A fuzzy knowledge-based system for railway traffic control

Alexander Fay

ABB Corporate Research Center, Automation and Information Technology Department, Speyerer Straße 4, 69115, Heidelberg, Germany

Abstract

Modern train traffic systems have to fulfill high requirements on service reliability and availability. This becomes especially important with competitive transport markets. Train operators can only meet these requirements by quickly developing an efficient action in case of traffic disturbances. This paper describes a dispatching support system for use in railway operation control systems. It also contains expert knowledge in fuzzy rules of the “IF-THEN” type. Various methods have been proposed for the representation of this kind of knowledge and for reasoning on this base. Expert systems can gain significant success by incorporating fuzzy knowledge and a graphical means of description. The paper describes a Fuzzy Petri Net notion that combines the graphical power of Petri Nets and the capabilities of Fuzzy Sets to model rule-based expert knowledge in a decision support system. Using this approach, a knowledge base is easy to design, analyze, test, enhance, and maintain. An assistant system for train traffic control is presented, and the advantages of this Fuzzy Petri Net notion are shown in the context of application in train traffic control decision support.

Keywords: Expert systems; Knowledge representation; Fuzzy rule bases; Petri Nets; Railway applications

1. Introduction

1.1. Train operation and control

Public transport providers face increasing demands concerning

1. competitive transport markets,
2. need for effective and efficient use of resources,
3. reduction of personnel for train operation and control,
4. changing and increasing customer requirements,
5. service reliability and availability.

High transport capacity should be obtained and maintained, and still the transport system has to be attractive for passengers. Irregular services and especially train delays or cancelations are severe obstacles in achieving these aims. The challenge of increasing train speeds, tight time schedules and higher traveller demands force train operators to improve the punctuality and reliability of their train services. Train operators can only meet these requirements by quickly developing an efficient action in case of traffic disturbances (deviations from schedule, resource lacks or the like). The use of computer-aided systems in both planning and operations control of train traffic is becoming more and more important to augment effectiveness and efficiency. Possible areas of application range from strategic planning to operational monitoring and control.

Concerning train operation and control, it is the dispatchers’ task to ensure optimal train traffic performance according to the schedule and to minimize the impacts of schedule deviations even in the presence of unforeseeable disturbances. The latter might be due to

1. planning mistakes (like poorly calculated headways or service times),
2. technical reasons (engine breakdowns, signaling failures, track closures, and the like),
3. organizational problems (late or absent staff members, extra trains for urgent transportation demands).

These disturbances can cause traffic conflicts like:

1. connection conflicts for passengers,
2. resource conflicts for trains and personnel,
3. availability conflicts for track sections, and
4. delays (usually in combination with one of the above).

In general, traffic conflicts result in situations where not all of the technical and operational requirements can be met. A conflict can be resolved by relaxing some of the requirements. Only operational requirements, such as arrival times or predefined platforms, can be relaxed, whereas technical requirements, such as maximum speed restrictions, can not be changed.

Possible dispatching actions might include
1. prolonged or additional stops at stations,
2. crossing and overtaking,
3. shifts and detours,
4. canceled or added trains.

To ensure an optimal dispatching process in an environment of crowded tracks and reduced personnel, the dispatcher has to be supported by improved tools which focus on the most important conflicts, and present all the necessary information, and offer effective solutions.

1.2. Algorithmic and heuristic approaches

Regarding the development of train traffic control, two main trends can be observed:

1. To optimize train traffic systems in the sense of an optimal use of resources, more and more sophisticated methods and algorithms are used in traffic planning. Railway planning and control have a similar problem structure (as explained in Fay and Schnieder (1997)), so there is desire to make use of them also for short-term problems such as traffic control. The difficulty in transferring the effective planning methods results from the shorter calculation time available: most algorithms developed for planning tasks are too time-consuming to yield immediate answers to traffic control problems. Therefore, heuristics have to be found and employed, not to replace the algorithms, but to adapt them to reduce the solution space significantly. These heuristics can be deduced explicitly from the human dispatchers’ knowledge (see Section 2).

