

# Non-Discriminatory Automatic Registration of Knock-On Train Delays

Winnie Daamen · Rob M. P. Goverde ·  
Ingo A. Hansen

Published online: 22 November 2008

© The Author(s) 2008. This article is published with open access at Springerlink.com

**Abstract** Incidents and waiting for train connections are registered by dispatchers as sources of train delays, but route and headway conflicts are not always clearly recognized. Moreover, traffic management and route setting are the primary task of dispatchers and signallers, whilst monitoring and incident registration is not allowed to take up too much of their time. This paper describes a tool that automatically and without discrimination identifies route conflicts and the train numbers involved. It is based on standard train describer and infrastructure messages recorded on the Dutch railway network. The logic of these messages is captured in a coloured Petri net (CPN) model on which a prototype tool for route conflict identification and estimation of knock-on delay has been developed.

**Keywords** Train describers · Knock-on train delays · Dispatching · Blocking times · Petri nets

## 1 Introduction

The standard registration of train punctuality is based on an automatic registration of revealed differences between the arrival times of the scheduled and realised train trips at a number of stations in the network, while the threshold for recording delays is different from railway to railway. In the Netherlands, any train passing with at least 3 min. delay at in total 32 major stations distributed over the Dutch network is reported. These punctuality figures do not make a distinction between primary and knock-on delays. The latter, in particular, is necessary in order to get better insight into the causes of delays and the impact on other trains. Apart from this, the reported delays are not measured exactly: the measurement points are situated near signals at some distance from the stop location in the station. Hence, the delays are estimated

---

W. Daamen (✉) · R. M. P. Goverde · I. A. Hansen  
Department of Transport & Planning, Delft University of Technology, P.O. Box 5048, 2600 GA Delft,  
The Netherlands  
e-mail: w.daamen@tudelft.nl

on the basis of standard off-set times, which do not distinguish between different types of rolling stock, and are rounded up to full minutes.

European Directive 2001/14/EC (European Commission 2001) contains instructions about transparent infrastructure access conditions and obliges the infrastructure managers to prepare a network statement in order to allocate scarce capacity in a non-discriminatory manner. Performance contracts between the government and train operating companies—as in Great Britain and the Netherlands—specify some minimum required punctuality levels and include fines for poor performance. A prerequisite for the non-discriminatory assignment of delays and allocation of fines, however, is a clear and objective determination of the cause and responsibility for the train delays. The current punctuality registration systems do not distinguish between primary and knock-on delays and the existing manual incident registration of train dispatchers in the Netherlands is insufficient with respect to comprehensiveness, reliability and objectivity (Ministry of Transport, Public Works and Water Management 2004; Weeda 2006). The development of an advanced non-discriminatory and consistent train delay registration system is therefore a high priority.

Train delays can be derived by filtering existing track occupation and release records and comparison with scheduled through, arrival and departure times at reference points. The assignment of primary delays due to technical failures of infrastructure or rolling stock is possible by matching the information on train number, time and location of incident records from the corresponding maintenance management systems, while an unbiased estimation of knock-on delays due to hindrance by other trains is possible by means of constructing blocking time graphs according to the UIC capacity norm 406 (UIC 2004) based on actual train characteristics, and signalling and train detection data (Daamen et al. 2006). This paper describes a recently developed prototype software tool for route conflict identification and for automatic distinct assignment of primary and knock-on delays.

The paper is outlined as follows. Section 2 briefly explains the logic of train describers and the developed coloured Petri net model which captures this logic. The route conflict identification tool based on this coloured Petri net model is explained in Section 3, while Section 4 considers a case study as an example of the working of the tool. Section 5 explains the relationship between the identified route conflict and other sources of knock-on delays (connection at stations) and how this could lead to an effective incident registration system. Section 6 presents some conclusions and issues of future research.

## 2 Train describers

The key to automatic delay registration is the use of train describers which keep track of train positions based on train numbers and infrastructure messages received from the signalling and interlocking systems (Exer 1995). A train number usually steps from one position to another when the train passes a signal. The internal logic of a train describer requires signal, track section and switch information from which it derives the route to the next signal determining the next position. All received infrastructure messages and all generated train number events can be recorded chronologically in train describer records. These records can be used for offline

applications such as replaying, punctuality analysis, and infrastructure capacity analysis (Goverde 2005). The Dutch infrastructure manager ProRail acknowledged the potential of these train describer records which have therefore been centrally archived to be used for these applications since March 2000. This paper describes a new application that automatically derives route conflicts and resulting knock-on delays using enhanced train describer records (VTL-files) of the Dutch TNV train describer systems. These VTL-files contain route information in the train number step messages which combined with infrastructure configuration files enable a fully automatic matching of train numbers to state changes of infrastructure elements such as signals, track sections and switches.

The main research idea is to derive the subsequent realized blocking times of the signal blocks for all trains and use this information to detect conflicts (Pachl 2002). The application has been developed using object-oriented programming in Java. Objects are all infrastructure elements (signals, track sections, switches) and signal blocks. The program scans the chronological messages in the VTL-files and updates the state of the objects accordingly. For instance, the relevant attributes of a track section in a station layout are its status (occupied, free), last release time and last occupying train number. When a train number steps into a block at a restricted signal (signal at the end of the block is at danger) then a route conflict has been detected and the conflicting train number can be found in the critical section of the next route block.

The accuracy of the blocking time estimates depends on the available information. In station areas full information of route blocks may be available, i.e., the passage time of the approach signal and the release time of switches, crossings, and route sections in yards and stations. On most open tracks however only partial information is available: neither track-clear detection nor signal states are logged and so they must be estimated from train number steps.

