Visual- and Model-based Techniques for Validating the corporate Traffic Information Chain

Paper submitted for publication and presentation at the
Transport Research Board 87th Annual Meeting, January 2008

Kin Fai Chan, Hans Wüst, Henk van Hulst, Jacorien Wouters
Department of Transportation, Rijkswaterstaat, Centre for Transport and Navigation
P.O. Box 1031, 3000 BA Rotterdam, The Netherlands
Phone +31-10-2825736
Email: kinfai.chan@rws.nl
URL: www.rws-avv.nl

Nanne van der Zijpp
Modelit,
Elisabethdreef 5, Culemborg, the Netherlands
Email: zijpp@modelit.nl

David Koh
Tenuki,
Distributieweg 30-32, Delfgauw, the Netherlands
Email: david@tenuki.nl

Paul Janssen
ICT Procos,
Kleine Landtong 15, Gorinchem, the Netherlands
Email: pjanssen@procos.nl

Keywords: data quality, data cleaning, data mining, validation
Word count: 5243+ 9 figures + 0 tables = 5243+ 9*250 = 7493 equivalent words
Abstracts: 245 words

ABSTRACT

One of the objectives of the Department of Transportation (DoT) in the Netherlands is to provide information about traffic conditions to road users and traffic managers. The usefulness of this information depends on the quality of the underlying traffic data. Traffic information is generated in a chain of applications, and in each processing step, data can get lost or get corrupted. Missing data are easy to detect, but corrupt data are not. So far, few applications were known that can check large amounts of data for abnormalities, and make an automated distinction between corrupt data and traffic-related outliers (e.g. because of accidents). The Da Vinci project, as described in this paper, aims at developing an application that can detect corrupt data and assist the specialist in analyzing the source of the error, so that the problem can be solved quickly.

Da Vinci includes different validation models which are used together to achieve the desired results. The models have access to the data from each processing station of the information chain. Furthermore, an analyzing module was built to help the specialist making case-specific drill-downs in the underlying data in order to find the cause of the corruption.

A prototype of Da Vinci has been operational since spring 2007. It successfully identified causes of different errors and help resolving operational problems which was unsolved for a long time. Different examples are presented in this paper showing how the Da Vinci can be used to improve data quality.

1. Introduction

The Dutch Department of Transportation (DoT) aims at providing a safe, reliable and optimal utilized traffic system. To this end, over 2500 km of the Dutch Freeway is instrumented with traffic sensors, mainly induction loops or radars. The detectors deliver a huge volume of data supporting real-time traffic management, monitoring, traffic information services, and other offline applications such as statistical and traffic engineering research. The traffic information is processed through different systems (or processing stations) before it becomes available to the end-users.

Unfortunately, throughout the sequence of generating and processing the information, data may get lost or corrupted because of technical and operational problems. Especially corrupt data pose a problem and may lead to unreliable information for traffic managers and road users (Wouters et al, 2005). Corrupt data are false data caused by measurement errors, processing errors or errors in the configurations.

With the huge volume of data, detection of corrupt data and trouble-shooting the causes of the problems becomes an extremely time-consuming task.

In order to cope with this problem, the Da Vinci (Data Validation & Inspection for Corporate Information Chain) project was started at the Centre for Transport and Navigation to develop validation methods for improving data quality. This paper gives a concise description of the achievements of the project.

2. Goal and approach of the Da Vinci project

Under operational situations, different kinds of errors may cause missing or corrupt data:

- technical errors: faults in detectors, out-stations, communications, inaccuracy in the technique of measurements under specific situations like extreme low speed;
- software and configuration errors: software bugs, configuration errors, incorrect design decisions;
- process errors: human errors like corrupt backup.

With the large amount of traffic data, the challenge is to develop methods which can (semi-) automatically detect corrupt data with minimum processing time.
It is known that error situations usually cause deviations in the data, when compared with neighbors or individual history. Therefore, if we can detect the obvious deviations in the data, then we should be able to identify the corrupt data automatically. Unfortunately, traffic events (e.g. incidents or traffic management measures) also cause deviations in data. Therefore, we need an approach to filter the false alarms caused by traffic events. The approach shown in Figure 1 is applied within the Da Vinci project. It involves the following steps:

1. Use validation models to automatically detect large deviations in data;
2. Conduct computer assisted analyses to find explanations for the deviations. Results of the analyses are stored in a database in order to help automating some of the analyse tasks;
3. Filter the deviations caused by known traffic events; label the data to avoid new false alarms in later validations (blue arrows in Figure 1);
4. The remaining cases with deviations are assumed to be caused by software bugs or configuration errors (green arrows);
5. Warn the maintenance departments to remove the sources of errors (green arrows);
6. If necessary, disqualify the corrupt data in the historical database. This introduces new missing data. As demanded by many traffic engineering applications, one need to re-estimate the missing data in the database (red arrows).

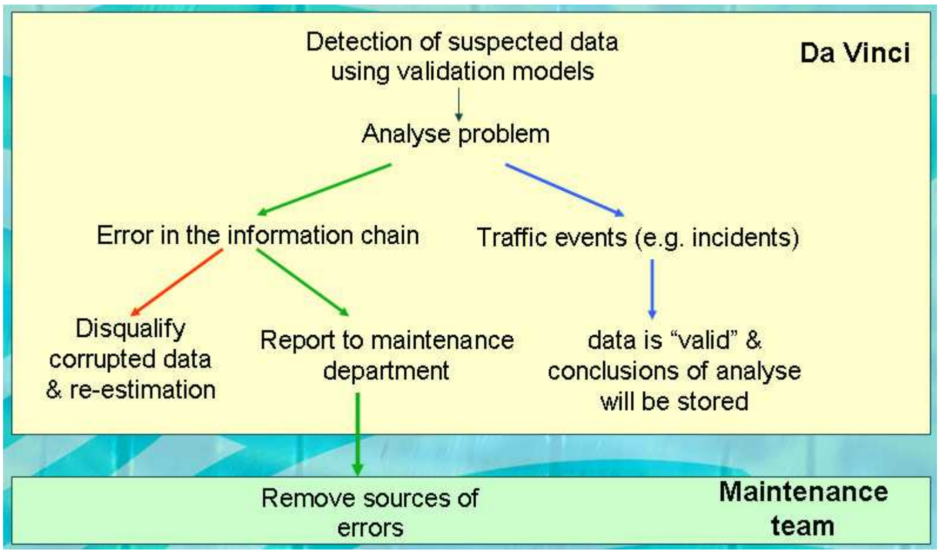

Detection of suspected data using validation models
Da Vinci
Analyse problem
Error in the information chain
Traffic events (e.g. incidents)
Disqualify corrupted data & re-estimation
Report to maintenance department
data is "valid" & conclusions of analyse will be stored
Remove sources of errors
Maintenance team

Figure 1: The Da Vinci approach

3. Traffic data within the information chain

Within the online information chain, detector data are processed through consecutive systems. The output of a system is retrieved online by the next system for further aggregation. At the end, traffic information is provided to end users through different media like VMS, radio, internet, SMS services, etc. Traffic managers in the traffic control centers also use the traffic information to monitor the traffic conditions on the road networks.

