Scalable Data Analytics,
Scalable Algorithms, Software Frameworks
and Visualization ICT-2013 4.2.a

Project	**FP6-619435/SPEEDD**
Deliverable	**D8.6**
Distribution	**Public**

http://speedd-project.eu

Final Evaluation Report of SPEEDD for Traffic management

Chris Baber, Natan Morar, Faye McCabe, Sandra Starke
(University of Birmingham)

Alain Kibangou
(CNRS)

Status: Final

February 2017

Project

Project Ref. no	FP7-619435
Project acronym	SPEEDD
Project full title	Scalable ProactivE Event-Driven Decision Making
Project site	http://speedd-project.eu/
Project start	February 2014
Project duration	3 years
EC Project Officer	Stefano Bertolo

Deliverable

Deliverable type	Report
Distribution level	Public
Deliverable Number	D8.6
Deliverable Title	Final Evaluation of SPEEDD dashboards for traffic management
Contractual date of delivery	M36 (January 2017)
Actual date of delivery	December 2016
Relevant Task(s)	WP8/Tasks 8.6
Partner Responsible	CNRS
Other contributors	UoB
Number of pages	27
Author(s)	C. Baber, S. Starke, N. Morar, A. Kibangou
Internal Reviewers	
Status & version	Final
Keywords	Evaluation, User Interface Design, Human Factors

Contents

Table of Figures

Executive Summary

This deliverable reports the final evaluation of SPEEDD prototype for the Road Traffic Management use case. In this report, we first present the results of an experimental comparison of the first and final dashboard designs on a simulated road traffic management task; the design process for this dashboard was described in D5.3 and an initial usability evaluation was reported in D8.5. It is shown that the final design is superior to the first design in terms of supporting decision making and responding to events. It is argued that the performance times compare favourably with baseline measures derived from observations on the control room. It is also shown that the reliability of the computer support can have an impact on user decision making. Second, the other components of the SPEEDD prototype are also evaluated (Event learning and forecasting, and decision making). It is shown that we can alert a future congestion in an average 3-4 min before the congestion is detected, enabling proactive actions. Eventually, comments of a traffic expert on SPEEDD outcomes are reported. It is stated that SPEEDD has contributed significantly to advance the state of the art and has open new possibilities for traffic management.

1 Introduction

1.1 History of the Document

Version	Date	Author	Change Description
0.1	22/11/2016	Chris Baber	First version of the document
0.2	14/12/2016	Chris Baber	Review and additional material
0.3	27/12/2016	Alain Kibangou	Additional material
0.4	21/02/2016	Alain Kibangou	Addition of section 5

1.2 Purpose and Scope of Document

The purpose of this document is to report the final evaluations of the SPEEDD dashboard in the Road Traffic Management use case. In terms of evaluation, the aim is to show how the dashboard can affect decision making and congestion detection by users. The other components of the SPEEDD prototype are also evaluated (Event learning and forecasting, and decision making). In addition, expert evaluation is provided.

1.3 Relationship with Other Documents

This document is related to the following deliverables: 8.1 User Requirements, 8.3 Initial Evaluation Report, D8.5 Intermediate evaluation report; D5.1 Design of User Interface for SPEEDD Prototype, D5.2 Design of User Interface for SPEEDD Prototype (year 2); D5.3 Design of User Interface for SPEEDD prototype (year 3); D.3.3 Third Version of Event Recognition and Forecasting Technology
D 4.3 - Third Version of Real-Time Decision-Making Technology.

1.4 Sources of Information

For this report, an experiment was conducted in which trained users of the dashboard undertook a series of road traffic management tasks using dashboard from prototype 1 and dashboard from prototype 2.

2 Defining a Baseline for Performance

2.1 Introduction

A continued challenge for the evaluation of the dashboards in the SPEEDD project lay in the definition of an appropriate point of reference for comparing performance with the SPEEDD prototype against 'conventional' practice. There are two reasons for this. First, SPEEDD took ramp metering as one of its target applications in the road traffic use case. This provides a focus for the decision and control work (WP4) and allowed the architecture (WP6) to have a feedback from decision to effect on the world (through the integration with the AIMSUN simulations). While the Grenoble DIRCE has been discussion with providers for installation of ramp metering, this is something that it not yet implemented. Consequently, we do not know how the DIRCE operators will perform ramp metering. Second, SPEEDD used congestion defined by data from in-road sensors as the basis of its modelling. In the DIRCE control room, operators rely on CCTV images rather than processed sensor data (the operators can have access to sensor data which describes traffic density on the rocade sud). Thus, it is not possible to make a direct comparison. However, the DIRCE control room (in early 2016) implemented a system which analysed CCTV footage to 'recognise' obstructions in the road, e.g., vehicles stopped on the road, material on the road, pedestrians, animals or cyclists on the road. This system scans CCTV content and, if there is a possible obstruction, will alert the operator (see D8.5 Intermediate Evaluation Report of SPEEDD Prototype for Traffic management). The operator then views the CCTV content and decides whether there is an obstruction and whether this needs to be responded to. From observations of operator response to this system, we have an approximate measure of response and decision time (although, of course, this is a crude index of the work that they actually perform).

In earlier laboratory experiments, we showed that decision time varied from around 6.5s to 10s when the task was simply to respond to computer suggestions about ramp metering decisions (D8.3 Initial evaluation report). In the control room, the decision involves a combination of receiving notification / alert, determining incident type, location and impact, determining response and completing a report on the incident management. Thus, a simple reaction time measure will not reflect the complexity of the work. Having said this, the video recordings that we made in DIRCE (May 2014) suggest that it takes around 20s for the operator to make an initial diagnosis of an incident in response to a radio notification. This time involves checking the map and CCTV for the location of the incident and then categorising the incident type. In our observations in January 2016, it took operators approximately 15s to respond to the alert from the automated video analysis. This involved noticing that one of the CCTV panels had processed an event, interpreting the event and then categorising the event. Consequently, we would anticipate that 15s – 20s would constitute an upper ceiling to a performance times – if the use of the SPEEDD system took longer to complete the task than the manual version then this would probably not be acceptable.