A coloured Petri net (CPN) (Jensen 1996) has been used to describe the complex logical and temporal interactions between the train number steps and the various infrastructure elements and signal blocks which are logged chronologically in the VTL-files. Based on this model a prototype route conflict identification tool has been developed.

The VTL-files contain a chronological list of all received infrastructure messages and the train number events generated by the TNV-system itself. A recorded infrastructure message only consists of the date and time, the code of the infrastructure element (section, signal, switch), and the binary state transition (occupied/free, stop/proceed, left/right), but does not include a train number. Goverde (Goverde 2005; Goverde and Hansen 2000) developed a tool (TNV-Prepare) to match infrastructure messages to train numbers using the recorded train number steps over subsequent windows and a database of route objects within the windows. This approach has been extended with signalling logic to detect route conflicts and identify the conflicting train number. The logic of the infrastructure and train number events is described in the CPN of Fig. 1. This section explains the CPN in detail. For technical details concerning CPNs see Jensen (1996).

A CPN consists of two types of nodes called places and transitions which are represented by ovals and rectangles, respectively. The places correspond to states of objects of the specified colour shown at their right-bottom. The CPN has five

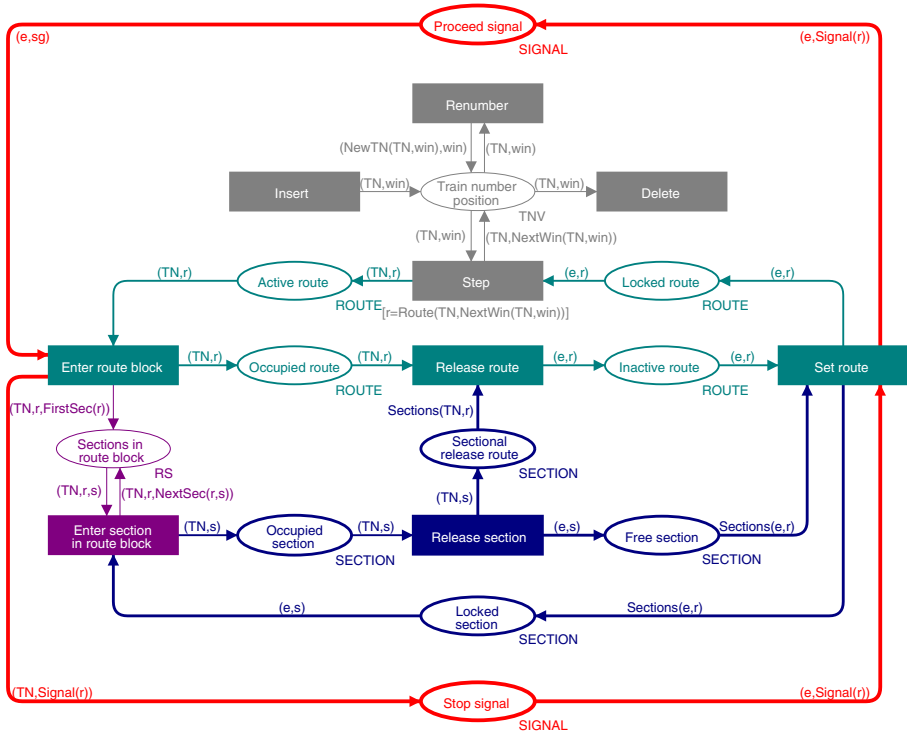


Fig. 1 CPN model of the train descriptor records

elementary colours (types) while the places have five colours consisting of combinations of the elementary colours, see Table 1. All place colours consist of a (physical or virtual) object and a train number. If a train number is not currently active in an object the empty train number variable *e* is used. Table 1 also contains a list of all variables used in the CPN. The transitions represent events causing state transitions of the objects, e.g., setting a route causes a Proceed signal, while a signal changes to a Stop aspect after a train enters the route block after the signal.

Table 1 Colour definitions and variables

Colour	Attributes	Variable
Train Number	list of train numbers	win
Window	list of window codes	TN, e
Route	list of route codes	r
Section	list of section codes	s
Signal	list of signal codes	sg
TNV	(train number, window)	
ROUTE	(train number, route)	
SECTION	(train number, section)	
RS	(train number, route, section)	
SIGNAL	(train number, signal)	

The positions of train numbers in windows are collected in the place called ‘Train number position’ with colour TNV. There are four train number messages corresponding to inserting a new train number in some window, renumbering a train number, stepping a train number into a next window, and deleting a train number. These messages correspond to transitions in the CPN called Insert, Renumber, Step, and Delete, respectively. Insert generates a new pair (TN,win). Renumber maps (TN, win) into (TN’,win) for some TN’, which is represented by the function  $NewTN(TN, win)$ . Delete just removes a pair (TN,win) from the CPN. Step is the most involved transition and puts a pair (TN,win) into a new position (TN,win’) using the function  $NextWin(TN,win)$ . The functions  $NewTN$  and  $NewWin$  correspond to the internal logic of the train describer and are not specified in the CPN description: the results are just revealed in the messages read from the VTL-files. The transition Step also reveals the route  $r$  which had been locked for train number TN within the next window. This is represented by the function  $Route(TN,win')$  with  $win'=NextWin(TN,win)$ . So a Step corresponds to the VTL-file message: train number step from win to win’ over route  $r$ . After this message it is known that route  $r$  is active and being occupied by train number TN.