On the other hand, the "historical" traffic data is processed through another offline information chain for the following purposes:
- developing traffic management scenarios;
- determining the trends of traffic growth and supporting decisions of long term development plans;
- supporting policy makers in making traffic related policies, etc.

4. The configuration items and aggregation of data

The reliability of the traffic information depends strongly on how accurate the road situations are modeled into the so-called configuration files. The configuration files include the geographical and relational description of the following configuration items. Figure 2 shows how road situations can be modeled within a configuration file.

ARCS Individual detectors can be grouped together to form an ARCS (Aggregated Road Cross-Section). Speed and flow data of the underlying detectors can be aggregated into mean velocity (V) and intensity (I) of an ARCS.

ALE An ALE (Aggregated Link Element) is a road element bounded by two ARCS. The travel time from ARCS to ARCS can be estimated based on the V, I values of the beginning and terminating ARCS. An ALE can include entrance(s) or exit(s). If an entrance or exit is not equipped with detectors, then the processing application will estimate the amount of traffic entering or leaving the ALE.

RSM A RSM (Road Segment) is defined as a road element between two road junctions. One or more ALEs can be assigned to a RSM. RSMs are the common references used for distributing traffic information to the road users.

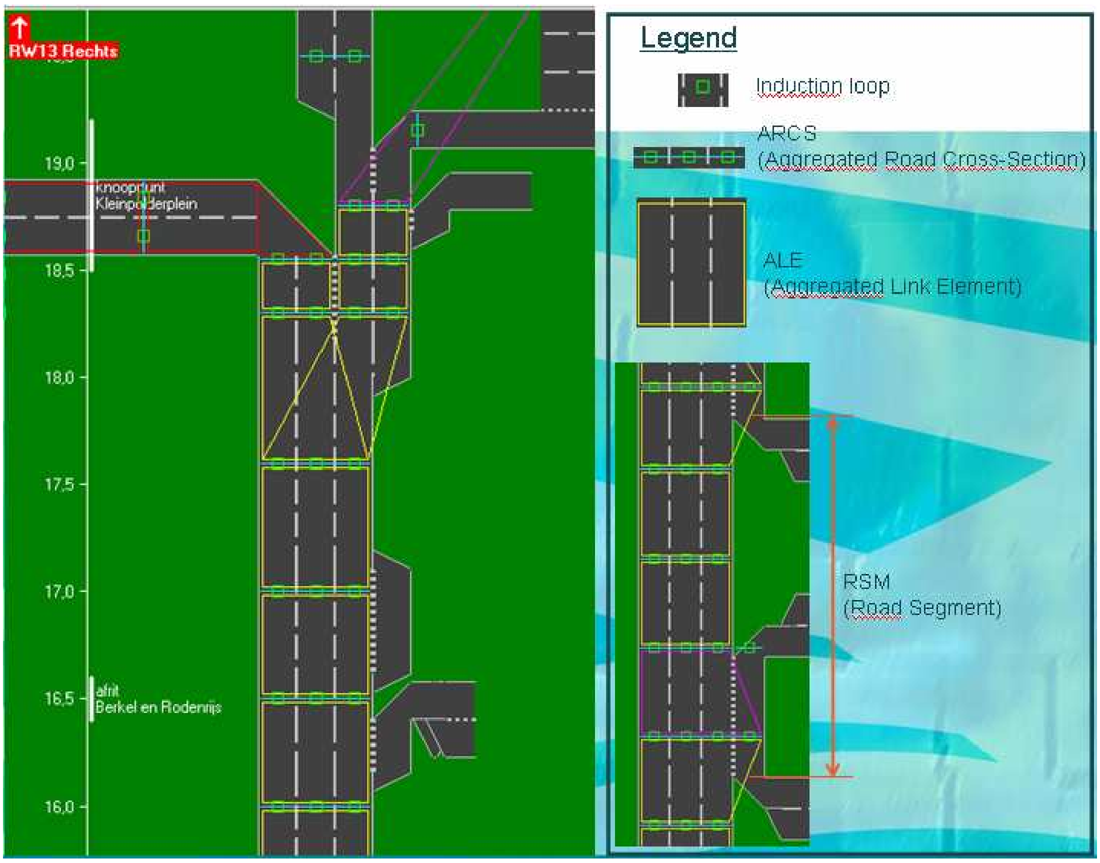

RW13 Rechts
19,0
knooppunt Kleinpolderplein
18,5
18,0
17,5
17,0
16,5
afrit Berkel en Rodenrijs
16,0
Legend
Induction loop
ARCS (Aggregated Road Cross-Section)
ALE (Aggregated Link Element)
RSM (Road Segment)

Figure 2: Examples of configuration items

5. DGL data historian

Very often, a maintenance team is responsible for his own system, but does not have easy access to data of related systems. This makes it difficult for the analyst to trace the sources of problems which take place at the previous stages of the processing.

The Dynamic Traffic Management Data Historian (DGL) is a data warehouse for integrating traffic data from different systems. Da Vinci makes use of the DGL to access data of the **entire** information chain. The DGL is implemented with the InfoPlus.21 database product of AspenTech. InfoPlus.21 is a data historian specially designed to collect and store large volumes of real-time data efficiently and makes it accessible for analysis and reporting. Unlike common relational databases, increases in the data volume has no significant degradation in the performance of InfoPlus.21 data historian.

At this moment, the DGL imports and processes data from more than 30000 configuration items every minute and it consists of more than 2 years history. The 30000 items includes 18000 detectors (per lane), 8000 ARCS, 3000 ALE and 1000 road segments. (The number of items is expected to increase rapidly in the near future.) Each item generates 11 to 21 data elements per minute. The InfoPlus.21 database acts as an object-based data historian. The historical data can be retrieved and processed with high performance.

6. The validation Models

In order to determine which validations methods are best suited, a survey was conducted in 2006 (Moon et al 2006). The existing methods were reviewed. In addition, validation methods applied in other domains and new methods were also considered. The conclusion is that different validation models should be combined to achieve the desired results, since each validation technique can only detect a specific subset of errors.
Based on this survey, the CoF (Conservation of Flow) model and the OSCAR (Outlier and Structural Change Auto-Recognition) model are incorporated in Da Vinci. Parallel to this, another logical model is developed

which is proved to be highly effective in detecting corrupt data. In the following paragraphs, the three models will be described separately.