For this report, we have decided to compare the initial dashboard of the SPEEDD prototype, which

was designed to support SPEEDD tasks of congestion detection and ramp metering within the SPEEDD architecture (figure 1), with the final version of the dashboard, which was optimised to support operator decision making (figure 2). The development of the dashboards is reported in detail in D5.3.

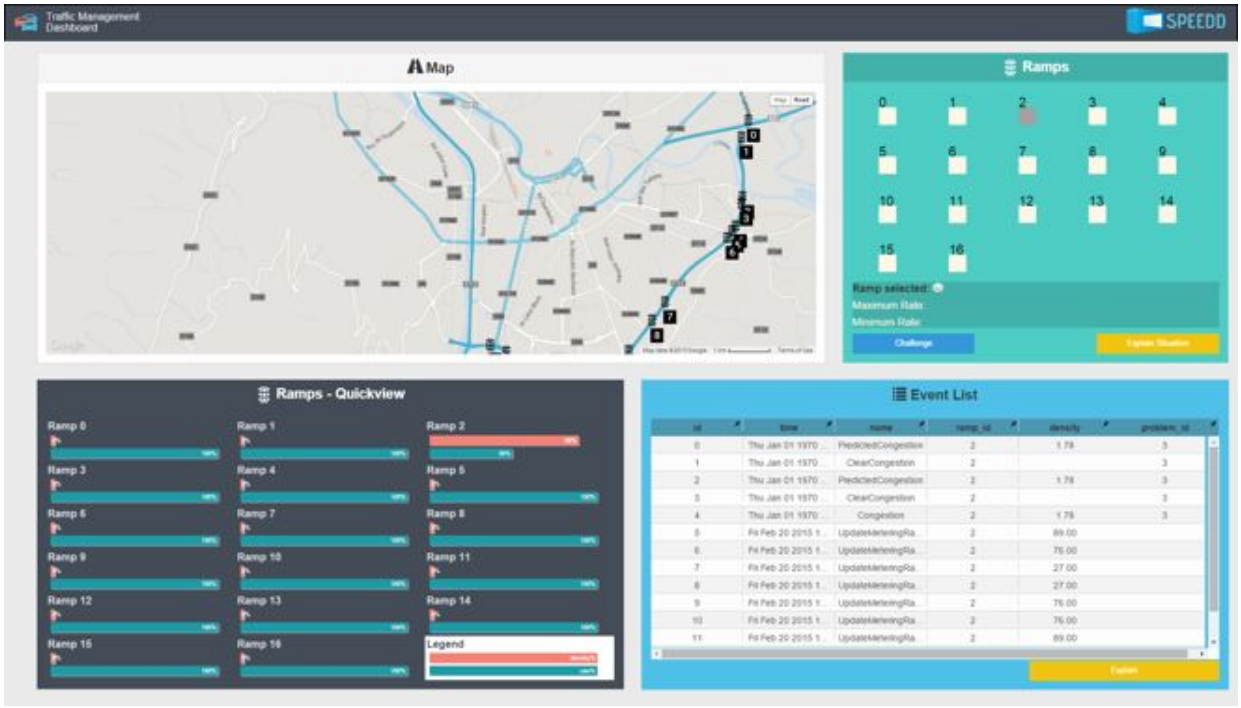


Figure 1: Dashboard version 1

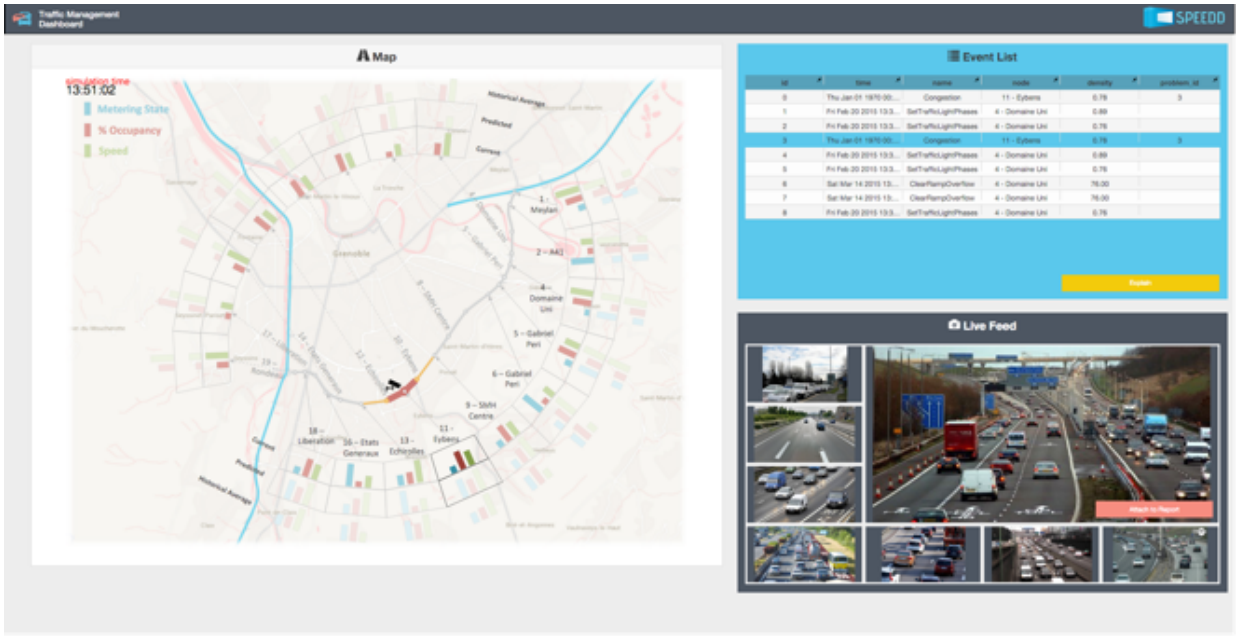


Figure 2: Dashboard version 2

For the experiment, a congestion event relates to the traffic on main artery. The appearance of this event signals a build-up of traffic in the vicinity of a ramp. In this case, the operator needs to limit the number of cars that can enter the main road. Therefore, a congestion event requires that the rate of the inbound ramp in closest vicinity to the alert is low. In the first version of the dashboard, congestion is shown by an orange circle on the map (Figure 3).