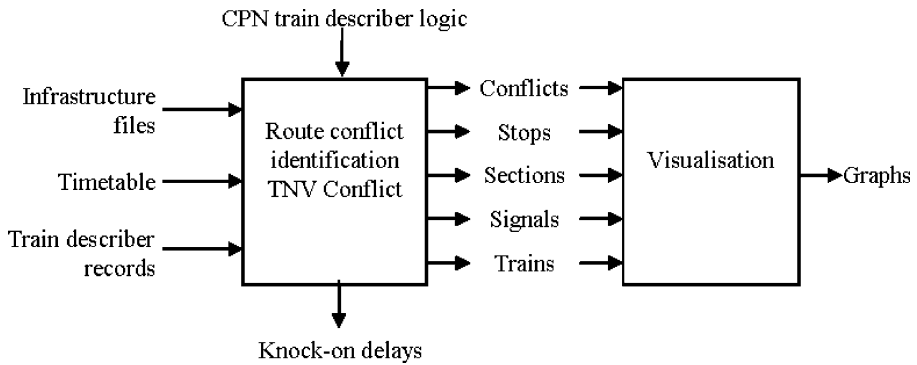
An active route immediately triggers the transition ‘Enter route block’ which causes three state changes. First the active route  $r$  changes to an ‘Occupied route’. Second, the proceed signal at the front of route  $r$ , denoted by  $Signal(r)$ , changes to a ‘Stop signal’. And third, the first section on route  $r$ , denoted by  $FirstSec(r)$ , changes to an ‘Occupied section’ via the RS place ‘Sections in route block’ and transition ‘Enter section in route block’. When the latter transition fires the next section on the route  $NextSec(r,s)$  is also put into the place ‘Sections in route block’. This transition then fires successively until the last section in the route has been occupied, in which case  $NextSec(r,s)$  is empty. Hence, the locked sections in a route  $r$  get occupied one by one. A train number TN is attached to all objects in route  $r$  fired by ‘Enter route block’ and to the sections fired by ‘Enter section in route block’.

An Occupied section changes to a Free section if ‘Release section’ is fired, and at the same time the released sections are collected in the place ‘Sectional release route’. If all sections in a route  $r$ , denoted by  $Sections(TN,r)$ , have been released the transition ‘Release route’ fires by which a route  $r$  becomes Inactive. If a route  $r$  is inactive, all its sections are free, and the signal at the front of the route is at danger, then the transition ‘Set route’ may fire, causing the route  $r$  to be Locked, all sections in the route  $Sections(s,r)$  to be Locked, and the signal  $Signal(r)$  to show a Proceed aspect. A locked route can again be used for a next train number Step.

The CPN description of the interlocking logic has been proved useful for deriving the functional design of the object-oriented Java program as considered in the next section. Moreover, it serves to explain the program to the railway staff.

### 3 The route conflict identification tool TNV conflict

This section describes the software tool TNV Conflict that identifies automatically knock-on delays due to route conflicts. Figure 2 shows the software tool as a black box specifying the required input and output. Also, the figure shows the relation



**Fig. 2** Black box representation of the route conflict identification software

between TNV Conflict and a dedicated visualisation tool that has been developed to demonstrate the results of TNV Conflict.

The input of the tool consists of infrastructure files, the timetable and train describer records. For each major station area, an infrastructure file is required, describing the basic link and node network of the rail infrastructure. Moreover, these files indicate the location of switches, track sections, signals, and virtual train describer windows. To determine the train stops and corresponding delays or early arrivals, the detailed timetable data is required as input, as well as the platform track sections and distances to each stop position. The train describer records consist of a chronological list of status changes of train numbers and infrastructure elements. At this moment, the input consists of offline train describer records, but the software is ready to read online train describer information.

Combining these information sources with the train describer logic of the CPN model described in the previous section enables the tool to automatically identify the knock-on delays of individual trains, as well as the corresponding hindering train and the hindrance time due to this train. Additional output consists of the (binary) status of sections ('Occupied' or 'Free') and signals ('Stop' or 'Proceed') over time, as well as statistics for realised stops.

The user interface of TNV Conflict shows the consecutive steps of the initialisation of the infrastructure, the timetable data and the stops, see Fig. 3. After this, the automatic identification of knock-on delays may start by reading the train describer files.

First, the infrastructure files are selected and read consecutively. During this reading process, all infrastructure objects are created. Figure 4 shows these infrastructure objects as well as their relations. Some of the information to relate the different infrastructure objects comes directly from the infrastructure files; other information, such as the signal at the end of a route, needs to be derived by the software itself.

Second, the files containing the timetable data are selected and read. Each file considers a single station area. The files contain the scheduled arrival times, departure times and through times for each train, indicated by its train number. Unfortunately, it does not indicate yet on which sections or platform a stop is planned. Since this information is essential to assign a longer occupation time on a section to a stop, to