6.1 Logical model

In order to achieve high performance, the 24-hour-indicators are introduced. Retrieval and analysis of the data of the 24-hour-indicators is at least 1500 times faster then retrieval of the minute values. Thereafter, validation rules are developed to detect corrupt data using the 24-hour indicators.

The 24-hour indicators for an ARCS

With the minute values of I and V, the following 24-hour-indicators can be computed:

- SPdaytot: total number of vehicle passages for a day
- Av_I: availability of minute intensity per 24 hours
- Av_V: availability of minute velocity per 24 hours
- V_{FF}: free flow velocity
- SV_{FF}: daily integral of the free flow velocity
- SV_m: daily integral of the measured velocity
- RSV: relative daily loss in velocity (=[$(SV_{FF} - SV_m) / SV_{FF}$])

V_{FF} and RSV are useful for validating the traffic data of an ARCS.
V_{FF} is the velocity with the maximal occurrence within a selected period. It can be computed using the histogram of the velocity (see Figure 3b).V_{FF} can be used to identify detectors which are not functioning properly.

In Figure 3a, the blue line represents V_{FF} while the red line represent the observed velocity of a day. RSV is the ratio between [the area bounded by the red line and the blue line] divided by [the area bounded by the blue line and the x-axis]. This example shows a congested location with RSV = 60.05. Generally, an extremely high value of RSV is an indication of error.

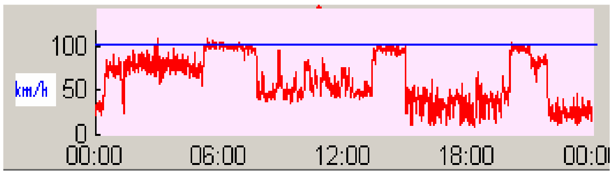


Figure 3a: Free flow velocity (blue line) and measured velocity (red line)

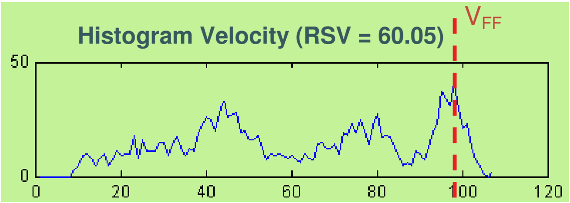


Figure 3b: Histogram of the velocity

The 24-hour indicators of an ALE

The following 24-hour-indicators can be computed for an ALE:

- Av_RT: availability of the travel time
- RT_{FF}: free flow travel time
- SRT_{FF}: daily integral of the free flow travel time
- SRT_m: daily integral of the measured travel time
- RSRT: relative daily loss in travel time (=[$(SRT_m - SRT_{FF}) / SRT_{FF}$])

RSRT is an effective parameter for validating the travel time data. The reliability of RSRT depends strongly on the accuracy of the free-flow travel time (RT_{FF}) (see Figure 4).

For an ALE with little congestion, the RSRT will be close to 0, because the measured travel time will be close to the free flow situation.
For an ALE where serious congestion is observed, RSRT will be much higher. In the example of Figure 4, RSRT = 113.77 due to the congestion in the evening and road works at night.
Generally, an extremely high value of RSRT is an indication of error.

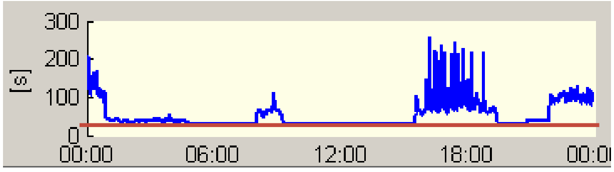


Figure 4a: Free flow travel time (red line) and measured travel time (blue line)

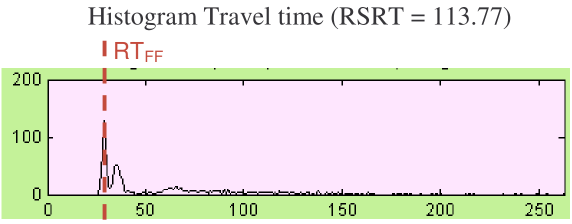
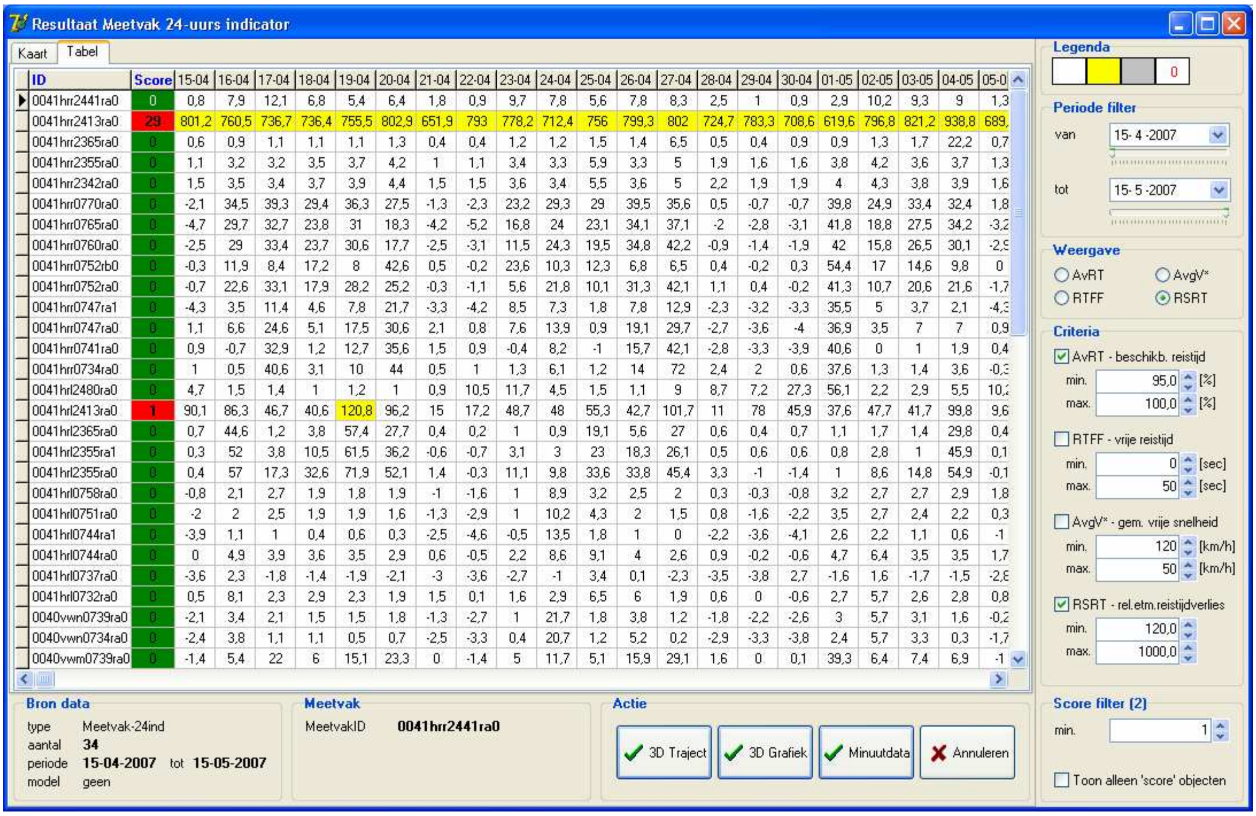