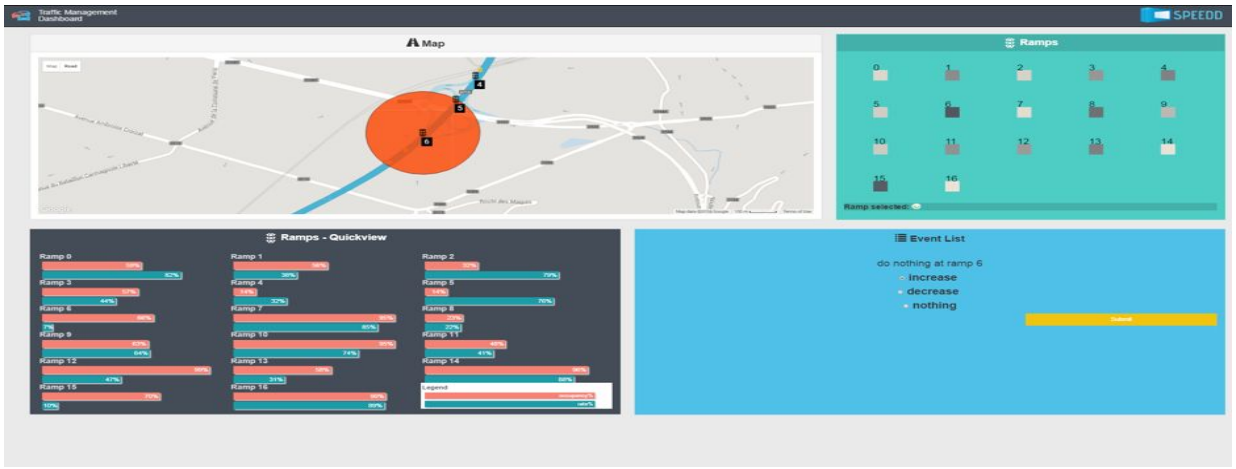


Figure 3: Congestion view in Dashboard 1

In the second version of the dashboard, congestion is shown by an increased width and a red colouring of the portion of the road in question (Figure 4).

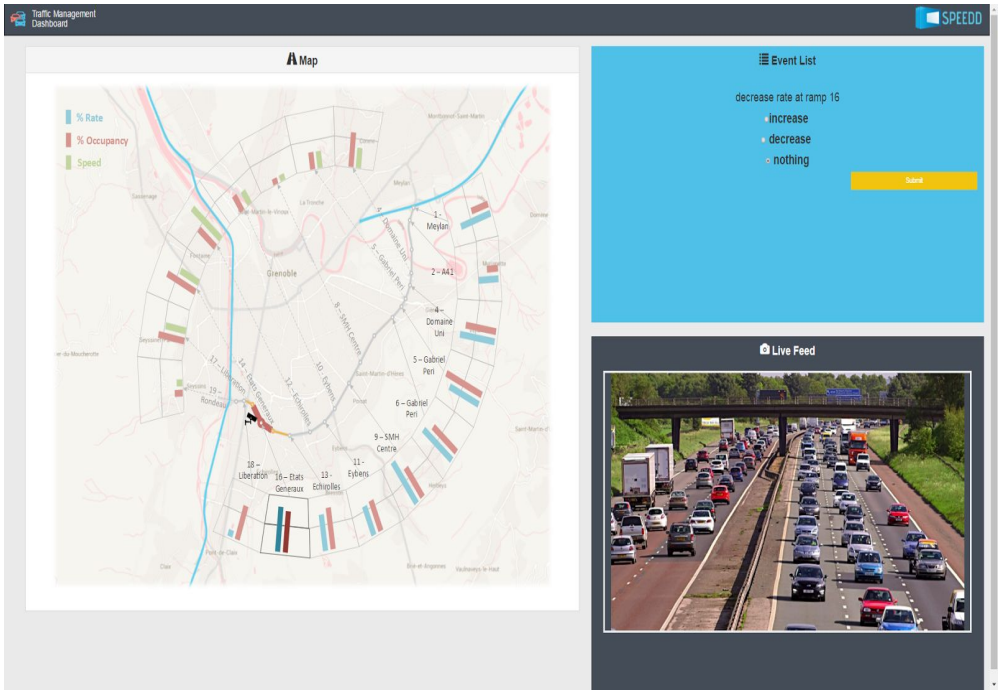


Figure 4: Congestion view in Dashboard 2

Overflow is defined as the build-up of traffic on one of the inbound ramps leading to the main road. In the case of an overflow alert, the operator is required to increase the amount of cars that are able to join the main artery, provided that there is no congestion at that location. Therefore, the overflow event requires that the metering rate at the ramp in question is high. Overflow is signalled by a value of over 50% of the density bar. In the first dashboard the density bar is in the "Ramps – Quickview" window (orange bar, Figure 5), while in the second dashboard, density is represented by a red bar on the map (Figure 6).