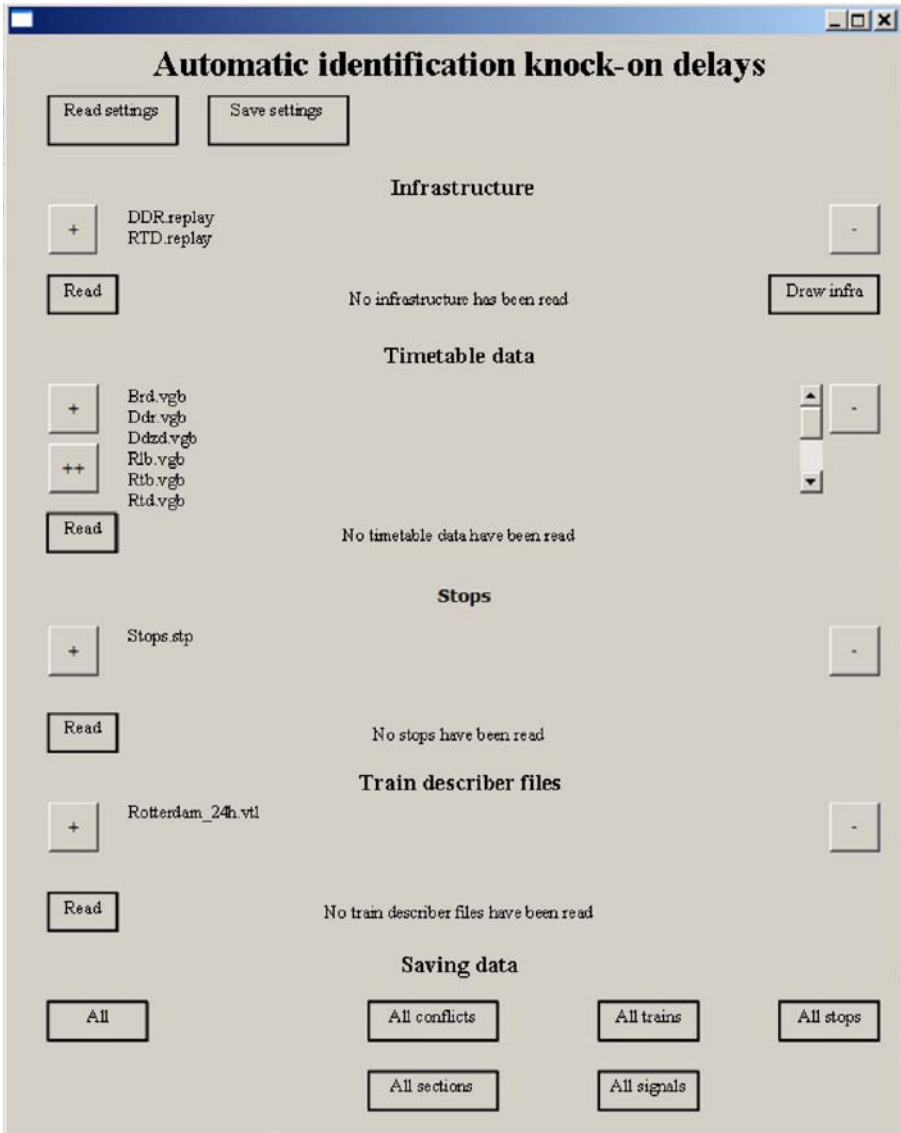


Fig. 3 User interface of the route conflict identification software (TNV Conflict)

date we have manually created the link between the stop sections and the station areas. The third step in the process is reading these so-called ‘stops files’.

The fourth and final step is the most important one, that is reading the train describer records line by line. As indicated before, the train describer records consist of chronological train describer messages and status messages received from the signalling and interlocking systems. By reading these messages one by one, the available information is similar to the information available online, demonstrating that the software is ready for online use. Six relevant messages can be identified: stepping of a train number into a next window (id=1), inserting a new train number

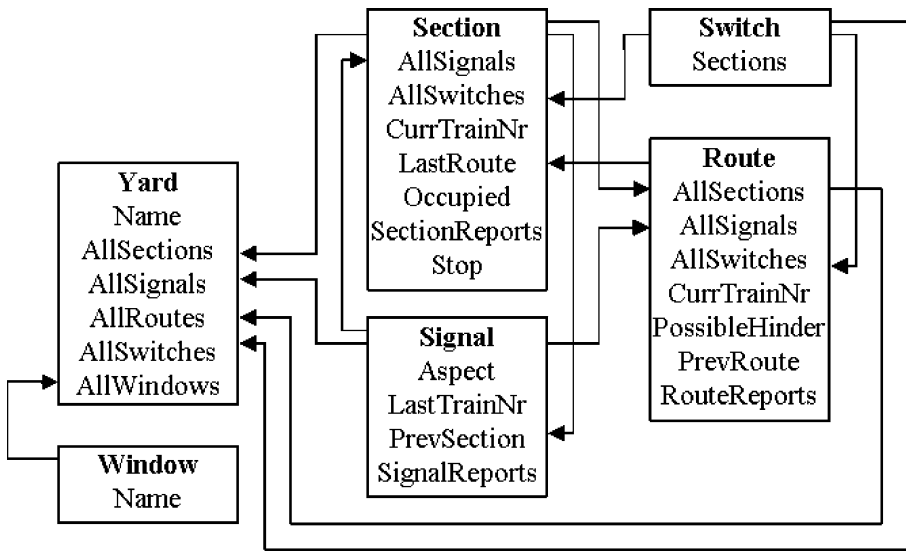


Fig. 4 Different objects describing the infrastructure

in a window (id=2), deleting a train number from a window (id=3), renumbering a train number (id=4) and the status messages of sections (id=5) and signals (id=6). Examples of these messages are given in Table 2 (the first two columns indicate the date and time of a recorded message; the third column indicates the type of message).

Type 1 messages are train number steps into the route block indicated in column 4. The train number is shown in column 5, whilst the origin and destination window are shown in column 6 and 7, respectively. When this message is read, the software creates an object called ‘WindowMovement’. The references to the corresponding Route and TrainNumber are set. All elements of the Route (sections and signals) are informed that this train number will cause the following status change (see CPN logic). To determine whether this train is hindered, we check whether the signal at the end of the route shows a ‘Proceed’ aspect. If this is not the case, the train will have to start braking and encounters a knock-on delay. The train that last passed the signal or occupied any of the sections of the new route is the train causing this knock-on delay. In case of hinder, all known characteristics are saved to a new object ‘Conflict’.