Figure 4b: Histogram of the travel time for 1 day

Applying the logical model

In Figure 5, it is illustrated how the logical model can be used to detect corrupt travel time data. Each column of the matrix contains the 24-hours values of different ALEs for a single day. Each row of the matrix contains the 24-hours values of an ALE for different days. The user can use the control panel on the right-hand side to select the different 24-hours indicators for inspection. He can also define combined criteria for detecting corrupt data. The column "Score" shows the number of days meeting the criteria. The threshold value for RSRT can be determined using the historical data of some highly congested locations. It is found that RSRT>120 is suspicious and RSRT>150 is practically unrealistic.

In Figure 5, with the criterium RSRT>120, one can notice that the ALE "0041hrr2413ra0" is suspect and should be inspected.



ID	Score	15-04	16-04	17-04	18-04	19-04	20-04	21-04	22-04	23-04	24-04	25-04	26-04	27-04	28-04	29-04	30-04	01-05	02-05	03-05	04-05	05-0
0041hrr2441ra0	0	0,8	7,9	12,1	6,8	5,4	6,4	1,8	0,9	9,7	7,8	5,6	7,8	8,3	2,5	1	0,9	2,9	10,2	9,3	9	1,3
0041hrr2413ra0	29	801,2	760,5	736,7	736,4	755,5	802,9	651,9	793	778,2	712,4	756	799,3	802	724,7	783,3	708,6	619,6	796,8	821,2	938,8	689,
0041hrr2365ra0	0	0,6	0,9	1,1	1,1	1,1	1,3	0,4	0,4	1,2	1,2	1,5	1,4	6,5	0,5	0,4	0,9	0,9	1,3	1,7	22,2	0,7
0041hrr2355ra0	0	1,1	3,2	3,2	3,5	3,7	4,2	1	1,1	3,4	3,3	5,9	3,3	5	1,9	1,6	1,6	3,8	4,2	3,6	3,7	1,3
0041hrr2342ra0	0	1,5	3,5	3,4	3,7	3,9	4,4	1,5	1,5	3,6	3,4	5,5	3,6	5	2,2	1,9	1,9	4	4,3	3,8	3,9	1,6
0041hrr0770ra0	0	-2,1	34,5	39,3	29,4	36,3	27,5	-1,3	-2,3	23,2	29,3	29	39,5	35,6	0,5	-0,7	-0,7	39,8	24,9	33,4	32,4	1,8
0041hrr0765ra0	0	-4,7	29,7	32,7	23,8	31	18,3	-4,2	-5,2	16,8	24	23,1	34,1	37,1	-2	-2,8	-3,1	41,8	18,8	27,5	34,2	-3,2
0041hrr0760ra0	0	-2,5	29	33,4	23,7	30,6	17,7	-2,5	-3,1	11,5	24,3	19,5	34,8	42,2	-0,9	-1,4	-1,9	42	15,8	26,5	30,1	-2,9
0041hrr0752rb0	0	-0,3	11,9	8,4	17,2	8	42,6	0,5	-0,2	23,6	10,3	12,3	6,8	6,5	0,4	-0,2	0,3	54,4	17	14,6	9,8	0
0041hrr0752ra0	0	-0,7	22,6	33,1	17,9	28,2	25,2	-0,3	-1,1	5,6	21,8	10,1	31,3	42,1	1,1	0,4	-0,2	41,3	10,7	20,6	21,6	-1,7
0041hrr0747ra1	0	-4,3	3,5	11,4	4,6	7,8	21,7	-3,3	-4,2	8,5	7,3	1,8	7,8	12,9	-2,3	-3,2	-3,3	35,5	5	3,7	2,1	-4,3
0041hrr0747ra0	0	1,1	6,6	24,6	5,1	17,5	30,6	2,1	0,8	7,6	13,9	0,9	19,1	29,7	-2,7	-3,6	-4	36,9	3,5	7	7	0,9
0041hrr0741ra0	0	0,9	-0,7	32,9	1,2	12,7	35,6	1,5	0,9	-0,4	8,2	-1	15,7	42,1	-2,8	-3,3	-3,9	40,6	0	1	1,9	0,4
0041hrr0734ra0	0	1	0,5	40,6	3,1	10	44	0,5	1	1,3	6,1	1,2	14	72	2,4	2	0,6	37,6	1,3	1,4	3,6	-0,3
0041hrl2480ra0	0	4,7	1,5	1,4	1	1,2	1	0,9	10,5	11,7	4,5	1,5	1,1	9	8,7	7,2	27,3	56,1	2,2	2,9	5,5	10,
0041hrl2413ra0	1	90,1	86,3	46,7	40,6	120,8	96,2	15	17,2	48,7	48	55,3	42,7	101,7	11	78	45,9	37,6	47,7	41,7	99,8	9,6
0041hrl2365ra0	0	0,7	44,6	1,2	3,8	57,4	27,7	0,4	0,2	1	0,9	19,1	5,6	27	0,6	0,4	0,7	1,1	1,7	1,4	29,8	0,4
0041hrl2355ra1	0	0,3	52	3,8	10,5	61,5	36,2	-0,6	-0,7	3,1	3	23	18,3	26,1	0,5	0,6	0,6	0,8	2,8	1	45,9	0,1
0041hrl2355ra0	0	0,4	57	17,3	32,6	71,9	52,1	1,4	-0,3	11,1	9,8	33,6	33,8	45,4	3,3	-1	-1,4	1	8,6	14,8	54,9	-0,1
0041hrl0758ra0	0	-0,8	2,1	2,7	1,9	1,8	1,9	-1	-1,6	1	8,9	3,2	2,5	2	0,3	-0,3	-0,8	3,2	2,7	2,7	2,9	1,8
0041hrl0751ra0	0	-2	2	2,5	1,9	1,9	1,6	-1,3	-2,9	1	10,2	4,3	2	1,5	0,8	-1,6	-2,2	3,5	2,7	2,4	2,2	0,3
0041hrl0744ra1	0	-3,9	1,1	1	0,4	0,6	0,3	-2,5	-4,6	-0,5	13,5	1,8	1	0	-2,2	-3,6	-4,1	2,6	2,2	1,1	0,6	-1
0041hrl0744ra0	0	0	4,9	3,9	3,6	3,5	2,9	0,6	-0,5	2,2	8,6	9,1	4	2,6	0,9	-0,2	-0,6	4,7	6,4	3,5	3,5	1,7
0041hrl0737ra0	0	-3,6	2,3	-1,8	-1,4	-1,9	-2,1	-3	-3,6	-2,7	-1	3,4	0,1	-2,3	-3,5	-3,8	2,7	-1,6	1,6	-1,7	-1,5	-2,8
0041hrl0732ra0	0	0,5	8,1	2,3	2,9	2,3	1,9	1,5	0,1	1,6	2,9	6,5	6	1,9	0,6	0	-0,6	2,7	5,7	2,6	2,8	0,8
0040vwn0739ra0	0	-2,1	3,4	2,1	1,5	1,5	1,8	-1,3	-2,7	1	21,7	1,8	3,8	1,2	-1,8	-2,2	-2,6	3	5,7	3,1	1,6	-0,2
0040vwn0734ra0	0	-2,4	3,8	1,1	1,1	0,5	0,7	-2,5	-3,3	0,4	20,7	1,2	5,2	0,2	-2,9	-3,3	-3,8	2,4	5,7	3,3	0,3	-1,7
0040vwm0739ra0	0	-1,4	5,4	22	6	15,1	23,3	0	-1,4	5	11,7	5,1	15,9	29,1	1,6	0	0,1	39,3	6,4	7,4	6,9	-1