2. To further improve traffic safety, quality and profitability, most functions on the level of train, track and signaling control have already been automated. This development is to be repeated in the area of traffic operation and control for the same reasons. This is much more difficult because the technical and organizational processes are less formalized on this level. Computer-aided traffic analysis and computer-aided development of conflict resolution proposals offer valuable support for this purpose.

The numerous approaches in international research and application to tackle train traffic planning and control can be divided into two main branches:

1. algorithmic optimization approaches, which stem from mathematics, classical computer science and operations research and can find the global optimum with respect to the goal function chosen,
2. heuristic approaches, which try to find good (not necessarily optimal) solutions fast by employing AI methods and expert knowledge.

These approaches are not necessarily mutually exclusive but can be combined in a useful manner.

Examples of algorithmic optimization approaches for real-time tasks are
1. a decision support system for routing trains through railway stations (Zwaneveld et al., 1996),
2. train scheduling on a single-track railway (Nachtigall, 1996),
3. optimal regulation for metro lines (Fernandez et al., 1996).

Most optimization approaches stemming from operations research show NP-complexity and therefore cannot be used for practical applications. They can be transformed to algorithms with polynomial time complexity, but only by making severe simplifications of the real problem. With increasing computer power, this problem will be relaxed. However, within the next few years, it will still not be possible to treat these NP-complex problems within the time frame of about 15 s allowed for online optimization. Hence, the search space has to be reduced significantly to achieve implementable results. This reduction can be performed with sophisticated search strategies or heuristics. Heuristics, which include knowledge of dispatching experts, can lead to straight-forward dispatching decisions. Examples employing heuristics for dispatching support are

1. ESTRAC-III, an expert system for train traffic control (Komaya and Fukuda, 1989),
2. PETRUS, an underground railway traffic expert system (Moirano et al., 1989),
3. an expert system for public transport control (Adamski, 1993),
4. an expert system for real-time train dispatching (Schäfer and Pferdmenges, 1994).

The system developed by the authors (described in Section 2) is also based on heuristics in the form of expert rules. The rules derived from expert knowledge are of paramount importance.
Yet heuristics alone will not necessarily always yield satisfactory solutions: the knowledge base is always limited, and not all situations can be covered in advance by appropriate rules. On the other hand, dispatching actions can have large impacts on traffic behavior and service quality. Therefore, it is desirable to estimate these effects in advance before implementing a dispatching strategy. For this purpose, simulation plays an important role. Though it is difficult to build an appropriate model of the traffic system, the most important features can be captured in simulation models, which then offer inexpensive and riskless means for evaluating dispatching alternatives.

Algorithmic and heuristic approaches are not mutually exclusive: they both have their merits for particular problems. Therefore, it is most beneficial to combine them. While the complete problem formulation for an algorithmic optimization is not suitable (as described above), specific algorithms exist for the solution of specific sub-problems. For example, the system developed by Zwaneveld et al. (1996) can be used for calculating the optimal path for a train through a railway station, and by means of the algorithm described by Nachtigall (1996), optimal crossing points for trains on a single track can be determined.

Therefore, it seems most suitable to determine the overall dispatching decision by means of heuristics (e.g.: “In case of a signal failure, should the trains (a) be canceled, (b) take a detour, (c) wait for the problem to be solved?”). The overall decision is then to be refined by algorithms (e.g., in case (b): “Where should the trains cross each other on the remaining track?” or in case (c): “At which platform should the trains wait to minimize the disturbance to the remaining traffic?”).

Thus, algorithms have their role in railway dispatching where they can contribute to system improvement. In the system developed by the author, they can be employed in the way described above: to optimize the solution of sub-problems. In Fig. 1, their input is taken into account together with the “hard constraints” for the “selection of feasible actions” (see Fig. 1).

1.3. Decision support systems based on fuzzy knowledge

The ever increasing complexity of large-scale technical systems (such as manufacturing sites, power plants or traffic systems) makes new approaches for their effective and efficient operation and control inevitable. Control — in a wide sense — denotes an appropriate reaction in the case of a malfunction or failure to regain a regular process flow. For this kind of “conflict management”, the usage of available expert knowledge about the process and its problems is of utmost importance. The key to a successful conflict management is to support the human decision-maker with modern information systems which do not only provide information but also utilize previously gained knowledge. Decision support systems (DSS) yield an enormous potential as they develop — on the basis of appropriately stored problem solution knowledge — solutions specially adapted for the current conflict situations. These solutions form a proposal for the human operator, who — in contrast to fully automated systems — is still in charge of deciding whether to follow this advice or not.

