Abstract— The goal of this project is the development of a solution for the conflict resolution in lines of low traffic, designated as the Meet & Pass Problem, in particular for the situations of a single line between stations.

The main conflicts to prevent and to detect are the presence of two trains with opposing directions in the same line and the proximity between trains, when a fast train follows behind a slow train. These conflicts derive essentially from the fact that there is a single line to separate the stations and that this line has to be used by the trains in both directions.

The proposed solution, which appears in the form of a decision support system, makes possible the accomplishment of a simulation of the schedules of a set of trains and the forecast of possible conflicts.

The developed application contains a graphical interface that allows the user to interact with the program. The user can introduce alterations in the schedules of the trains. These alterations can correspond to conflicts or to possible solutions. It is also possible to ask for a graph with the execution of the train schedules, which give a global perspective of the situation. As a final result, the simulator produces a list of the detected conflicts.

The idea of the developed application, in the scope of the resolution of the initial problem, is to be used by a railroad traffic manager that, allying its experience to the conflicts forecasting capabilities of the simulator, producing more complete and efficient solutions for the conflict resolution.

Keywords— Conflict, decision support system, low traffic line, single line, train schedule, meet and pass

I. INTRODUCTION

Despite the fact that rail roads were created more than 150 years ago, they continued to have an enormous importance in the sector of transportation; it is still a privileged way for the transport of merchandises and passengers. The inherent management problems of all the process go since the creation of the necessaries infrastructures to the passengers flow control in the stations and trains.

The problem that will be treated in the current paper, is trains schedules management, particularly the Meet and Pass problem. Meet occurs always when there are two trains in the same line, travelling in opposite directions and that they will have to cross each other sometime. Pass corresponds to the situation when there are two trains, in the same line, travelling in the same direction, and that the train that follows behind, is faster and will have to pass over the first one.

A pre-established schedule has in consideration the meet and passes points. The problem is when some unpredictable change happens. These changes are typically delays at the departure and arrival times, but can also come from problems in the tracks or train malfunction during their course. The occurrence of any of the problems will make that the original pre-established schedule stops assuring the absence of conflicts.

So, the problem is being able to create changes in the original schedule, when a conflict is detected, that will result in a modified schedule free of conflicts.

The idea of this thesis is the creation of a support system that allows at first, a tool to aid the creation of schedule changes and in the future, proposes automatically the best solution for the conflicts resolution.

The solution was design to solve low traffic lines conflicts, particularly in single lines. Since between two stations, all the trains have to use the same line, the conflicts occurrence when there are delays, increases immensely.
II. PROBLEMS TO SOLVE

As already was indicated, the existence of only one line demands a great coordination of the trains in order to prevent the creation of conflicts.

Before talking about the possible conflicts, it's necessary to speak about the considered model of the line. As shown in Fig. 1, the linking between two stations is called Troço, and a Troço is composed by one or more Cantões and in the case of single lines, for one line only. Each Cantão is a measure of security for the distance between two trains that is, in each Cantão only one Train can circulate.

We can have two types of delays, primary delays and secondary delays. Primary delays are the first delays to be detected and they are usually unpredictable, a secondary delay is a consequence of a primary delay and can be planned.

When a train operator changes a schedule, he usually inserts delays. Although the circumstances are different, a change introduced by an operator or a changed forced by a conflict, have the same problem, they can originate more conflicts.

Like was said at parte I, the meet and pass problem suggests the two most important types of conflicts that can occur, the train entrance in a busy line with one or more trains travelling in opposing directions and the entrance in a busy Cantão by a train with the same direction. There is also the conflict when a train tries to enter a full station.

The solution for these problems must be a system that suggests schedule changes, respecting all the restrictions and with the ability to foresee the consequences of the inserted alterations. When comparing different solution proposals, it will be necessary to create a cost function.

III. EXISTING SOLUTIONS

The problem of the conflict management in most cases, is solved through one or more operators that, when confronted with a situation that can generate a conflict, look for a usual alteration to corrects the problem, and wait for the following conflict… This method has the advantage that it uses the personal experience of the operator with a moderate success however, the lack of human capacity to foresee bigger complexities will make that the chosen solution, probably, originates new conflicts.

The solution is typically composed by the introduction of delays in one or more trains. With merchandises, there is also the possibility to advance a schedule. These options come essentially by the fact that with passenger’s transportation, it must be guaranteed that the customers will
always catch the train if they arrive at least at train’s departure hour. This means that a passenger’s train can not leave earlier than his departure time. There is another type of solutions based in changes in the course of the train, rerouting, or changes in the trains order, reordering.

In the scientific literature there are basically two types of studies, off-line or clean-slate solutions and on-line or incremental ones, [1, 2]. An off-line solution is not used in real time, typically they are algorithms that create a schedule for every train from scratch. The on-line solutions are those that will typically be used by an operator in real time. The on-line solutions have strong time restrictions due to the necessity of solving the conflicts in the shortest time possible.

