is referred to as the “Vehicle Routing Problem with Time Windows” (Kallehauge et al., 2005) which is under the bigger set of problems classified as “Multiple Depot Vehicle Scheduling Problem (Mesquita and Paixao, 1992). Despite the fact there has been so much research in this field, a survey around bus fleet management software applications revealed that it is not easy to find applications that include dynamic time-scheduling functionalities based on AI techniques. The majority of the fleet management applications focus on real time bus GPS tracking, fuel consumption management, route optimization and manual resource handling (e.g. Aceroute, GoChart, RTA Fleet Maintenance System).

Scheduling on the field of private bus hiring transportations is a hard task. These types of transportations do not have daily fixed routes. Routes are on demand and change daily. Moreover the duty cycle of a bus trip includes many unpredictable hazards such as time

delays, bus malfunctions etc. making fully automated production of timetables inefficient and unreliable for the users.

BUS SCHEDULLING CONTEXTUAL DESIGN

In order to capture all the crucial factors that influence the daily trip scheduling process of a private bus hiring company we followed a contextual design approach. Contextual Design is a method for developing applications that focuses on how the user performs tasks within the context of the work environment itself (Beyer and Holtzblatt, 1998).

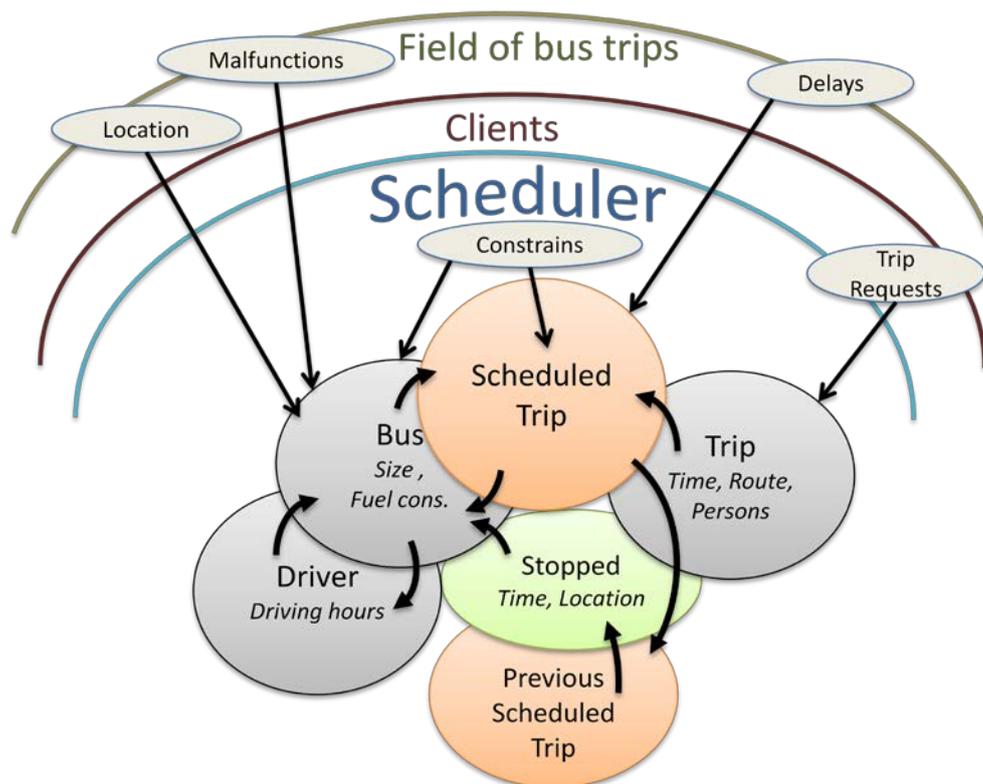


Figure 1. Contextual Analysis diagram

In order to proceed with such an approach several interviews were held with bus scheduling experts. The interviews were exhaustive regarding their daily routines for gathering information, organizing data and criteria but most importantly regarding the production of the bus fleet timetable. The context analysis unveiled the most crucial entities that get involved in the scheduling workflow and their interactive roles which influence the quality of the resulting solutions. It became very clear to us, however, that the scheduler is on top of the whole chain of actions, and it is crucial to stay on top without getting the feeling that he/she is substituted by an AI machine. The overall view of the Contextual Analysis diagram is presented in Fig.1.