For most technical systems, much problem solution knowledge exists. Usually, this knowledge is not explicitly available, but exists as the experience of experts who have dealt with these problems for a long time. By the use of sophisticated knowledge acquisition and knowledge representation techniques, this valuable know-how can be gathered and utilized.

The core of the DSS is the rule base. It contains the relevant dispatching expert knowledge. Preliminary investigations on the basis of related existing research (Komaya and Fukuda, 1989) and on dispatching field studies resulted in the insight that the dispatching knowledge is mainly coded in rules, with a few special situations added as case-based knowledge. A rule-based implementation in a computer system can consequently represent most of the knowledge in a manner very close to the original (Puppe, 1993). This is a crucial prerequisite for the latter system to produce results which are traceable and understandable by the
dispatchers, and this is essential for the acceptance of the system.

Both the description of a conflict situation and the formulation of expert knowledge regarding appropriate actions for this conflict are frequently vague. Hence, the success of knowledge-based systems is mainly dependent on how this knowledge could be modeled, explicitly taking into consideration these types of vagueness. Fuzzy concepts have emerged as a suitable means to be applied here (Zimmermann, 1993). The fuzziness is due to two sources:

1. The conditions which have to be fulfilled for the application of a certain rule can only be specified imprecisely by the experts. Typical examples are “When the train is much delayed, . . .” or “When the connection is important for many passengers, . . .”. These formulations elude an exact fixation. Precise limits or intervals cannot be given, nor are they necessary for the dispatchers’ everyday work.

2. The conditions which have to be fulfilled for the application of a certain rule are not available precisely. Instead, rough numbers are given. So — to stay with the example used above — the number of passengers who want to catch a certain train is not recorded exactly but estimated by the conductor.

Both kinds of fuzziness have to be taken into account during knowledge modeling. Artificial boundaries or exactness must not be introduced, but the fuzziness has to be modeled explicitly to make the best possible use of the expert knowledge.

2. The train traffic control assistant

2.1. System structure and functioning

In this section, a dispatching support system is described which has been developed at the author’s institute (Fay and Schnieder, 1997). The system is considered for use in railway operation control centers and comprises of a knowledge-based decision support system, a simulation tool and a graphical user interface. The assistance provided by the system consists of

1. simulation of the traffic development in the near future (approximately 1 h),
2. detection of conflicts,
3. display of relevant information,
4. prediction of certain dispatching measures’ impacts,
5. proposal of adequate dispatching actions based on accumulated expert knowledge.

The rest of this paper focuses on the last point: the development of dispatching actions. The other features have been described in detail in Fay and Schnieder (1997), for example.

The complexity of the problems to be dealt with in real-time traffic control makes the employment of sophisticated expert knowledge indispensable to guide the search for appropriate dispatching actions. Therefore, dispatching experts’ knowledge has been acquired, and a rule base has been derived from this knowledge which can be used for the automatic development of dispatching possibilities.

On this basis, the expert system tries to develop a solution for the actual traffic conflict (Fig. 1). The knowledge base is scanned for rules which are suitable for tackling or solving the actual conflict with regard to overall traffic objectives and strategies. By application of the appropriate rules to the conflict situation, a set of promising dispatching actions is derived. They are checked for fulfilling the hard constraints (e.g., for overtaking, the parallel track has to be long enough). Only the measures that fulfill all hard constraints are further considered.

To predict the probable effects of each measure, they are applied to the actual traffic situation in parallel simulation runs. The emerging scenarios are evaluated automatically with regard to different quality criteria, taking into account the various desires of all persons and organizations involved in the traffic process. These criteria might be operational ones like punctuality or smoothness of traffic flow, or economical criteria like minimization of costs and optimal use of resources or maximal customer satisfaction.

If the measures seem to be inadequate to solve the conflict, or if additional conflicts arise during simulation, the selection and assessment of dispatching measures is repeated iteratively for further improvement.