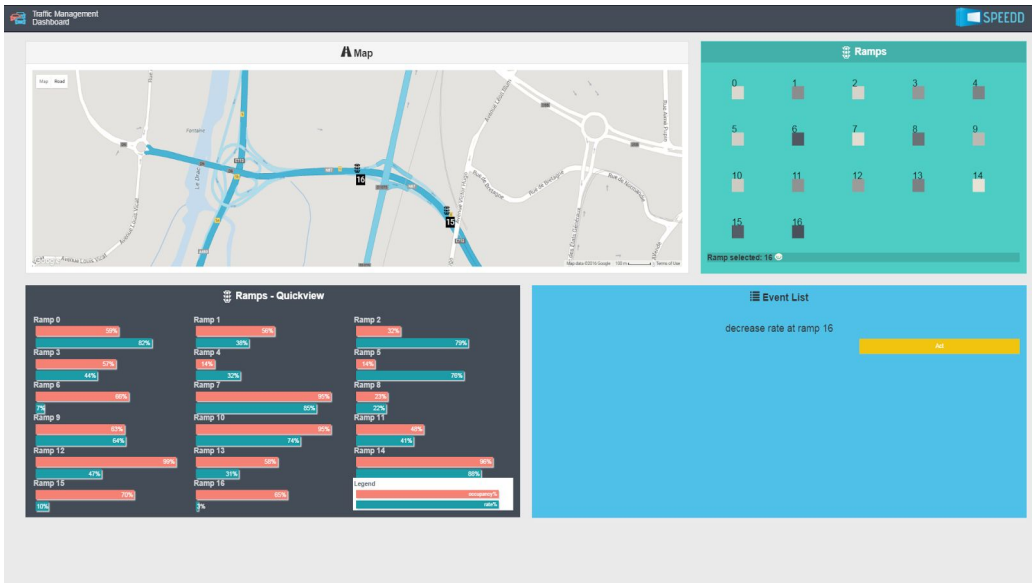


Figure 5: Overflow view in Dashboard 1

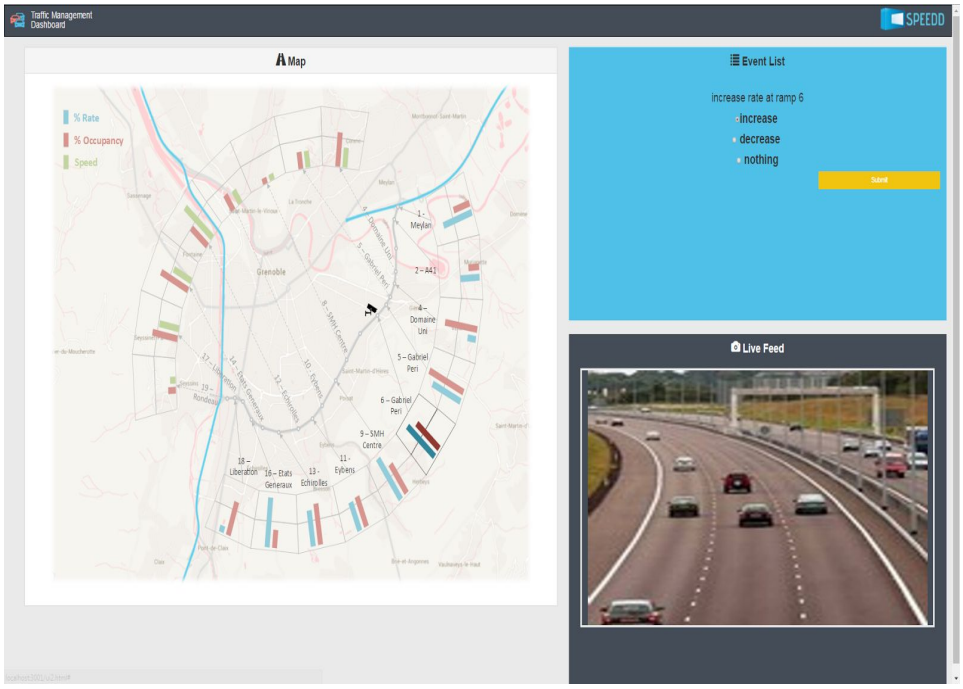


Figure 6: Overflow view in Dashboard 2

Each new event was triggered by a computer generated message appearing in the event list window (figures 7 and 8). This message was a recommendation of whether to increase, decrease or leave the metering rate at a particular ramp unchanged and simulated the output of an automated system which gives operators suggestions on the best course of action given a detected event.