Bron data
type Meetvak-24ind
aantal **34**
periode **15-04-2007** tot **15-05-2007**
model geen

Meetvak
MeetvakID **0041hrr2441ra0**

Legenda
□ 🟨 ▨ 0

Periode filter
van 15- 4 -2007
tot 15- 5 -2007

Weergave
○ AvRT ○ AvgV*
○ RTFF ● RSRT

Criteria
☑ AvRT - beschikb. reistijd
min. 95,0 [%]
max. 100,0 [%]
□ RTFF - vrije reistijd
min. 0 [sec]
max. 50 [sec]
□ AvgV* - gem. vrije snelheid
min. 120 [km/h]
max. 50 [km/h]
☑ RSRT - rel.etm.reistijdverlies
min. 120,0
max. 1000,0

Score filter (2)
min. 1
□ Toon alleen 'score' objecten

Figure 5: GUI of the logical model for detecting corrupt travel time data

Detailed analysis of the minute data reveals that a configuration error causes the problem: in a complex junction, an exit is accidentally omitted in de configuration. The travel time algorithm then registers only vehicles entering the ALE but very little vehicles are found exiting the ALE. For this reason, extremely high travel time is estimated and passed on to other applications. Figure 6 shows clearly the difference in travel time after the configuration has been corrected.

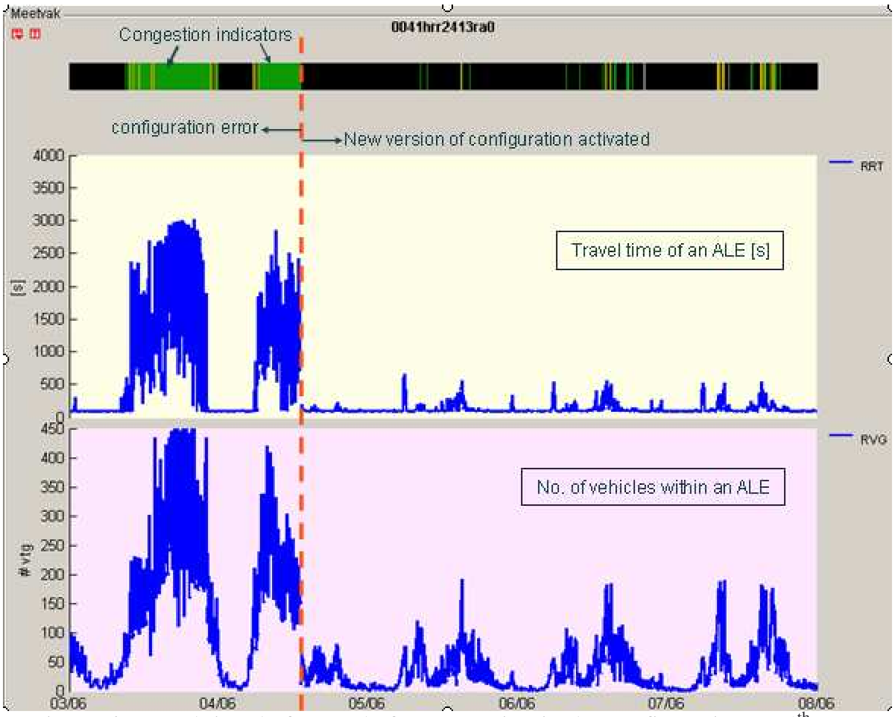


Figure 6: Travel time before and after correction in the configuration on 4th June

***6.2 OSCAR model** - a Bayesian Forecasting model for validation in "time"*

In the time series context, the primary interest for the validation are *outliers* and *indicators of structural change*. Traditional methods are designed to detect outliers using historical day-curves. For example, if the measured speed (or flow) deviates too much from the historical day-curves, then the measurement can be identified as a outlier. However, these methods fail to recognize discontinuities or structural changes in the data. As a result, valid data may accidentally be identified as outliers giving rise to false alarms. Intensive expert judgment is then required to verify the outcomes. For these reasons, the OSCAR model was developed which has the flexibility to detect outliers, abrupt and slow structural changes simultaneously.

The computational demand of the model is reduced by modeling every single time-step of a week-curve by a simple constant level state space model which follows the Kalman filter recursive update cycle. So a one minute time-step time series is modeled by 10080 state space models, each with a one week time-step. Each model has a one dimensional state, representing the level of a single minute of the week-curve.

This modeling approach is combined with an adapted version of the Bayesian Model Monitor as introduced by West and Harrison (1986).

The standard model represents the stable, constant level process of a week-curve point. The model always starts from a prior state estimate:

$\mu_{t-1}|D_{t-1} \sim N\ (m_{t-1},C_{t-1})$, with
m_{t-1} = the state estimate, and C_{t-1} its variance

"$|D_{t-1}$" stands for "conditional on all observations from t=0 to time-step t-1" , while
"~ N" means "normally distributed".

The initial state μ_0 is chosen as the median of the first five observations with variance of half this median level. The state prediction for time t is given by the state evolution equation

$\mu_t=\mu_{t-1}+w_t$, where

w_t is a fraction of the state variance C_{t-1}

The state prediction is stochastically described as:

$\mu_t|D_{t-1} \sim N\ (a_t, R_t)$,

with

$a_t = m_{t-1}$ and

$R_t = C_{t-1}/\delta$

δ is a discount factor quantifying the variance increase due to the evolution noise term w_t as a fraction of the state variance C_{t-1}. δ may take positive values less than 1, typically between 0.8 and 1.