The dispatching measures which result in the best conflict solutions are proposed to the dispatcher, together with explanations of how they were achieved, to what degree they contribute to the solution of the problem and the general traffic optimization, and what other solutions may exist. The dispatcher can either follow the advice and accept the proposal, modify it or try out his own solutions. In any case, the dispatcher has still the final say over the executed dispatching measures.

2.2. System implementation

The crucial steps in the development of the knowledge-based component (as of any expert system (Puppe, 1993)) are:

1. the knowledge acquisition process,
2. the appropriate modeling of the (fuzzy) knowledge,
3. the structuring of the rule base for evaluation purposes.

Therefore, these steps have been carried out to prove the feasibility of our approach and are described in the
remainder of this section to some extent. This is followed by an example in Section 3.

2.2.1. Knowledge acquisition

Knowledge has been acquired at the dispatchers’ control centers during several sessions. The first sessions served to get acquainted with the domain and the terminology used. In the following sessions, a combination of interviews and open questions was used to gradually build up the rule base. Thus, the elicited knowledge could be worked upon and structured in between, and gaps could be closed and ambiguities and contradictions could be solved in the following session. In total, it took about five sessions (each of these lasted about 2 h) to elicit the knowledge.

Several times during knowledge acquisition, conditions for the fulfillment of a rule could only be expressed vaguely by the experts, e.g., “When the delay of the train is large...” or “When the connection is important for many travelers...”. In these cases it was tried to gather additional information from the experts to model these conditions by means of fuzzy membership functions (see Section 2.2.2).

Altogether, about 100 rules have been accumulated this way, some of them have been further refined later. The rules have been sorted according to several conflict classes. Each rule is based on two to eight conditions, which partly overlap in each conflict class. Some rules are structured hierarchically. During this procedure, even a small set of rules (three to ten rules) in one conflict class can become difficult to handle. Therefore, a graphical means of description of the rule base structure and evaluation has been employed (see Section 2.2.3).

2.2.2. Modeling of fuzzy knowledge

The knowledge has been modeled in fuzzy rules to respect the vagueness of the knowledge and the uncertainty of the fulfillment of the conditions of the rules.

In a rule-based form, a rule $R_{x}$ looks like:

$$R_{x}: \text{IF } b_{i} \text{ THEN } b_{j} \text{ (CF = } c_{x}\text{)} \text{,}$$

where $b_{i}, b_{j} \in [0,1]$ are fuzzy values which are assigned to the conditions represented by the respective places. CF is the credibility factor of this rule. Every transition represents the corresponding rule of the rule base. Condition $i$ is only fulfilled to the degree $b_{i}$, which is a fuzzy value between zero and unity. For example, condition $i$ could represent “small delay”. Every delay belongs more or less to the concept of a “small delay”. According to the statements of our application domain’s experts, the membership function of “small delay” is modeled as shown in Fig. 2.

Likewise, the membership functions for all conditions relevant to the problem in question are determined. The more the situation in question equals the precondition expressed in rule $R_{x}$, the larger is $b_{j}$. The value $c_{x} \in [0,1]$ represents the degree of credibility (or truth) of the rule $R_{x}$. A stronger belief in the rule corresponds to a larger $c_{x}$.

In case of $b_{i}$ and $c_{x}$ being given, $b_{j}$ can be calculated. According to Chen et al. (1990) and Chun and Bien (1993), $b_{j}$ is calculated by

$$b_{j} = b_{i} \times c_{x}.$$

Thus, the resulting degree of fulfillment of the right-hand side of the rule is both proportional to the degree of fulfillment of the condition and to the credibility of the rule.

In many cases, a rule contains more than one precondition:

$$R_{x}: \text{IF } b_{i} \text{ AND } b_{k} \text{ THEN } b_{j} \text{ (CF = } c_{x}\text{)} \text{.}$$

The preconditions $b_{i}$ and $b_{k}$ have to be combined by an appropriate operator. In contrast to Chen et al. (1990), who make use of the classical MIN-operator for AND, the arithmetic mean operator

$$b_{j} = \frac{1}{2}(b_{i} + b_{k})$$

is employed in this case, as proposed by Zimmermann (1993). Thus, advantages and disadvantages can be weighed against each other to avoid the loss of an otherwise very good solution with a weak support by only one precondition.