Table 2 Example messages in a train describer record (VTL-file)

Date	time	id				
1-12-05	8:50:44	1	RTDSBAA03400666	1929	4S.0	HAS.0
1-12-05	8:51:15	2	5131	RTD CFU.1		
1-12-05	8:52:54	3	2220	RTD DFU.0		
1-12-05	8:51:45	4	4229	4220	RTD 1BS.0	
1-12-05	8:51:30	5	RTD\$A666AT	B		
1-12-05	8:51:36	5	RTD\$337BT	V		
1-12-05	8:50:46	6	RTD\$340	S		
1-12-05	8:54:32	6	RTD\$636	G		



Type 2 messages indicate inserting a new train number into a window. Column 4 indicates the train number and column 5 the window. A train number insertion is generated manually by a dispatcher, automatically if a train enters from an adjacent traffic control area, or automatically after a detected train movement of an unknown train or a shunting train unit. The software acts similarly to reading a type 1 message, but also creates a new 'TrainNumber' object.

Type 3 messages indicate deleting a train number. The fourth column indicates the train number and the fifth column the window. A train number deletion is either generated manually by a dispatcher or automatically after a train enters an adjacent traffic control area. The corresponding actions are simple: after the required bookkeeping, the train is set to inactive.

Type 4 messages specify a train renumbering. Column 4 contains the old train number and column 5 the new one. The sixth column indicates the current window. A train renumbering occurs automatically for trains reversing at terminal stations, or when a dispatcher manually corrects a temporary train number that was given to an unknown train. The software looks for the old train number in the list of active trains. If such a train can be found, the train id is changed. Otherwise, a new train number is created.

Type 5 messages indicate the occupation of a section (named in column 4). A 'B' stands for occupied, whereas the 'V' indicates that a section has been released. The software looks for the corresponding section. The attribute 'SectionReports' saves the status of a section and the changes of this status over time. A section report therefore contains the time the section gets occupied and the time it is cleared, including the associated train number. If an appropriate section report does not yet exist, a new section report is created. This way, possible flaws in the data streams are dealt with.

Finally, type 6 messages indicate a signal aspect change for the signal named in column 4. 'S' indicates a 'Stop' aspect and 'G' a 'Proceed' aspect. Similar to the logic of the section report, the status changes of a signal are kept in a signal report. This report contains the time the signal aspect changes as well as the train causing this status change.

Line by line, all train describer records are read, during which trains step over windows, the status changes of sections and signals are maintained, and route conflicts are detected. When reaching the end, the user indicates which results are to be stored. If all data are stored, five files are created, containing the conflict characteristics, train characteristics, section status changes, signal status changes and stop characteristics. The files are input for the visualisation tool (the user interface is shown in Fig. 5), whose aim is to demonstrate the train dynamics using blocking time diagrams and to create insight. In the next section, a case study is presented, showing some results computed by the tool TNV Conflict and the corresponding diagrams.

## 4 Case study

We demonstrate route conflict identification with a case study in the area between Rotterdam and Dordrecht in the western part of the Netherlands. This part of the network is densely trafficked including many freight trains. In the case study, we will

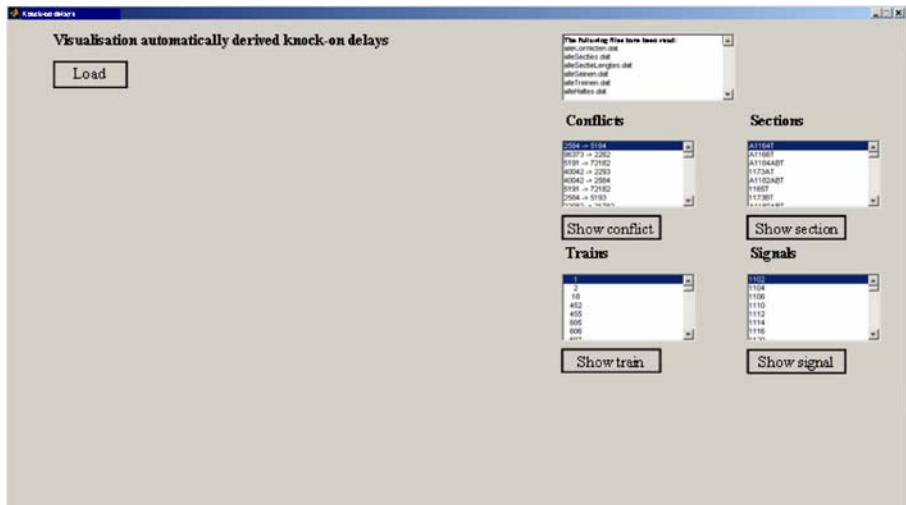


Fig. 5 User interface visualisation tool

focus on the western part of the Rotterdam yard, indicated by the oval in Fig. 6 and the detailed infrastructure sections and signals in Fig. 7.