The answer to the problem in hands is an on-line solution. In this area there are several studies. Pranzo, D’Ariano and Pacciarelli, [2], developed a solution based in the Job-Shop problem where, each job is a train a each block is a Cantão, resulting in an enormous graph. They then use heuristics to come with a solution. The solutions obtained is strongly based in the rerouting of the trains.

Also from D’Ariano, [3], there is a study based in graphs, with a fixed velocity model. The Branch-and-Bound method is used to find the final solution.

Higgins, Ferreira and Kozan, [4], presented a model for low traffic single lines where there are defined priorities for each train. Using the Branch-and-Bound method, it decides which train will wait at the nearest crossing point.

Ahuja, [1], defined a method based in graphs that can give off-line and on-lien solutions. They are based in rerouting and reordering, trying to find the fastest courses.

Missikoff, [5], presents n object-oriented approach, base in real-time monitoring of lines and trains. When it detects or predicts a conflict, it decides if one of the trains will travel slower or make that one of them stops in a Station to be crossed or bypassed.

One last reference is the LOCROPOL project, [6], to be used with low traffic lines, based in the utilization of satellites and sensors on the trains, so that it knows, at any time, where the trains are.

IV. PROPOSED SOLUTION

The proposed solution consists in a system to be used with low traffic single lines and by an operator. This system would permit, from a given schedule, to make changes in the schedule and predict if those changes will solve all the conflicts or create new ones. It will also produce several solutions, and using a cost functions, suggests which the best solution is.

In this thesis is developed the application that will predict conflicts from a given schedule. It does not intend to solve the problem of conflicts managements, but create a tool that in conjunction with the expertise of an operator, produce simpler, faster and more efficient solutions.

This tool is composed by a discrete events simulator that given: the description of the line and the description of the trains and corresponding schedules, can foresee the occurrence of conflicts in the supplied schedule. The developed application has also a graphical interface, that allows making alterations in the introduced schedules. This means that with this tool the operator can experiment possible solutions for the resolution of one or more conflicts and get richer information on the consequences of the introduced alterations (see Fig. 2).

Comparing to the routing and reordering techniques, the created application is easier to implement because it doesn’t change what is actually done, it improves it.

Nevertheless, the application can be used with some of the existing algorithms, working as a test method for the solutions that these algorithms will produce.

V. DEVELOPED APPLICATION

The created application was developed using the oriented-base language, Java, [7, 8, 9].

The application contains a simulator, a plot generator and an interface to interact with an operator.
The interface allows the operator to introduce the necessary data for the simulation; it allows modification on these same data of entrance, not only to introduce conflicts but also to introduce alterations that can solve conflicts.

The interface gives also the possibility of showing a graphical (plot) representation of the Schedules loaded.

The simulator functions by discrete events, it receives a description of the Lines, the trains, and its Schedules, and then simulates the schedule of the various trains. At the same time it registers a log with all the detected conflicts.

When the schedules are loaded, it is created one event of departure for each train. During the simulation, whenever a train arrives departures, it’s created an arrival event to the next Station, whenever a train arrives and has still some schedules to do, it’s created a departure event. When there are no more events, the simulation ends.

The plot generator consists in a tool to help in the interpretation of the log. The received log is presented in the Interface through a list of the foreseen conflicts. The operator has then the possibility to call the plot generator, getting a more global representation of the simulation results. Before making the plot generation it is possible to add some customization, like plot only the trains related with.

VI. TECHNOLOGY

As said before, to develop the application, it was used an object-oriented programming language, Java.

Instead of using a language oriented to the functionalities, for example the C, where any update pass obligatorily by the alteration of the structure, a much more complicated and fastidious process. The object-oriented languages possess some characteristics that make these operations a lot more intuitive.

It’s programming models the real world. Everything in the world can be modelled as an object. For example, a train is an object, a station is an object, and a file is an object. Everything can be perceived as an object. Java is object-oriented because programming in Java is centred on creating objects; manipulating objects, and making objects work together Java provides encapsulation, Polymorphism, Inheritance and the possibility of creating graphical interfaces. These aspects make Java a lot more extensible, in the future if there’s a necessity to modify or to add something, the alterations to introduce in the code are much smaller and intuitive.

The only disadvantage face lower level programming languages is the necessity to adopt a programming concept a little more complex and the fact that normally, the performance is a little inferior.
VI. Final Conclusions

There will always exist problems during the day-by-day of any kind of railroad, or even any kind of transportation system. Most of the existing challenges in railroad traffic management come from activities that depend on the coordination of a lot of entities, since the load and discharge of merchandises and passengers management until the allocation of trains in a line. The railroad traffic management of low traffic lines is only one of the many existing problems but an important one.

The present thesis constitutes, particularly in Portugal, a step for an improvement of the actual existing services. The developed application does not solve the problems by itself but, it supplies a strong starting point to the creation of a more professional and complete tool.

The creation of the simulator is part of a larger idea, an idea that passes by the creation of a conflict resolution model and ends in a automatic system that not only detects conflicts, but also choose and implement the best solution possible.

Having such solution for conflict resolution, the project can go even further and reorganize the process, in order to allow the applications of such tools in medium and high traffic lines, where the complexity and density of the existing lines is much bigger.

References