Furthermore, a suitable operator $f()$ has to be chosen that combines the contributions of several rules to the same decision:

$$R_{x}: \text{IF } (\ldots) \text{ THEN } b_{j(x)} \text{,}$$

$$R_{y}: \text{IF } (\ldots) \text{ THEN } b_{j(y)} \text{,}$$

$$b_{j} = f(b_{j(x)}, b_{j(y)}) \text{.}$$

The operator should reflect that the support $b_{j}$ for such a decision might be higher than each single support by one rule, at least when the rules are independent. This implements the statements: “The more (independent) arguments one has, the better founded is his/her opinion” and “Repeating the same argument does not increase the soundness of the conclusion”. Therefore, the operator proposed by Tano et al. (1995) is used:

\[ f(a, b) = \frac{a + b}{2} \]
\[ b_j = t_1 \times \text{MAX} (b_{j(x)}, b_{j(y)}) + t_2 \times (b_{j(x)} + b_{j(y)} - b_{j(x)} \times b_{j(y)}) \]

with \( t_1, t_2 \in [0, 1] \), \( t_1 + t_2 = 1 \),

which yields results for the combination of \( a \) and \( b \) between \( \text{MAX}(a, b) \) and \( a + b - ab \), according to the degree of reinforcement of the contributing rules. The parameters \( t_1 \) and \( t_2 \) have to be adjusted by the expert with the knowledge according to the rules in question.

2.2.3. Graphical representation of rules in a Fuzzy Petri Net

As pointed out above, the rule base consists of about 100 rules. Each of the rules combines two to eight conditions. The conditions are usually made use of by several rules. The chain of rules — from the initial conditions to the final conclusion — comprises of up to eight rules to be evaluated subsequently. Up to six rules contribute to one dispatching decision. Therefore, it is difficult to build up and maintain such a rule base and to keep track of the evaluation of the rules during the employment of the rule base in a decision support system. Even a small set of rules (three to ten) in one conflict class can become difficult to handle. To keep the rule base clear, however, is very important for several reasons:

1. During rule base compilation, errors can be found more easily.
2. During the solution of an actual conflict, the user can trace the rule evaluation — an important feature for the user’s acceptance of the system.
3. Parallel to the practical use of the system, the user can maintain and enhance the rule base by adding, modifying, and deleting rules to adjust the system to his singular aims and ideas, without the assistance of a knowledge engineer.

To enhance the clarity of the rule base, the rules are represented in a Fuzzy Petri Net (Looney, 1988). In a Fuzzy Petri Nets representation, each transition of the Petri Net equals one rule of the rule base. Fig. 3 shows the FPN representation of rule \( R_x \) before (Fig. 3a) and after (Fig. 3b) the reasoning process took place.

Fuzzy Petri Nets offer both a graphical representation and a sound mathematical background, based on analysis and graph theory. Thus, relations, redundancies, and contradictions can be found in the rule base, and the rule base execution can be traced visually. While the latter is explained in the example in Section 3, the former can be stressed by the following facts:

- A Fuzzy Petri Net Reachability Graph was developed which allows assessment of the influence of different conditions on the final decision. This greatly helped to adapt the rule base for the application domain.
- A circular conclusion, which was not noticed during several weeks of editing the rule base and which could have led to an infinite loop in the rule engine, was found by employing the mathematical strength of the Fuzzy Petri Net approach.

The Fuzzy Petri Net concept is described in detail in Fay and Schnieder (1999). In Section 3, the FPN concept described above is used in the frame of the decision-making problem in on-line train traffic control.

3. Implementation

3.1. Implementation overview

The crucial aspects of the decision support system are:

1. Is it possible to acquire the necessary knowledge in a reasonable amount of time?
2. Is it possible to model the knowledge in an appropriate way (appropriate for easy editing, testing, maintaining, and using (evaluating))?
3. Is it possible to have a simulation tool that is fast enough to produce results in a few seconds (compare Fig. 1: the simulation is on the “critical path” of decision-making)?
4. Does the rule-based system provide optimal or near-optimal results?