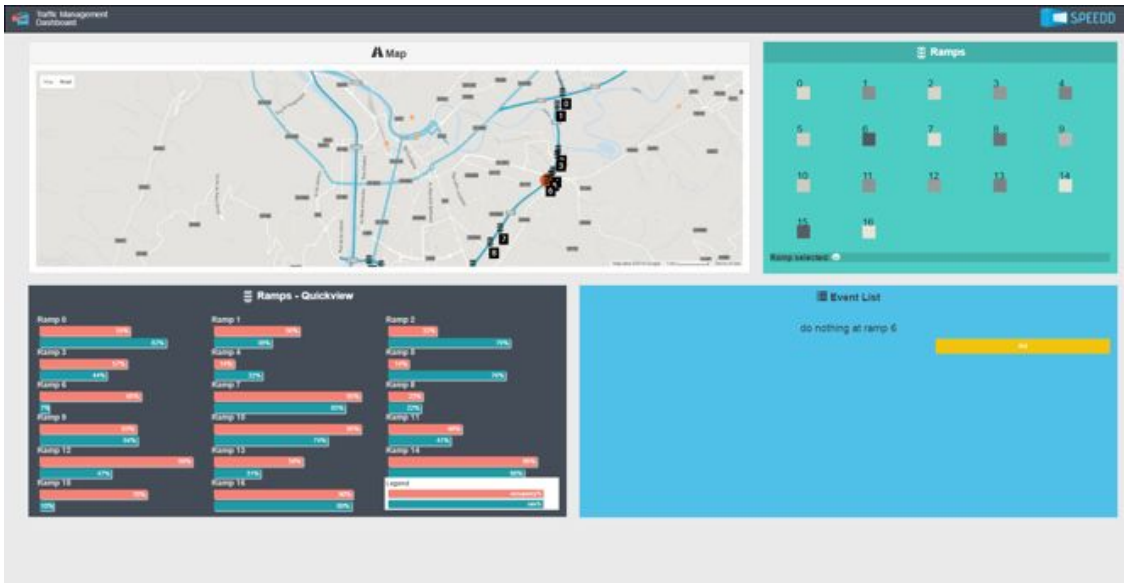


Figure 7: Event list updated in Dashboard 1

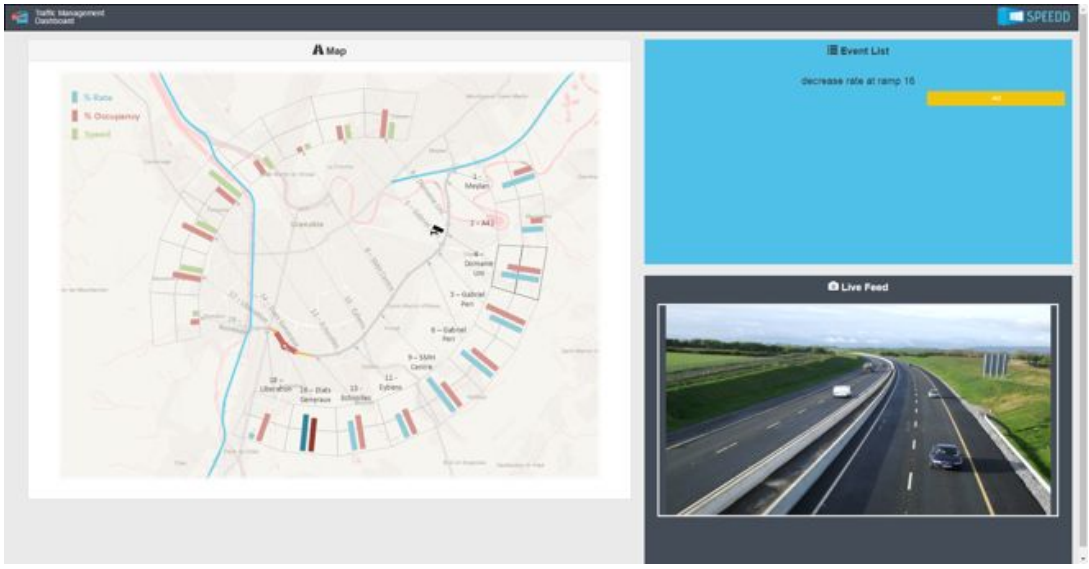


Figure 8: Event list updated in Dashboard 2

3 Experimental Comparison of Dashboards

3.1 Introduction

An experiment was devised in order to test how performance and overall user behaviour differs while using the two different user interfaces and also in response to varying degrees of computer reliability.

3.2 Method

3.2.1 Task

The experimental task was developed around a realistic Traffic Management scenario. The participants had to respond to two types of alerts (or events) presented by the automatic system: congestion and overflow. For simplicity, ramp metering rate was equated with the frequency that cars are able to pass at a traffic light so that a high rate means that cars can pass quicker than on a low metering rate. A low metering rate is defined as a value below 50%, while a value above this mark is considered to be a high rate. In dashboard 1, the current ramp metering rate is shown by the green bar in the "Quickview" window (Figure 7) and in dashboard 2, by the blue bar in the map window (Figure 8).

3.2.2 Procedure

Participants were given a briefing on the experimental task followed by instructions on how to use the interfaces. Participants then began a practice session in order to familiarise themselves with the user interfaces. The practice session consisted of 10 trials, 5 with each user interface. The practice trials were in the same format as the main experiment, but generated randomly for each participant. The computer reliability level was set to 50%. During this session, participants were encouraged to ask any clarifying questions regarding both, the user interfaces and the experimental task.

Two independent variables were defined: 1) the user interface that was used to complete the task, and 2) the reliability of the automated system that presented the participants with the suggestion of what action to be performed. To better control the experiment, users were required to respond to one event at a time. This lead to only one computer suggestion being shown in the event list, for both dashboards, and one CCTV view in the case of dashboard 2, compared to multiple views in the initial interface.

Automation reliability was split into three levels: low (20%), medium (50%) and high (80%). The reliability levels related to the amount of computer suggestions that were correct. Therefore, in the high reliability condition 80% of the suggested actions were correct solutions to the events that were presented in that condition.

The main experiment consisted of 60 trials and was split into six blocks, 10 trials per block. Each block represented one of the possible combinations of the two user interfaces and the three computer reliability levels. Participants were given a 10-second break between each block.

The start of each new trial was signalled by a computer suggestion appearing in the Event List window. The message consisted of a recommended action and the number of the ramp controller in

question. In order to validate the computer suggestion, participants had to identify whether the event was of a congestion or an overflow type. The users then had to decide whether to increase, decrease or leave the ramp metering rates unchanged, depending on the current rate levels as seen in table 1. Participants were allowed to use this table for reference throughout the experiment. This bypassed the need of memorising the rules and was also in accord with the information that Traffic Operators gave us, more specifically that the procedures for taking action are fixed and there is very little, if any, variability when making a control action. When the user was ready to give a response, he would click on the "Act" button present in the event list window. This revealed a list of the possible actions (in the form of a radio buttons list) to take in regards to the metering rate (i.e. increase, decrease, nothing). The user would then select their answer and press the "Submit" button below the list. This signified the end of the trial and the beginning of a new one.

Table 1: correct responses in terms of event and rate level

	Congestion	**Overflow**
Low Rate	do nothing	increase rate
High Rate	decrease rate	do nothing

Five dependent variables were defined in terms of the two user interface versions and the three reliability levels: average time to make a decision, average time to act, average time to submit a decision, % correct decisions and % match decisions. The time to make a decision was calculated from the beginning of the trial (i.e. the appearance of a computer suggestion) up to when the final decision has been made by pressing the "Submit" button. Time to act was defined as the time from when the trial began to the time when the user pressed the "Act" button. The time to submit a decision was defined as the interval beginning with the press of the "Act" button and ending with the press of the "Submit" button. The % correct decisions metric was calculated as the proportion of total decisions in each block that were correct and the % match was defined as the proportion of user decisions that matched the computer recommendation.