A conflict has been identified by TNV Conflict between train 22082 (local train from Utrecht to Rotterdam) and train 21782 (Intercity from Utrecht to Rotterdam), just before entering the Rotterdam area. A visualisation of this conflict using blocking time diagrams is shown in Fig. 8. The blocking time consists of the sight and reaction time (taken fixed as 12 s), the approaching time (similar to the running time in the previous block), the running time in the current block, the clearing time (running time over the train length to clear the block), and a switching time (taken fixed as 2 s) (Pachl 2002). The blocking time diagram also shows the section occupation times consisting of the running and clearing time of a section. The

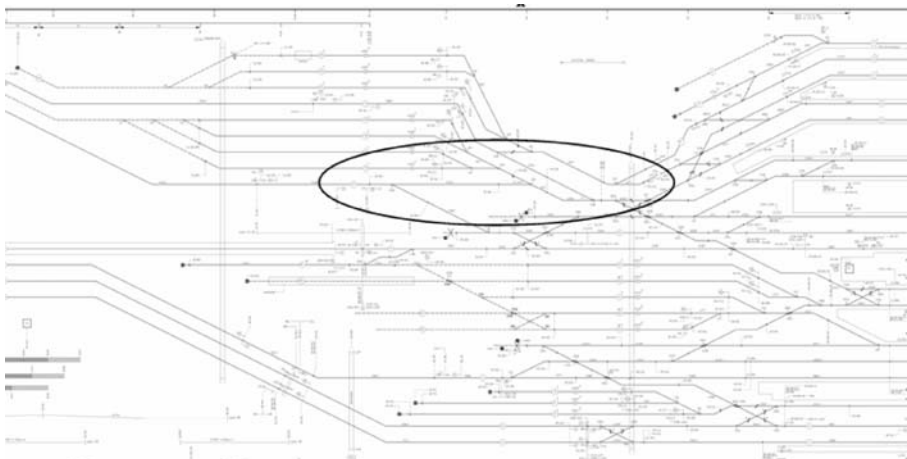


Fig. 6 Rotterdam-west track infrastructure

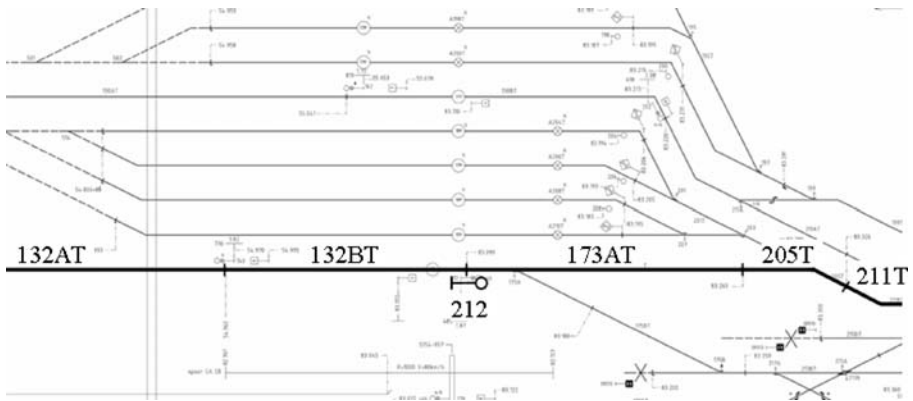


Fig. 7 Infrastructure detail in the Rotterdam-west station layout

conflict is indicated in the figure by an ellipse. At the top of the figure the infrastructure is shown with the successive signals.

In the blocking time diagram, both the blocking times (light blocks) and the track section occupation times (darker blocks) are shown. Missing information on the route and the corresponding sections around metre 54,000 should be noted. This is probably due to a fault in the track circuit of this route or another hardware problem. In the situation shown above, train 21782 is hindered by train 22082. Train 22082 is the upper train in the blocking time diagram (dotted lines). Figure 9 zooms to the conflict and indicates the names of the different track sections in the block diagram.

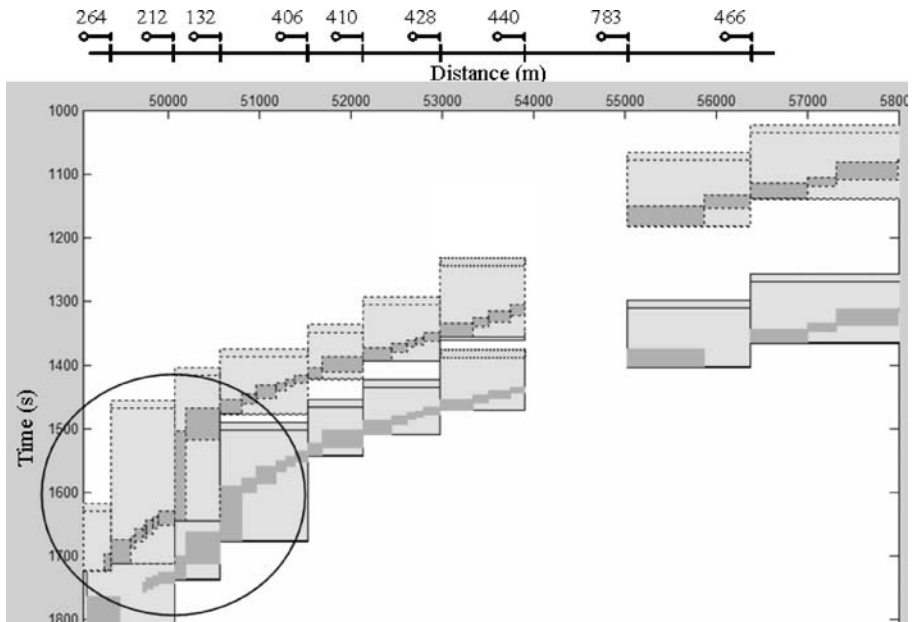


Fig. 8 Conflict between trains 22082 (top) and 21782 (bottom)