Based on the findings of this analysis the functionalities of the system were defined. Although the most important specifications that aroused from this analysis concern the design of the AI engine and also the data models which are used during the scheduling process.

CONCEPTUAL DESIGN OF BUSPLANNER

The system has been designed to offer several functions assigned to a number of modules. For better understanding of the system usage a brief view of the conceptual architecture is presented in Fig. 2.

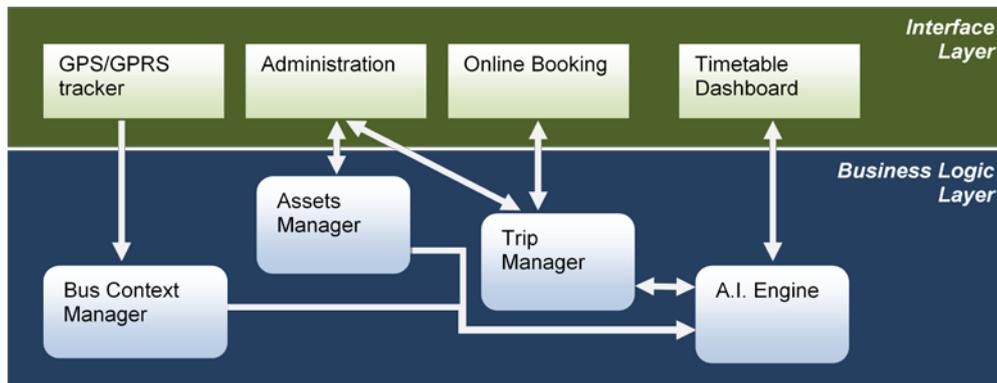


Figure 2. Abstract System Architecture

The system comprises two layers, the Interface and the Business Logic. While the users interact with the system through the Interface layer, the Business Logic layer processes all the information that can be gathered in order to find the best solutions for the user while managing the fleet resources in the best way.

We concentrate here in the heart of the system, the Business Logic layer. Its basic modules are:

- **Bus Context Manager.** This module interfaces with the GPS/GPRS trackers on the buses. The tracker transmits constantly the current position of the bus and also messages concerning the state of the engine, doors, etc. The module stores and processes this information and is able to infer the geographical name of the bus position, the state of the bus (moving, stopped, refueling, etc.) and also to re-estimate the duration of the currently executed trip.
- **Assets Manager.** This module is designed for managing the information that refers to drivers, buses, bus stop locations, travel information (distances and estimated traveling time), financial data, fuel consumption and the clients of the company.
- **Trip Manager.** The functionalities of this module support trip handling requests. This module is coupled with an online booking interface where the clients can place trip requests. It also offers a set of tools to the administrator (through the administration interface) for checking validity and confirming the trip request.
- **AI Engine.** This module is used specifically for the production and evaluation of timetables. It is coupled with an interactive dashboard where the administrator has all the necessary tools for administrating the scheduling process. More details for this module are presented in the next paragraph.

ARTIFICIALLY INTELIGENT ENGINE

The Artificially Intelligent (AI) Engine module of the BusPlanner system has been implemented as an AI production system (Luger and Stubblefield, 1999). The basic idea behind such system is to automatically produce a large number of possible solutions of a problem with the ultimate goal to locate the optimal solution, which can be identified using evaluation properties that have a cost effect for the given problem. The implementation of the routines in this module has been based on the OptaPlanner (former Drools Planner) java framework. Other good examples of the use of this specific framework in the field of scheduling can be found in (McCollum, 2009) and (De Smet, 2008).

The Timetable Production System

For our problem a solution is defined as a full day's timetable that includes all the buses of the fleet and all client requirements for the day. In the current implementation every possible solution is evaluated in terms of feasibility, fuel consumption and dispatching cost for the bus.