In this experiment, a commonly used subjective workload measure was employed. This is the NASA TLX (Task Load Index) [1]. This is a rating scale with six workload dimensions. It can be administered in either a computer or paper based format. We used the paper and pencil version of the test[1]. The rating scales are presented as questions that the participants scores on a scale of 1 (low) to 20 (high). The questions relate to mental demand, physical demand, temporal demand, effort, performance and frustration (figure 9).

[1]

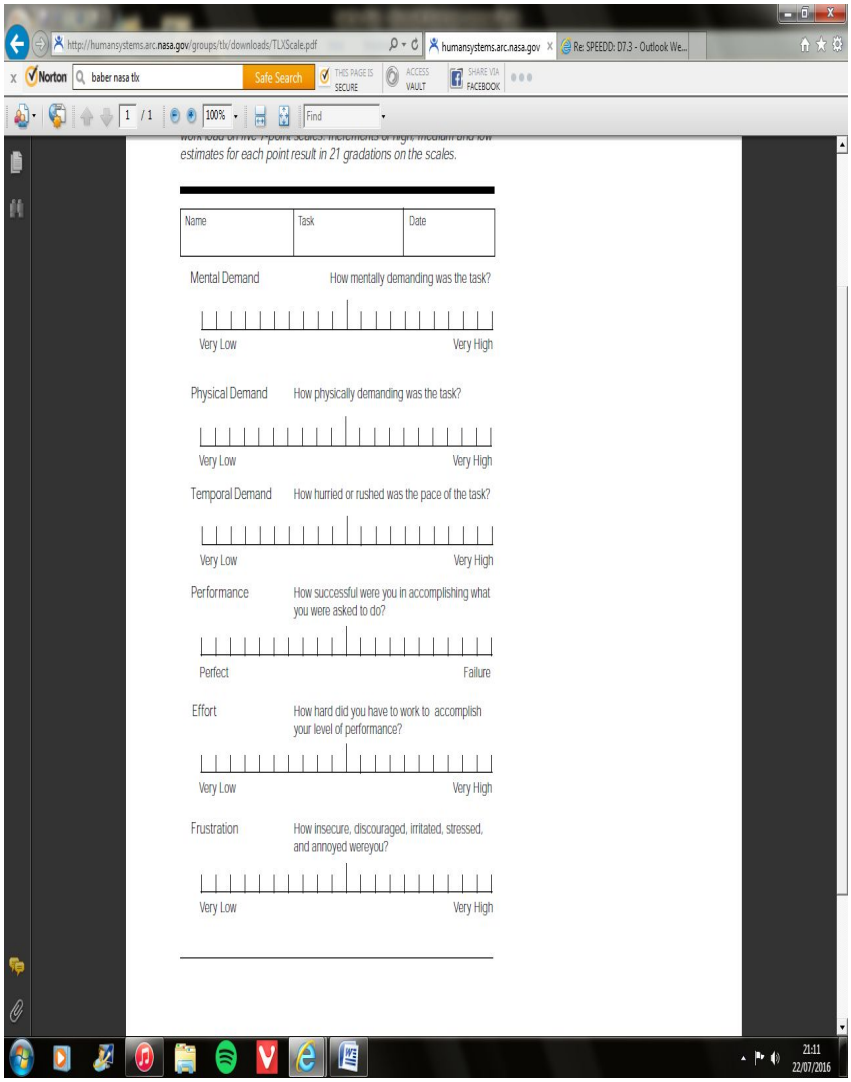


Figure 9: NASA TLX rating form

[http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html]

3.2.3 Participants

24 people took part in the experiment [13: male; 11: female; age range: 22-29]. None of the participants had experience of working in Traffic Management. It was considered that there was no need to include domain experts as participants in the experiment because the task of ramp metering control had not yet been adopted in the control centre the project partnered with at the time of writing (see section 1.1). Therefore, the traffic managers did not have any expertise in the task of controlling metering rates and they would have had to undergo training. Considering that the traffic experts are busy, and that there is only a small number of them on shift at any one time, training non-experts was deemed a good alternative.

3.2.4 Data Gathering, Storage and Analysis

The experiment met University of Birmingham ethics approval. All data were anonymised and participants provided informed consent.

The user interfaces for both, the practice session and the main experiment, were displayed on a 22" monitor (1920x1080 resolution) and the interaction was achieved using a standard mouse.

For each trial the following data were gathered: trial start time, trial end time, act button press time, submit button press time, trial number, user decision, computer suggestion, block number, dashboard version and the following derived metrics: act interval, submit interval, total trial duration, user-computer decision match and user decision correctness.

Data for each participant was stored in a separate Comma Separated Variables (csv) file on a secure University server. Pre-processing was carried out on each participant data in order to remove outlier trials (trials which took very long to respond to) where participants may have been engaged in other tasks or where clarifying questions have been asked. A thresholding of mean + 1 s.d. was performed on the total trial duration. This resulted in the exclusion of 13.68% of the total number of trials. An Analysis of Variance was then performed on the remaining data.

3.3 Results

3.3.1 Mean Decision Time

Mean decision time was defined as the average time needed to make a decision, in other words, to complete a trial. When comparing the two user interfaces we identified a large, main effect of the dashboard version [F (1,23) = 142.987; p = 0.0; Partial Eta squared = .861]. Figure 10 shows that time for dashboard 2 was faster than for dashboard 1. Pairwise comparisons (using t-test) showed significant differences between dashboards under all computer reliability conditions (i.e., for D1Low vs D2Low: t(23) = 7.534, p<0.0001; for D1med vs D2med: t(23) = 7.586, p<0.0001; for D1highvs D2 high: t(23) = 6.858, p<0.0001).