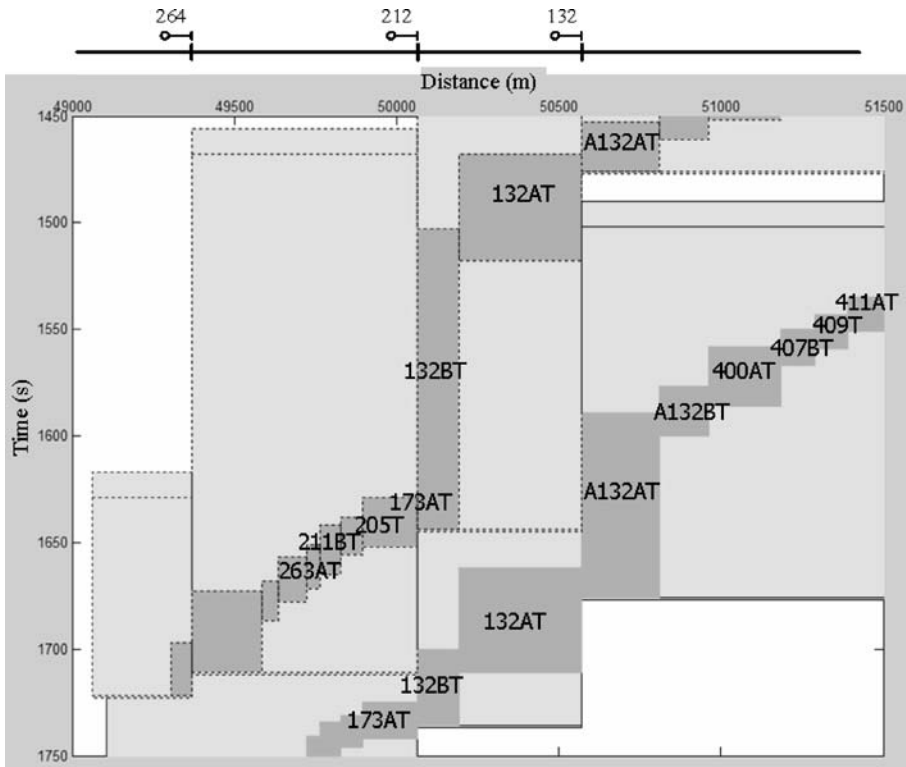


Fig. 9 Conflict between trains 22082 and 21782, zoomed in and with indication of section names

The cause of the route conflict to train 21782 becomes clear from Table 3, which shows the detailed section and blocking times of both trains. The conflicting route is boxed off. To proceed without any hindrance, train 21782 should have had access to the route consisting of the blocks 132AT and 132BT after 1,523 s. However, this

Table 3 Section occupation times and blocking times for trains 22082 and 21782 (in seconds)

22082		Section		Block		21782		Section		Block	
Section name	Occ.	Free	Start	End	Section name	Occ.	Free	Start	End	Section name	End
428ET	1372	1393	1294	1394	428ET	1487	1508	1422	1509		
410AT	1386	1410	1337	1422	410AT	1502	1529	1453	1543		
410BT	1404	1421	1337	1422	410BT	1522	1542	1453	1543		
411AT	1415	1428	1374	1477	411AT	1535	1551	1490	1677		
409T	1422	1433	1374	1477	409T	1543	1559	1490	1677		
407BT	1427	1439	1374	1477	407BT	1550	1567	1490	1677		
400AT	1432	1452	1374	1477	400AT	1558	1586	1490	1677		
A132BT	1445	1461	1374	1477	A132BT	1577	1600	1490	1677		
A132AT	1453	1476	1374	1477	A132AT	1589	1676	1490	1677		
132AT	1468	1518	1403	1645	132AT	1662	1711	1523	1737		
132BT	1503	1644	1403	1645	132BT	1700	1736	1523	1737		
173AT	1629	1652	1456	1712	173AT	1725	1742	1650	23875		
205T	1638	1656	1456	1712	205T	1731	1746	1650	23875		
211BT	1642	1665	1456	1712	211BT	1734	1751	1650	23875		
263AT	1651	1672	1456	1712	263AT	1741	1757	1650	23875		

route has been occupied by train 22082 from 1,403 s to 1,645 s, thus causing a delay to train 21782. To determine the hinder experienced by train 21782, we compare the section and blocking times of the hindered train to the section and blocking times of an unhindered train from the same train line (see Table 4).

Table 4 shows that train 21782 occupies the route consisting of sections 411AT to A132AT 90 s longer than the unhindered train. A similar extra occupation time is observed for the route consisting of sections 132AT and 132BT (+88 s).

The occupation times of the sections give more detailed comparisons. We see that already in the first few sections, train 21782 runs a little slower than the unhindered train (free running times), leading to section occupation times that are slightly longer (a few seconds) compared to the unhindered train. Starting at the third route in Table 4 (sections 411AT to A132AT), the occupation times increase significantly up to 69 s (section A132AT). At the end of this section, signal 132 shows a red aspect, until train 22082 has cleared this route. Adding all differences of the track section occupation times (in this case 129 s) leads to the total knock-on delay of train 21782. The last route block differs for both trains due to the follow up actions of the rolling stock. The corresponding blocking times therefore cannot be compared.