The criteria that are used for the evaluation of solutions are classified either as hard, which produce hard scores, or soft which produce penalty scores. The overall score of a timetable solution is the sum of all scores produced by the criteria. Ideally an optimal solution would have 0 hard and soft score.

Hard criteria refer to constraints that cannot be violated; otherwise the solution would be infeasible. Hard constraints for our problem are:

- *Overlaps between assigned trips.* This constraint refers to the case where one bus has been assigned to more than one trip at the exact same or overlapping travel time-period.
- *Insufficient bus capacity.* This constraint refers to the case where the capacity of a bus assigned to a trip is smaller than the one demanded.
- *Insufficient travel time.* This constraint is violated when a bus is not given enough time to reach the starting point of the next trip.

Soft criteria are related mainly to constraints that declare "preference", for example:

- *Fuel consumption,* it is preferred to have as little as possible.
- *Working hours violations,* some violations may be permitted but it is preferred to have as few as possible.
- *Fuzzy time constraints.* Since small delays between trip executions can be tolerated this type of constraints are classified as fuzzy. For example when a bus can include a trip on its timetable with a 5 minute delay this should not be excluded as a possibility but rather be penalized by a soft penalty score. The fuzziness threshold is configured by the user.
- *Number of buses dispatched.* It must be noted that, due to this criteria, soft penalty score can never be zero because this would mean that zero buses have been used.

In this project the criteria have been implemented as a set of rules scripted with the Drools Rule Language. The rules, during the evaluation stage produce hard or soft penalty scores. For example, a rule that checks the bus capacity scripted in DRL is presented in Fig. 3.

```

rule "BusNotEnoughSeats"
  when
    $bus : Buses($capacity:capacity);
    $trip : PlannedTrip(bus==$bus,passengerDemand>$capacity);
  then
    insertScore("Bus too small for the assigned trip",
      ConstraintType.NEGATIVE_HARD,10, $trip, $bus);
  end

```

Figure 3. A DRL script for checking the capacity of a bus

The rule in Fig. 3 says that the system will “fire” and produce a penalty score of 10 for every bus that is assigned to a trip with larger passenger demand than its capacity.

One of the key factors that contextual design determined was the entities that compose a daily timetable. The most important entities and their properties that function as the elements of the timetable can be classified in two domains.

The constraints domain which includes the following entities:

- **Route**, which is a linked list of locations accompanied with predetermined travel time estimations (provided by the Bus Context Manager)
- **Requested Trip**, which refers to a request for a predetermined route, with a specific starting time and a specific load of passengers.
- **Driver**, which refers to a driver that may be assigned to a bus and he/she is associated to a number of working hours.

The planning domain which includes two planning entities:

- **Bus**. The specifications of the bus that affect planning are the size (seats capacity) and its fuel consumption. Every bus is dispatched with a driver which is a planning variable. Also a stopped bus is characterized by the location that is stationed.
- **Planned trip**. A planned trip extends the entity of the requested trip. In addition it includes the bus that will perform the trip. The bus is considered to be a planning variable.

The above entities can be classified also as movables or static. Movable entities are the ones that can be moved during the planning process in order to produce a bus timetable. So for example a planned trip can be produced from a requested trip (static entity) when a bus (movable entity) is assigned to it. In case an alternative planned trip is needed then the existed planned trip is altered by changing (moving) the assigned bus with another.

Taking into account the solution criteria and the domain entities the AI optimization engine was designed to follow the steps that are described below:

- Step 1.** An initial solution is created by the user or by an algorithm that takes into account the evaluation rules and a random seed. The initial solution is passed over step 2.
- Step 2.** The solution is evaluated based on a set of rule criteria which create a penalty score for the given solution.