If an observation is available for time t, the state prediction is updated through a Kalman filter update step, otherwise the evolution equation is applied again for consecutive time-steps until an observation becomes available for a Kalman filter update step.

The state update requires a link between the state prediction $\mu_t|D_{t-1}$ and the observation y_t which is stochastically described by the measurement equation:

$y_t \sim N(f_t, Q_t)$, with

$f_t = a_t.Q_t = R_t+V_t$, V_t being the observation noise.

For traffic intensities V_t is modeled using a variance law $V_t=f_t^p$, with p=1.15, which reflects the over-dispersion relative to the Poisson variation of these count data.

The updated state is described by:

$\mu_t|D_t \sim N\ (m_t,C_t)$

with update equations:

$m_t=a_t+A_te_t$ with $e_t=y_t-f_{t0}$,and

$C_t = A_tV_t$, with $A_t= R_t/Q_t$ being the Kalman gain

The model monitor of West and Harrison (1986) monitors the model performance at each time-step by comparing it with an alternative model. This alternative model is constructed in a computationally efficient way by inflating the prediction variance Q_t of the standard model to Q_t/ρ using the variance inflation factor ρ.
As long as the Bayes factor $H_t = \exp[0.5(\rho-1)\ e_t^2/Q_t]/\sqrt{\rho}$ exceeds a threshold τ, the standard model is considered as adequate, otherwise the alternative is favoured and y_t is marked as a discrepant observation. Values of ρ between 0.1 and 0.3, and $\tau < 0.4$ are generally recommended.

For the purpose of deciding whether observation y_t is an outlier or the onset of a structural change, an outlier model and a change model are defined. Both models are based on the standard model. The outlier model prediction is:

$y_{t+1} \sim N(fo_{t+1}, Qo_{t+1})$,

with

$Qo_{t+1} = R_t/\delta + V_{t+1}$

It is based on the standard model evolved to time t+1 ignoring observation y_t, i.e. without a Kalman filter update at time t.

The change model can quickly adapt to the observation y_t, which is assumed to be the onset of a structural change. This is achieved by inflating the original state prediction variance R_t to R_t/Δ_t, with inflation factor $\Delta_t = \rho/(1+(1-\rho)V_t/R_t)$.
No extra parameter is needed since this factor is derived from the variance inflation factor ρ and the ratio V_t/R_t. The inflated state variance allows the required quick adaptation during the Kalman filter update, which is followed by predicting f_{t+1} for observation y_{t+1} using the standard state space and observation equations.
If the (log)Bayes factor

$$M_t = \log(\sqrt{Qo_{t+1}}/\sqrt{Q_{t+1}}) + (y_{t+1}-fo_{t+1})^2/(2Qo_{t+1}) - (y_{t+1}-f_{t+1})^2/(2Q_{t+1})$$

exceeds a preset threshold, say 3 to 6, y_t is seen as the onset of a structural change, otherwise as an outlier. The preferred model is then chosen as the standard model for further monitoring.

Further, a cumulative Bayes factor W_t with run length l_t, being the product of l_t consecutive monitor Bayes factors H_t, is used to identify slow drift events, for which the above mentioned monitor does not signal or signals too late. The cumulative Bayes factor is calculated recursively following

$$W_t = H_t \min[1, W_{t-1}]$$
$$l_t = l_{t-1}+1 \qquad \text{if } W_t < 1$$
$$l_t = 1 \qquad \text{if } W_t \geq 1$$

So as W_t reaches a value of 1 or higher, the monitor is reset.

As long as W_t does exceed τ, the standard model is considered as adequate, otherwise a slow drift event is signalled at approximate time-step $t-l_t+1$ and the model is adapted to the observation y_t using the same inflated state variance Kalman filter update step as for the above-mentioned abrupt change model.
The monitor can be made more sensitive to slow drift events by also signalling when the run length l_t exceeds 4 or 5.
After a signal, the slow drift monitor is reset (W_t=1, l_t=1). It is also reset when an abrupt change is detected.

This parameterised monitor has proved to be very robust for its parameter settings. It also shows effective change detection performance in situations with frequent missing values and many outliers. It can be effectively applied to intensity data of different temporal aggregate levels. Time-series of a year's length with time-steps of one hour and one minute and take about one second and a tenth of a second computation time respectively.
As yet the method has only been applied to traffic intensity data. Time-series of different type need different ways of assessing the observation variance estimate V_t.

Applying the OSCAR model

Case 6.2: Freeway A13R at exit Overschie
Figure 7 shows the results of OSCAR runs for an ALE. The runs is conducted using 1 year data of SPdaytot (number of vehicle passages on a day).
The figure represents a matrix of 7 x 52 cells. The first 5 columns contain data of the working days. The last 2 columns contain data of the weekend. On the left hand side, the trend of the 52 Tuesdays of the year is shown. The OSCAR model successfully detects the abrupt structural change (green markers) in the second week of April. The structural change is the result of a correction in the configuration. This example also shows how OSCAR can be used to monitor when a corrective action is carried out and whether the action really leads to the expected results.

The last Tuesday of December (26th December 2006) is recognized as an outlier (red marker). The outlier is caused by an unexpected system shut-down from 24th December. The defect was restored at the evening of 26th December which caused the missing flow data, and therefore an outlier in SPdaytot.
Furthermore, 28th November (the last Tuesday of November) is recognized as a gradual structural change (blue marker). A system failure at that night in combination with other unknown factors leaded to this conclusion.

Other applications of OSCAR

The OSCAR method can be applied for data of different resolutions, for example, 1 minute, 15 minutes or hourly interval. Theoretically, OSCAR can also be used to validate other parameters than intensities. The capability to detect structural change in velocity (e.g. due to traffic management measures or correction in the detector functioning) has already been demonstrated.