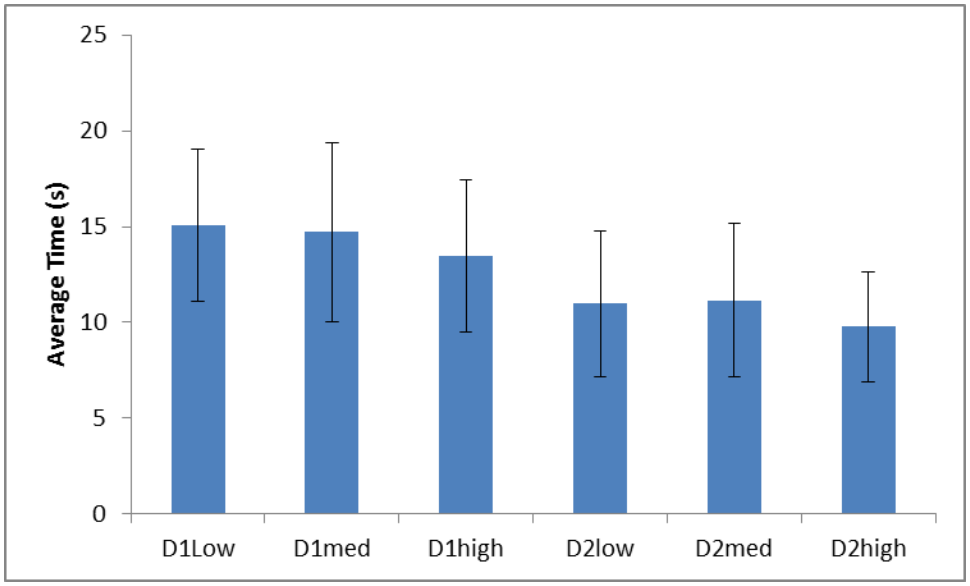


Figure 10: Average Time to complete the task using the two dashboards under different levels of computer reliability

We also found a large, main effect of reliability [F (2,46) = 4.609; p = 0.015; Partial Eta squared = .167].

Figure 10 shows that, while time for low and medium reliability were similar, the time for high reliability was faster. A pairwise analysis was carried out and significant differences found between high and low or medium reliability levels were found for dashboard 1 (i.e., D1low vs D1 high: $t(23) = 2.748$, $p<0.05$; D1med vs D1 high: $t(23) = 2.146$, $p<0.05$), and significant difference between medium and high reliability for dashboard 2 (i.e., D2med vs D2 high: $t(23) = 2.753$, $p<0.05$).

3.3.2 Mean Time to Act

Time to act was defined as the interval between the start of a trial and the time the user pressed the "Act" button which revealed the answer options. A main effect of dashboard version was found [$F(1,23) = 39.573$; $p = 0.0$; $p = 0.0$; Partial Eta squared = .632] (Figure 11). No other effects were discovered.

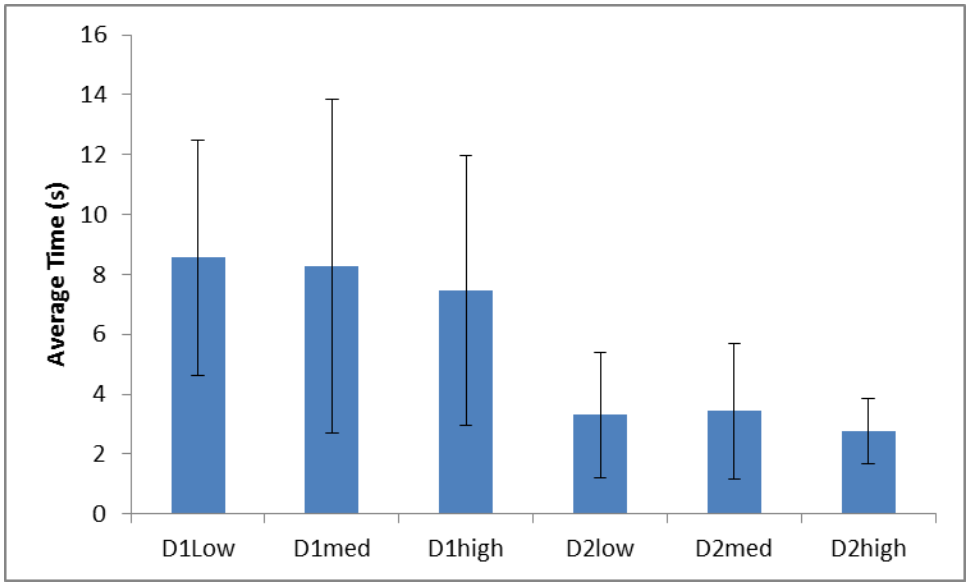


Figure 11: Average time to act for the two dashboards under different levels of computer reliability

3.3.3 Mean Time to Submit Decision

Time to submit was defined as the interval between the time the answer options were revealed to the user up to the time the final decision was submitted. No effects were identified, the mean times for the two user interfaces being 6.3s (std. error = .7s) for dashboard 1 and 7.5s (std. error = .67s), for dashboard 2.

3.3.4 Decision Match

When a user decision was the same as the computer suggestion for a particular trial, we said that a decision match occurred. A significant effect of reliability was found for decision match [$F(2,46) = 272.365$; $p = 0.0$; Partial Eta squared = .922]. This is illustrated in figure 12.