### 5 The incident registration system

Two main classes of knock-on delays can be distinguished: hindrance at shared track sections and waiting for a scheduled connection in a station. The latter includes passenger transfers, coupling of trains, turnrounds at terminals, and changing train crews. Hindrance of trains at tracks include route conflicts of an arriving or passing train in a station or junction (including platform conflict), route conflicts of a

**Table 4** Section occupation times and blocking times of train 21782 and a related unhindered train

21782	Section		Block		Blocking Time	Section Occ.	Section Unhindered	Blocking Time	Section Occ.
	Occ.	Free	Start	End					
428ET	1,487	1,508	1,422	1,509	87	21	428ET	80	19
410AT	1,502	1,529	1,453	1,543	90	27	410AT	83	23
410BT	1,522	1,542	1,453	1,543	90	20	410BT	83	17
411AT	1,535	1,551	1,490	1,677	187	16	411AT	97	13
409T	1,543	1,559	1,490	1,677	187	16	409T	97	10
407BT	1,550	1,567	1,490	1,677	187	17	407BT	97	10
400AT	1,558	1,586	1,490	1,677	187	28	400AT	97	15
A132BT	1,577	1,600	1,490	1,677	187	23	A132BT	97	12
A132AT	1,589	1,676	1,490	1,677	187	87	A132AT	97	18
132AT	1,662	1,711	1,523	1,737	214	49	132AT	126	38
132BT	1,700	1,736	1,523	1,737	214	36	132BT	126	35
173AT	1,725	1,742	1,650	203,875	22,225	17	173AT	884	16
205T	1,731	1,746	1,650	23,875	22,225	15	175BT	884	0
211BT	1,734	1,751	1,650	23,875	22,225	17	213BT	884	0
263AT	1,741	1,757	1,650	23,875	22,225	16	215BT	884	0

departing train from a station, and headway conflicts on the open track. These route (and headway) conflicts are identified automatically using the program described in Section 3. The train interrelations at transfer stations may be included afterwards—e.g., if a train waits for a delayed feeder train, then a dispatcher may set a conflicting route for another train first; this conflicting route is thus not the cause of a delayed departure.

The dispatchers have to identify the incidents leading to primary delays of trains. The knock-on delays due to route conflicts with a primary delayed train are then automatically identified by the program. Chains of conflicts are represented by linked lists of train numbers with the train number for the train suffering the primary delay at the top. The conflicting route block and resulting delay jump is also recorded for each train number in the list. Incidents could be assigned to the primary delayed trains by coupling the messages of the incident registration system to this list including the train numbers, codes of the affected track sections and kilometres. The delays corresponding to some disruption may thus be obtained as a list of the trains incurring a primary delay and the subsequent knock-on delayed trains. Aggregated delay statistics can be computed and assigned to the disruption as a measure of total hindrance. The automatic identification of knock-on delays enables a very accurate determination of primary and knock-on delays separately, while supporting the insight and tasks of dispatchers and timetable designers.

## 6 Conclusions and future work

This contribution describes a software tool which automatically identifies knock-on delays in an accurate and non-discriminative way. The input consists of data describing the infrastructure and the timetable as well as standard train describer records containing dynamic information about the railway system. The tool is based on the logic of the dynamic system, which has been described with a CPN model.

The software program used to derive the knock-on delays in the case study is currently a prototype. Although the structure of the tool and its basic functioning is complete, additional development work will be done on the interface, the visualisation and statistical analysis of the results. One of the further developments is to create a linked list, relating all trains with a knock-on delay to the trains causing this hindrance. Also, characteristics of this delay will be shown, e.g. the total hindrance time and section(s), where the hindrance occurred.

As already described, two main classes of knock-on delay are distinguished: hindrance at conflicting track sections and waiting for scheduled connections in stations. The first class is currently implemented in the tool. The scheduled connection times may depend on rolling stock requirements, transferring passengers or personnel connections that are known beforehand and can be added easily to the tool. Furthermore, the computation of the size of the differences between the measured and scheduled arrival, through and departure times of the individual trains and of the location with a level of precision of seconds will be implemented soon.

Finally, we will develop an on-line version of the tool, which is intended to be used as a decision support system by dispatchers.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Daamen W, Houben T, Goverde RMP, Hansen IA, & Weeda VA (2006) Monitoring system for reliability of rail transport chains. In Proceedings of the 7th World Congress on Railway Research, June 4–8, Montreal
- European Commission (2001) Directive 2001/14/EC of the European parliament and of the council of 26 February 2001 on the allocation of railway infrastructure capacity and the levying of charges for the use of railway infrastructure and safety certification. Off J Eur Communities L75, pp 29–46
- Exer A (1995) Rail traffic management. In: Bailey C (ed) European railway signalling. Institution of Railway Signal Engineers, A & C Black, London, pp 311–342
- Goverde RMP (2005) Punctuality of Railway Operations and Timetable Stability Analysis, PhD Thesis. Delft University of Technology, Delft
- Goverde RMP, Hansen IA (2000) TNV-Prepare: analysis of Dutch railway operations based on train detection data. In: Allan J, Hill RJ, Brebbia CA, Sciutto G, Sone S (eds) Computers in Railways VII. WIT, Southampton, pp 779–788
- Jensen K (1996) Coloured Petri nets: Basic concepts, analysis methods and practical use, 1, 2nd edition. Springer, Berlin
- Ministry of Transport, Public Works and Water Management (2004) Monitoringstudie, Letter to the chairman of the Dutch Parliament, 3 February 2004
- Pachl J (2002) Railway Operation and Control. VTD Rail Publishing, Mountlake Terrace
- UIC (2004) Capacity, Leaflet Code 406 R, Union International des Chemins de Fer (UIC)
- Weeda VA (2006) Analysis of dispunctuality, incident registration, and running and dwell time: Results of case studies Rotterdam-Dordrecht (in Dutch), Report no. T & P 2006.007. Department Transport & Planning, Delft University of Technology, Delft