Regarding question 1: As described in Section 2.2.1, the knowledge acquisition could be completed within about 10 h of the experts’ time. The separation of the knowledge from the technical constraints proved to be advantageous: while the constraints (e.g., the track parameters) differ between the experts’ locations and work areas, the knowledge is very generic. This allows to combine the experiences of several experienced experts within the rule base. The knowledge is universal and can be applied to every dispatching domain within German Railways (and most probably to other high-density mixed-mode railway systems as well). The constraints have to be adapted for each workspace. However, this has to be done only once since the constraints change little over time. Thus, the separation of knowledge and constraints eases maintenance and improves the power of the knowledge base.
Regarding question 2: As described in Section 2.2.3, the Fuzzy Petri Net approach proved to ease both establishing and testing (off-line) and evaluating and maintaining (on-line) of the rule base. Because commercial Petri Net tools are not able to cover Fuzzy Petri Nets appropriately, a special tool has been developed (based on Visual Basic, the necessary effort amounted to one person/month only) that allows to edit and execute Fuzzy Petri Nets. The experiments described in Sections 3.2 and 3.3 are carried out on the basis of this tool.

Regarding question 3: Commercial railway simulation tools utilize a fine-grained model of the railway network under consideration and usually employ a fixed time-step simulation. These tools have response times of several minutes to calculate the journey of one train, which has orders of magnitude slower than needed. Therefore, a simulation tool has been developed and implemented which is based on object-oriented technology. This simulation tool employs pre-calculated acceleration and braking curves wherever possible, thus reducing the simulation time significantly with only minor deviations from the exact solution. Details on the simulation tool can be found in Fay and Schnieder (1997).

Regarding question 4: An extensive evaluation of the decisions given by the rule base has been carried out. For this purpose, typical scenarios have been considered, and the decisions of the rule base have been compared with simulations of all possible dispatching actions. The latter have been rated according to a utility function. This utility function takes into account:

- the delay of all affected trains, weighed according to the rank of the train,
- the summed delay for all passengers,
- the costs of additional resources involved, such as additional trains, staff, and track fees.

In more than 90% of the scenarios, the rule base opted for the best possible dispatching action. Taking into account that the scenarios chosen were not trivial but rather border-line cases, it can be stated that the rule base finds the best dispatching action in more than 9 out of 10 cases.

In Sections 3.2 and 3.3, one of the tested scenarios is presented as an example.

3.2. Application example

In this section, an example of the system functioning in a typical scenario is given: in a certain station, a

![Fig. 4. Rule base for connection conflict (part 1).](image)
long-distance train ("no. 2") waits for another long-distance train ("no. 1") to arrive to allow passengers to transfer from train 1 to train 2. In case of train 1 being late, a decision has to be made whether train 2 should wait or rather depart. The concerns of the transfer passengers of train 1 have to be balanced against those of the passengers of train 2 (and the interests of passengers who intend to board train 2 during the rest of its trip), and an optimum has to be found regarding the divergent aims mentioned above.

The decision is influenced by the following conditions:

1. the length of the delay of train 1,
2. the number of transfer passengers,
3. the length of the further trip of train 2,
4. the time interval of trains in the direction of train 2,
5. and whether train 2 is the last train in this direction on this day (and, thus, the last chance for transfer passengers to reach their destination on this day).

The main aspects (as stated by dispatching experts) are:

1. The longer the delay, the stronger is the argument for letting train 2 depart according to schedule.
2. A high number of transfer passengers are in favor of letting train 2 wait.
3. The longer the further trip of train 2, the more effects has a delay of train 2 on other trains.
4. The shorter the time interval, the less the transfer passengers have to wait for another train.
5. If transfer passengers do not reach their destination on this day, additional costs have to be covered for customer care — a reason to let train 2 wait for train 1.