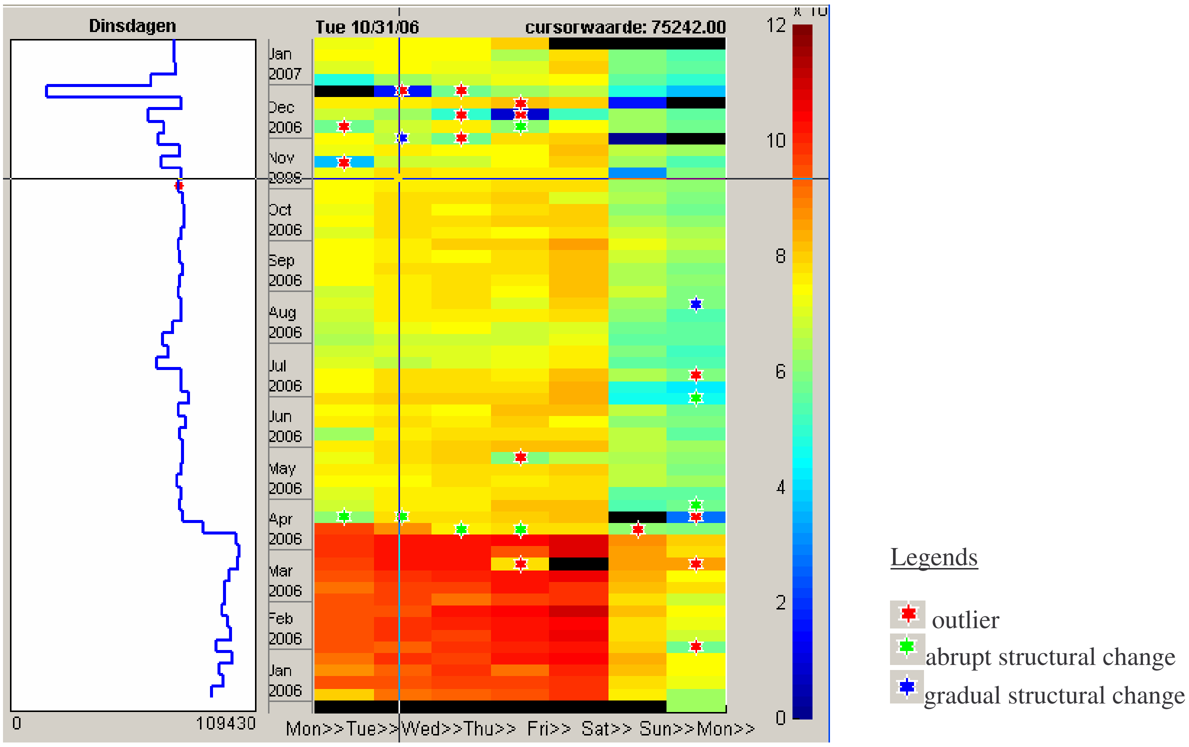


Figure 7 : Results of the OSCAR runs based on 1 year data

***6.3 CoF model** - the Conservation of Flow model for validation in "space"*
As a matter of fact, certain kinds of corrupt data cannot be detected by OSCAR. For instance, if a configuration error occurs in a system for a long time, then it will no longer cause any outliers and structural changes in the history. In such cases, the CoF model can be used to detect the error. Unlike other common traffic engineering models, CoF is easy to use and requires less processing time.

The notion that vehicles that enter a given area of a traffic network should sooner or later also leave this area is called the Conservation of Flow (CoF) principle. The CoF principle is a powerful aid in checking traffic data as it enables checking the consistency of observed flows.
If observed traffic flows are not consistent this may be caused by:
a. poor technical functioning of a road side based detector,
b. errors in the metadata describing the position of a detector, or
c. errors in the network representation of the traffic system.

From the viewpoint of applying the traffic data to monitor or manage traffic, either of these are equally problematic, and therefore worthwhile to recognize.

In practice, a visual inspection of the data and network is of great help in clarifying problems that involve corrupt data. Because this is a laborious task, the CoF check can be used as a first filter to narrow down the number of potential problems. In previous research the CoF technique was successfully used to perform the necessary data-cleaning before applying a dynamic OD-matrix estimator (see Taylor et al, 2004), but the CoF-equations where still specified by hand. In the present project we present a method that automates this task. This opens the possibility to apply the method to nation wide datasets. These typically contain tens of thousands detectors and road sections.

The procedure for automated generation and application of the CoF model consists of the following subtasks:
a. transform a GIS description of the road infrastructure to a directed graph representation;
b. match detector-metadata to this graph;
c. aggregate detector-flows over ARCS (road cross sections) and a sufficiently long time interval;

d. generate and apply the CoF equations;
e. identify the consistent and inconsistent observations.

Tasks [a], [b] and [c] are specific to the way the monitoring system is organized, Tasks [d] and [e] are more generic and are described below.

Task [d]

The left hand side of a CoF equation denotes the observed flows that enter a pre-defined study area. The right hand side denotes the observations of outgoing flow. A CoF-equation is hence a linear combination of traffic counts. Each study area implies a CoF equation. The question is however, how many independent CoF equations can be built for a given network and how to construct this set in an automated manner. Figure 8 shows the mechanism that builds the CoF equation for a study area that involves node *n*. The study area is initialized with node *n*. All links ending in and starting from this node are considered. If either of these links is unobserved, the study area is expanded with the start- or endnode of this link. This process is continued until a study area is found with only observed entry- and exit links, or an so-called origin- or destination node is encountered. Origin and destination nodes are nodes where traffic emerges or disappears from the network. By repeating this process for each node in the network all relevant CoF equations will be found.

Task [e]

The CoF equations that satisfy the following test are categorized as *inconsistent*:

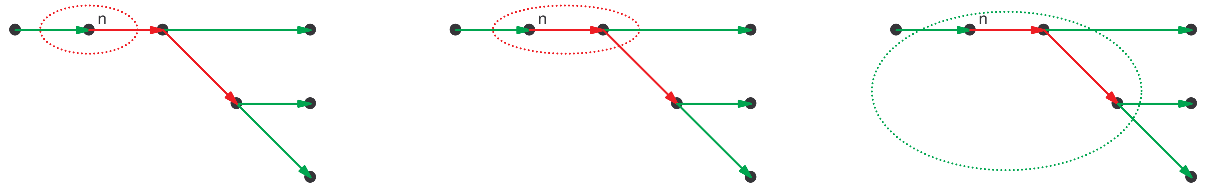
50%*abs(total flow in-total flow out)/ (total flow in+total flow out) > threshold_fail

In an analogous way CoF equations that pass the following test are categorized as *consistent*:

50%*abs(total flow in-total flow out)/ (total flow in+total flow out) < threshold_pass

After the last iteration, traffic counts are categorized as consistent, inconsistent or neutral. The latter category represents counts that can not be accepted or rejected with a sufficient level of certainty. If a CoF equation is consistent, then all involved traffic counts are flagged as consistent. If a CoF equation is not consistent, then at least one of the counts involved must be inconsistent as well. The inconsistent count can only be identified if there is only one candidate, in other words: if all but one have been categorized as consistent.

Tasks [d] and [e] are therefore applied in an iterative manner. First as many as possible consistent counts are flagged, then inconsistent counts are identified. Then a new iteration that ignores the inconsistent observations is started. The process is continued until a next iteration does no longer alter the consistency status of any count. When the iterations are complete, all traffic counts have been categorized as consistent, inconsistent or neutral.



[a] [b] [c]

Figure 8: mechanism for constructing the CoF equations for a study area
red arrow: unobserved flow. Green arrow: observed flow.
Red circle: study area with one or more unobserved entry- or exit-flow.
Green circle: study area with fully observed entries and exits.