- Step 3.** As the set of solutions are piling up one is selected depending on a search strategy. The search strategy, followed in this project was a combination of Tabu Search (Glover and Manuel, 1999) and Simulated Annealing (Van Laarhoven and Aarts, 1987). This is an innovative feature of the production process where the Simulated Annealing strategy is engaged when the Tabu algorithm gets stuck for too long in a local minimum. In such a case the Simulated Annealing algorithm takes over and incorporates randomness and scoring criteria to detect a search subspace that might include a better solution than the one found so far.
- Step 4.** If the processing time limit has been surpassed then the best solution produced so far is exported as the optimal timetable; else the solution selected in this step is used for step 5.
- Step 5.** Given a feasible solution a new solution is generated by moving around one movable entity. The newly created solution is fed back to step 2

An advanced feature of the AI Engine is that the configuration of the rules or the search strategy is fully customized while the system is online. This feature makes the optimization of engine a cost effective process. The schedule expert can easily enhance the rule scripts by adding new ones or by fine tuning the existing ones without engaging a software developer. Also parameters that affect the search strategies (eg. Tabu size, annealing temperature) can be altered for optimizing the performance of the engine.

AI production performance

The AI Engine and the solutions it produces have been evaluated for more than a year. During the evaluation period different variations of the rule scripts were tested. The initial script was based mainly on the knowledge gathered by the scheduler expert. Later on as the evaluation experiments progressed the results were reviewed and the scheduler proposed variations to the rules that improved further the produced timetables.

The evaluation went through two phases. First we established the validity of the rules. A few experiments were performed where the schedule expert verified that the produced timetables were feasible. The next phase focused on optimizing the effectiveness of the produced timetables. This phase dealt mainly with fine tuning of the search strategy scheme and the penalty scores produced by the rules. Despite the fact that the optimization is an ongoing process at this moment the engine has reached a quite satisfactory level of performance.

During the second phase there were collected real trip data for a period of 6 months. Using this data several simulation experiments were performed with different planning configurations. After establishing a satisfactory level of quality for the produced timetables the processing time was also optimized. Fig. 4 presents performance results gathered from a set of experiments that were performed on a data set that included different days with a variety of trip loads and complexity. The chart in this figure presents (on the Trip load axis) the number of trips that were processed by every experiment, (on the Processing time axis) the processing time spent by the AI Engine to process the trips and the penalty score (on the Penalty Score axis) as it was improving during processing.

The experiments proved that the system produced timetables of the same quality regardless of the daily load for trips. It is also evident that in less than 6 minutes (in days with a moderate load less than 3 minutes) the engine can produce a satisfactory daily timetable. It must be noted the evaluation experiments were held by using the AI Engine in debug mode which reduces to a great extent the performance of the engine.