These aspects are covered in eight rules, which are shown in Figs. 4 and 5. (The display of the rule base is split into two figures for screenshot resolution purposes only. The user of the tool can zoom in to watch details and zoom out to get an overview of the rule base.) The number within a rule (transition, shown as a square) denotes the credibility factor of the rule. The number within a condition/action (place, shown as a circle) denotes the degree of fulfillment (of a condition) or the degree of support (for an action), respectively. The black token on the last (rightmost) places show that the evaluation (execution) of the rules has been completed up to the very end.
3.3. Comparison of simulation and rule-base results

Typical conditions for the scenario described above might be a delay of train 1 by 10 min, 25 transfer passengers, a headway (and thus a waiting time) of 20 min and a further trip length of 125 km. At first, these conditions (henceforth called “scenario A”) have to be transformed into degrees of fulfillment by use of membership functions. The membership functions for delay, further trip length, number of transfer passengers, and headway, are shown in Figs. 6–9, respectively. By a comparison of the conditions with the membership functions, fuzzy numbers are derived (as described in Section 2.2.3) which are inserted in the leftmost places in the rule base Fuzzy Petri Net (see Figs. 4 and 5). The stepwise evaluation of the rules yields a value of 0.84 for the alternative “train should wait” and a value of 0.81 for “train should depart on time”, i.e., the rule base suggests for train 1 to wait in this situation.

For the evaluation of the rule base’s proposal, both dispatching alternatives for scenario A have been simulated under realistic assumptions (concerning timetable margins, etc.). The emerging traffic development has been assessed with regard to the weighted sum of passenger delay minutes.

The conditions of scenario A, the costs of each alternative, and the proposal given by the rule base are grouped in the first row of Table 1. A comparison of the total costs reveals that waiting is indeed the better alternative in this case.

Similar to scenario A, further scenarios are examined both by the rule base and by simulation. The results are listed in Table 1.

In case of scenario B (equal to scenario A, but with a shorter headway), the rule base evaluation does not end up in a clear proposal. This is correct, since the costs do not significantly differ. In scenarios D, E, and F, train 2 has a shorter further trip length and waits instead (compare scenario B with D and C with E). In case of a longer trip (scenarios G and H), a departure on time is advisable (compare scenario A with G). The further scenarios I to N stress that the rule base proposes the alternative with less costs, even in case of higher costs (scenarios I and J) or significantly lower costs (scenarios K to N).

In the previous cases, train 2 has never been the last train on this day for transfer passengers. If in scenario K, for example, train 2 was the last train, the rule base would have proposed to wait (0.96 for “wait”, 0.87 for “depart”).

Similar to this example of a connection conflict, the rule base can deal with other types of conflicts and proposes appropriate and optimized dispatching actions.

4. Conclusion and future prospects

It has been shown that by systematically making use of the knowledge of experienced dispatching experts within a dispatching support system, traffic quality can be improved, and operation costs can be reduced.

The proposed assistant system for dispatching support presented in this paper can be integrated in an operating center. Due to its flexible and modular structure, the system is the core for the development of dispatching support systems for various public transport systems and can contribute to an
improvement in traffic performance, reliability, and customer satisfaction.

The approach described in this paper has recently been awarded the “Forschungspreis der Deutschen Bahn 1998” (German Railway Research Award 1998) for its contribution to the optimization of railway traffic and its practical relevance.

Acknowledgements

The system described in this paper has been developed during the author’s stay at the Institute of Automation and Control at the Technical University of Braunschweig. The author would like to thank the head of that institute, Eckehard Schnieder, for having provided the financial support for this work and for his comments.

References


Table 1

Results of the rule base evaluation for the connection conflict

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Delay (min)</th>
<th>Transfer passengers</th>
<th>Headway (min)</th>
<th>Further trip length (km)</th>
<th>Total costs “wait” (min)</th>
<th>Total costs “depart” (min)</th>
<th>Rule base proposal</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>25</td>
<td>20</td>
<td>125</td>
<td>596</td>
<td>700</td>
<td>wait: 0.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.81]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>undecided:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>depart: 0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.80]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.79]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.79]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.82]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.80]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.80]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.80]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.82]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>undecided:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>depart: 0.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.80]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[depart: 0.81]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>wait: 0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>[wait: 0.75]</td>
</tr>
</tbody>
</table>


Alexander Fay obtained a Diploma degree and a Ph.D in Electrical Engineering from Technical University of Braunschweig in 1995 and 1999, respectively. He is currently with the Corporate Research Center of Asea Brown Boveri in Heidelberg. His main research interests are automation and control of complex systems, especially with the use of knowledge-based systems.