Applying the CoF model
Case 6.3: Freeway A13L, rush-hour-lane
In Figure 9, results of the CoF runs model are shown. The validation is carried out for the flow of a day for a selected trajectory on the freeway A13L. This trajectory includes a rush-hour-lane on the hard shoulder which is only opened to traffic during rush hours. The graph represents the data of a matrix containing 18 rows x 1440 columns. The vertical axis shows the consecutive locations (ARCSs) while the horizontal axis shows the minutes of the day. The blue markers indicate suspicious data based on a CoF test applied to one hour aggregated data.

The CoF model indicates the locations 119, 131, 155, 160, 122 as suspicious when the rush-hour-lane is closed (outside the rush hours). Figure 9 also visually shows the distinguished pattern for these locations.
Further analysis shows that the reported intensities outside rush hours are wrong. The problem is caused by a software error in the signaling system at the very beginning of the information chain. This leads to errors in the messages concerning the states of the rush-hours-lane whenever the lane is closed. With the direct access to data of the entire information chain, we can quickly trace back the cause of the problem.

For this particular error, the implications are particularly severe because it causes the Decision Support System to suggest opening the rush-hour-lane at the wrong moment. Again this is an example of how problems in one subsystem (wrong aggregation of traffic flows over lanes) cause problems in another subsystem (opening the rush hour lane at the wrong moment).

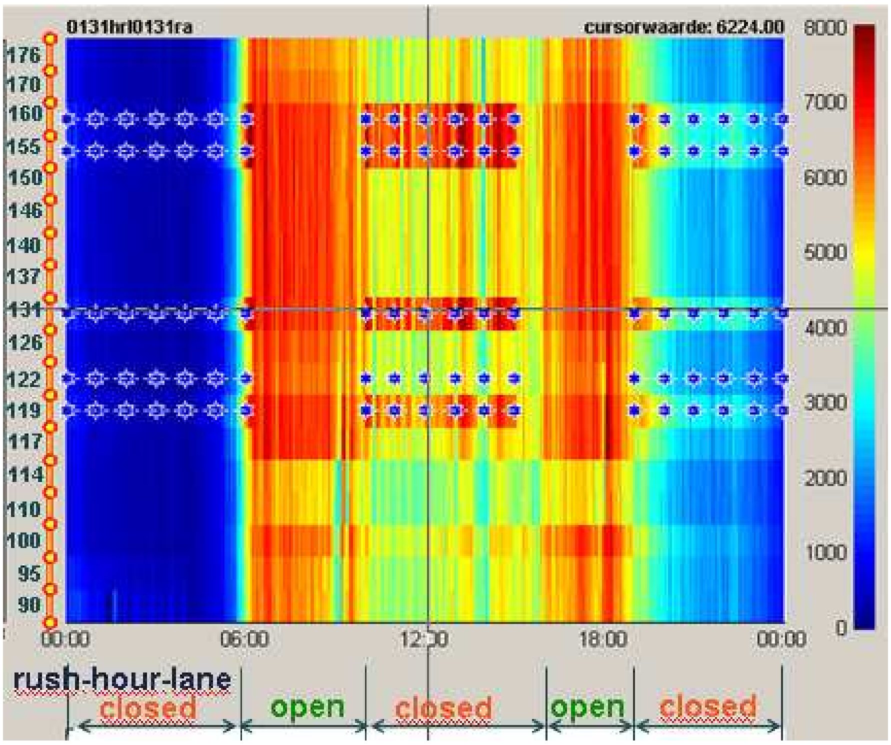


Figure 9 : Results of the CoF runs based on 1 hour interval

7. The way ahead
The following improvements are planned for the Da Vinci development:

- *event-based information*

Some of suspected data may be caused by incidents, road-works, special events or extremely bad weather. Therefore, *event-based information* will be used to improve the validation models and reduce the number of false alarms.

- *conclusions registration module*
  After the detailed analysis, conclusions can be drawn mentioning whether a specific event or an error within the information chain leads to the suspicious data. In that case, the data will be marked and the conclusions will be stored. After that, users of the data can be informed about the causes of deviations.
- *applying Da Vinci under real-time conditions*
  Technically, the validation methods can be applied to detect corrupt data under real-time conditions. However, the challenge is to develop strategy to support problem analysis under real-time conditions, since traffic managers normally have limited time to analyse the problems while conducting other traffic management tasks.
- *estimation of missing data*
  As stated in Chapter 2, corrupt data should be disqualified. Consequently, new missing data will be introduced. Since many traffic management applications do not support missing data, one needs to provide strategies to estimate missing data. In fact, the OSCAR model has the capability to estimate missing data in the historical dimension. Future research will be conducted to develop suitable strategies for estimating missing data in multiple dimensions using combination of OSCAR and CoF.

8. Conclusions

Da Vinci makes it possible to identify errors and to trace back their causes through different applications where this was not possible before. This will significantly improve the quality of data that is made available to end-users while reducing development and maintenance cost. Also Da Vinci improves the understanding of the traffic information chain by making the data more accessible. This gives better insight in the sometimes unintended consequences of design and configuration decisions, and enables the identification of problems before rather than after they have caused problems in systems that depend on them. A welcome side effect of the project is that knowledge about the different data processing within traffic information chain has been made more explicit and can be transferred to new specialists easier than before.

9. References

E. Moon, T. Brijs, (2006), “Survey of validation techniques applied for traffic engineering data”. Final report, Transport Research Institute IMOB, University of Hasselt.

Chan, K.F. (2006) “Automatic detection of missing and corrupt data”. *Intelligence to Trans-European Road Network Conference*, Barcelona, Spain.

Wouters, J.A.A., Chan, K.F., Kolkman, J., Kock, R.W. Customized Pretrip Prediction of Freeway Travel Times for Road Users. (2005), *Transport Research Record: Journal of the Transport Research Board, No. 1917-04*, TRB 2005.

West, M. (1986), Bayesian Model Monitoring, *Journal of the American Statistical Society*, B48:1, p 70-78.

West, M, Harrison, P. J. (1986),“Monitoring and Adaptation in Bayesian Forecasting Models” , *Journal of the American Statistical Association*, 81, 395, pp-73-83, 1986.

West, M, Harrison, P. J. (1997),“ Bayesian Forecasting and Dynamic Models” , *Springer.*

Taylor, N. , X Zhang, N. van der Zijpp, C White, S. Beale (2004). “Origin-destination matrix estimation for the active traffic management project” , *12th IEE International Symposium on Road Transport Information and Control*, London, UK

Aspen Technology, Inc., (2002). Power Generation Industry Integrated Solution.

Robinson, S. and Polak, J.W., (2006), “ILD data cleaning treatments and their effect on the performance of urban link travel time models”, *85th Annual Meeting of the Transportation Research Board*, Washington D.C., January 2006.