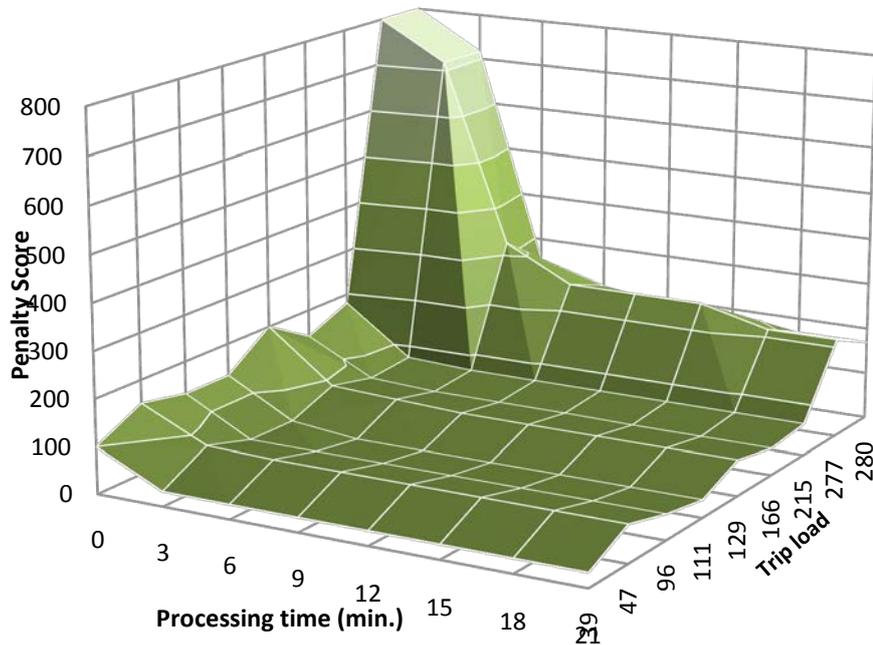


Figure 4. Evaluation of solutions produced by the AI Engine

IONIAN TRANSPORT CASE STUDY

Ionian Transport is a private bus company located in the island of Zakynthos and works mainly within the tourist industry. It has funded the BusPlanner system and has actively participated in the evaluation of the system.

Ionian Transport owns a fleet of 50 buses, which may be hired every day by different clients requesting their services. On an average day the company has to accommodate requests for more than 200 trips, a number that usually doubles during the month of August. In most cases the requests are registered just one day ahead. The fact that the trip requests completely vary from day-to-day in terms of starting time and duration, route and bus type renders the scheduling of fixed timetables completely useless.

In addition to the aforementioned problem, the realization of scheduled timetables quite often is not possible since many of the requested trips serve transit routes from the airport and are associated to arriving flights, which of course are subject to delays. Whenever a flight delays a chain reaction “shocks” the scheduled timetable of all buses. Of course other unpredictable factors such as breakdowns, traffic jams, problem with passengers, etc. may add more delays or even reduction on bus availability.

The collaboration with Ionian Transport has been very important for the development of BusPlanner. Initially several interviews were held with the staff, who were going to be the potential users but mainly emphasizing on capturing the expertise of the schedulers. These interviews established the first set of requirements and specifications. In the second phase a prototype was implemented for the BusPlanner system, configured and customized to meet the specifications that aroused from the interviews and the contextual analysis. The customization of the system introduced changes in the interface layer and the configuration specifications affected the AI engine module. The prototype was used to run several experiments specifically for the evaluation of the AI Engine and the user interaction with the prototype.

The second phase proved to be extremely important since it established a totally different course of action for the scheduling process. Initially the requirements were set so that the AI Engine would generate a full timetable for the whole day just after all incoming trip requests had entered in the system. Then the produced timetable would be implemented as it was by the buses of the fleet unless there was some detrimental delay that would force the scheduler to alter it and if necessary engage the AI Engine in the rescheduling process. Nevertheless this proved to be a non-productive workflow.

During the evaluation experiments of the prototype it became very clear that the scheduler would not trust a full timetable created automatically and therefore would not rely on it. On the contrary the scheduler was feeling more secure to engage the AI Engine for smaller portions of the schedule. With the completion of the evaluation period it was evident that the schedulers needed an interface where he/she would be able to view all the relevant information concerning the progress of the current bus trips and easily be able to alter the schedule while having evaluation feedback or engagement from the AI Engine.

A Use Case for AI-supported Timetables

Since automatic generation of timetables with the use of AI Engine was hard to be accepted by the scheduler, even if in most cases proved to be better than a human-generated timetable, the system design team decided to take a different turn. Instead of an AI black box solution the system would provide a set of AI tools that would assist the scheduler during the timetabling process. The Use Case displayed in Fig. 5 gives the final workflow implemented for Ionian Transport and elaborates upon the scheduler's experience on using the AI production system.

As depicted in Fig. 5 for the system there are four major user roles. The role of the *registrar*, who is responsible for collecting incoming trip requests, the *scheduler*, who is responsible for producing feasible timetables, the *client* who places requests and the *bus* which executes the trips. These user roles interact between themselves daily and their interaction, depicted with the workflow, comprises the following activities:

1. Clients enter their trip requests
2. The registrar evaluates the requests and groups them into valid trips.
3. The trips are forwarded to the scheduler
4. Every active bus reports its location through a GPS/GPRS module
5. The scheduler administers the overall scheduling by performing several asynchronous tasks. Its main tasks are:
 - a. Assign/Change a bus to a requested trip. The bus assignment can be either nailed or proposed. A nailed assignment means that the AI Engine is not allowed to change it. A proposed assignment means that the AI Engine may use it as part of the initial solution and is allowed to change it if such change may produce a better solution. It is worthwhile mentioning that since the scheduler can be involved in the generation of the initial solution, this result on boosting up the performance of the AI Engine.
 - b. Supervise progress of active bus trips. This task involves monitoring of the current position of the busses, the last position of the busses after all the trips have been executed, conflicts that concern overlaps, time constraints, etc.
 - c. If there is a need, the scheduler may change the starting time of a scheduled trip so to overcome problems that arise from delays or from other factors.

- d. Request support from the AI Engine for the generation of a full-day timetable or a portion of it.
- e. Review a detailed AI evaluation analysis for the current timetable.

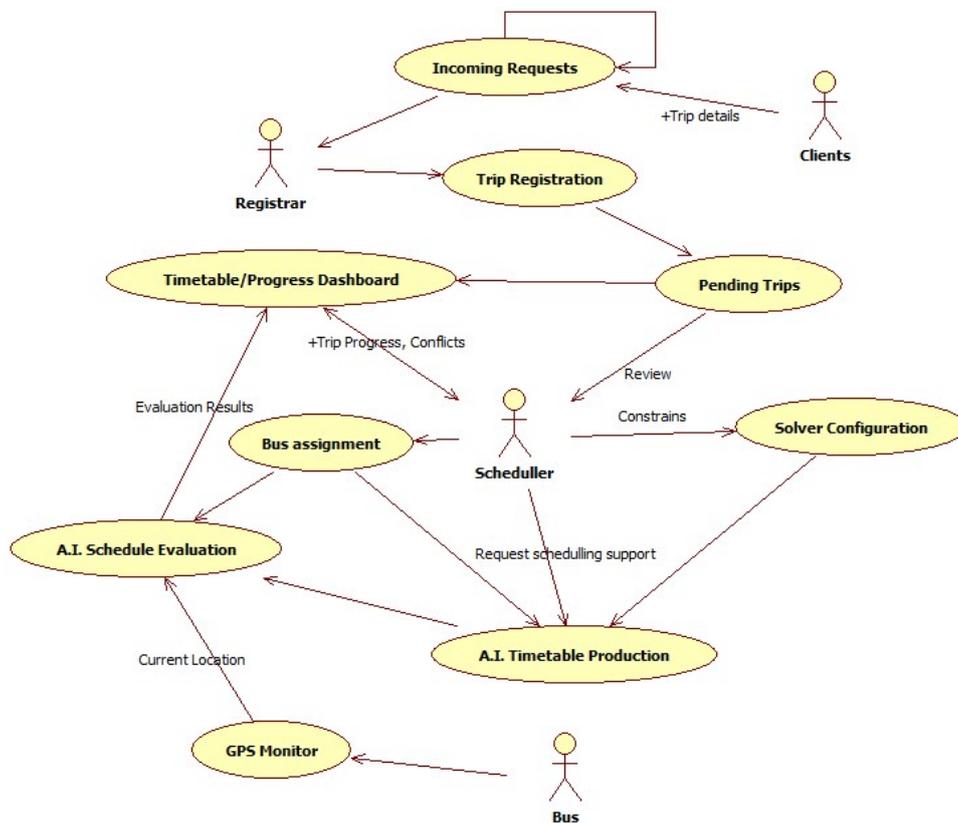


Figure 5. Use case for scheduling bus trips

The case study outlined the need of a user interface where the scheduler will be able to perform all the tasks captured by the use cases quickly and efficiently. As a good practice an interactive timetable dashboard (Fig. 6) has been implemented. The dashboard includes the following functionalities:

- Present progress of the current trips through a Gantt chart. The Gantt chart should also present with visual aids possible conflicts among the scheduled trips and possible upcoming delays.
- Provide administration control over buses and scheduled trips through table view including information that is valuable during the scheduling process such as current location of the bus, number of assigned trips etc. The administration tables can be filtered or sorted by various properties that characterize the buses or the trips as they have been defined by the case study
- Quick assignment of a trip to a bus with drag drop techniques.

able to reduce the processing time even further, while the quality of the produced timetables will be more enhanced.

On the other hand it was also shown that it is crucial to build a level of trust between the user and the system that provides automated solutions as claimed also in Waern and Ramberg (1996). In the case of Ionian Transport this level of trust was built by reducing the level of automation imposed while allowing more control to the user during the initialization stages of the scheduling process.

For the future our plans include further improvements of the working environment after additional evaluations on the usage of the dashboard and other modules of the system. More specifically there are plans for introducing AI rules that will take into account the GPS location and status of the buses in order to inference automatically whether the bus is on a valid stop or if there is an imminent hazard.

Lastly, in the near future a larger set of experiments has been planned where the human created timetables over a year will be evaluated and contrasted to the solutions created by the AI optimization Engine. These results will allow us to quantify the improvement a company may achieve by implementing such a system in their daily operations and will give us chances for further improving the performance of our system.

REFERENCES

- Beyer, H., and K. Holtzblatt, (1998). Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann.
- Burke, E. K., and Newall, J. P. (1999). A multistage evolutionary algorithm for the timetable problem, *IEEE Transactions on Evolutionary Computation*, Vol. 3(1), pp. 63-74.
- Burke, E.K., B. MacCarthy, S. Petrovic and R. Qu, (2001). Case-based reasoning in course timetabling: an attribute graph approach. In Case-Based Reasoning Research and Development, Vol. 2080, pp. 90-104.
- Cardoena B., Demeulemeesterb, E., and Beliën J. (2010), Operating room planning and scheduling: A literature review, *European Journal of Operational Research*, Vol. 201(3), pp. 921–932.
- Chun, H.W. and Chan, S.H.C. (1999), The Design of a Multi-Tiered Bus Timetabling System, *Proceeding of the 12th International Conference on Industrial and Engineering Applications of AI and Expert Systems*, Cairo, Egypt, 1999.
- Costa, D. (1994). A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, Vol. 76(1), pp. 98-110.
- De Smet, G. (2008). ITC2007-Examination Track, *Practice and Theory of Automated Timetabling (PATAT 2008)*, pp. 19-22.
- Desrochers M., Lenstraa J.K. , Savelsbergh M.W.P. (1990). A classification scheme for vehicle routing and scheduling problems, *European Journal of Operational Research*, Vol. 46(3), pp. 322–332.
- Ernst A.T, Jiang H., Krishnamoorthy M, Sier D. (2004). Staff scheduling and rostering: A review of applications, methods and models, *European Journal of Operational Research*, Vol. 153(1), pp. 3–27.

- Glover, F. and Manuel, L. (1999). *Tabu Search*, Springer, U.S.A.
- Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). Vehicle routing problem with time windows, pp. 67-98. Springer, U.S.A.
- Luger, G. F. and Stubblefield, W.A. (1999). *Artificial Intelligence Structures and Strategies for Complex Problem Solving*, Addison Wesley Longman Inc. pp.171-186.
- McCollum, B., P. McMullan, A.J Parkes, E.K. Burke and S. Abdullah, (2009). An extended great deluge approach to the examination timetabling problem. *Proceedings of the 4th Multidisciplinary International Scheduling: Theory and Applications 2009 (MISTA 2009)*, pp. 424-434.
- Mesquita, M., and Paixao, J. (1992). Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. In *Computer-Aided Transit Scheduling*, pp. 167-180. Springer Berlin Heidelberg.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, Vol. 41(4), pp. 421-451.
- Socha, K., Sampels, M., and Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art, *Applications of Evolutionary Computing*, Vol. 2611, pp. 334-345.
- Van Laarhoven, P.J. and Aarts, E.H. (1987). *Simulated Annealing*, Springer, The Netherlands.
- Waern, Y., and Ramberg, R. (1996). People's perception of human and computer advice, *Computers in Human Behavior*, Vol. 12(1), pp. 17-27.