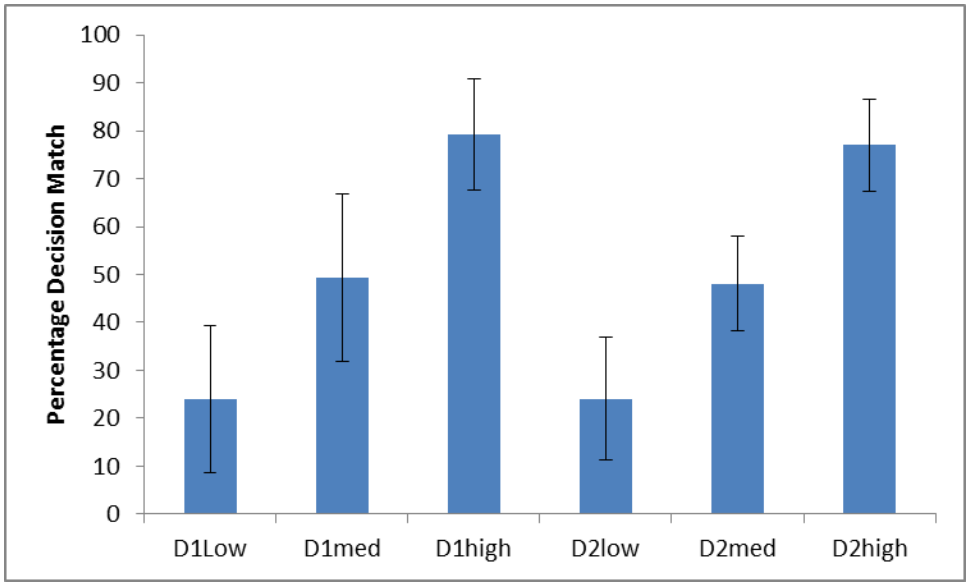


Figure 12: Percentage of User Decisions matching computer recommendation

3.3.5 Decision Correctness

Decision correctness refers to the percentage of correct decisions out of the total number of trials engaged in. Results show a statistically significant main effect of reliability [F (2,46) = 5.025; p = 0.011; Partial Eta squared = .179] but no effect of dashboard. Paired comparisons using t-tests revealed significant differences between high and other levels of reliability (i.e.. D1med vs D1high: t(23) 2.078, p<0.05; D2low vs D2 high: t(23) = 2.689, p<0.05).

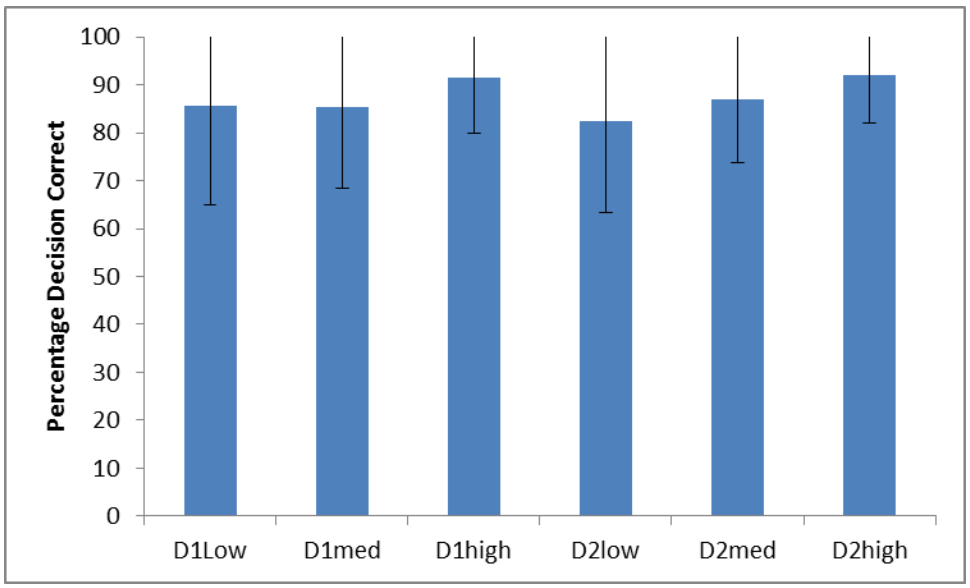


Figure 13: Percentage of user decisions which are correct

3.3.6 Event Type

When splitting the data into the two individual event types (congestion and overflow), a significant

interaction of event type and reliability on the mean time to act was identified [F (2,46) = 4.005; p = 0.025; Partial Eta squared = .148]. Figure 14 shows that, while there is variability in the time to act in terms of the reliability level of the automation for the congestion event, the overflow event shows similar mean times.

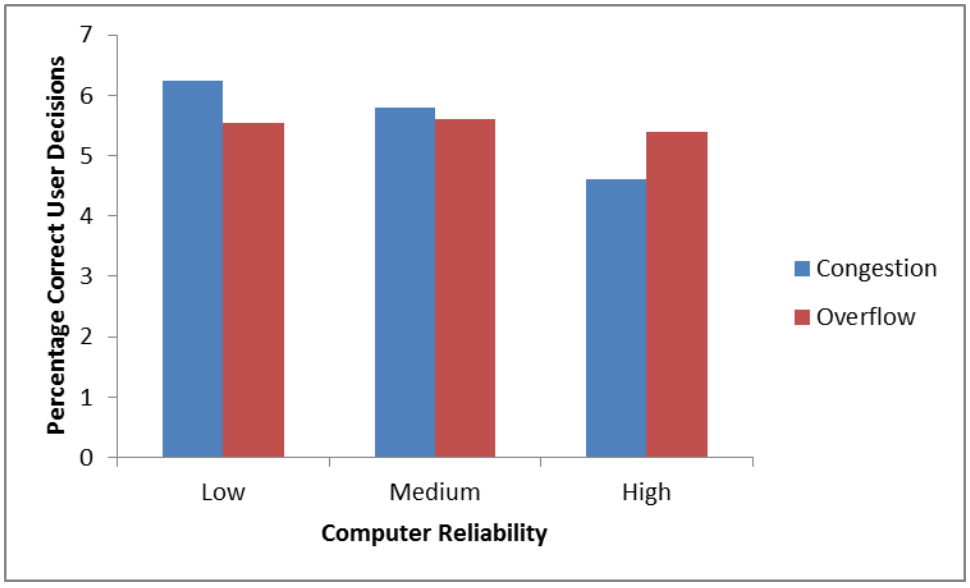


Figure 14: Comparing types of decision against computer reliability

3.3.7 Workload

The NASA TLX questionnaire results show no significant difference in workload between the two user interfaces (Figure 15). The mean score for the first interface was 59.08 (stdev = 18.58), while for the second, 52.25 (stdev = 21.77).

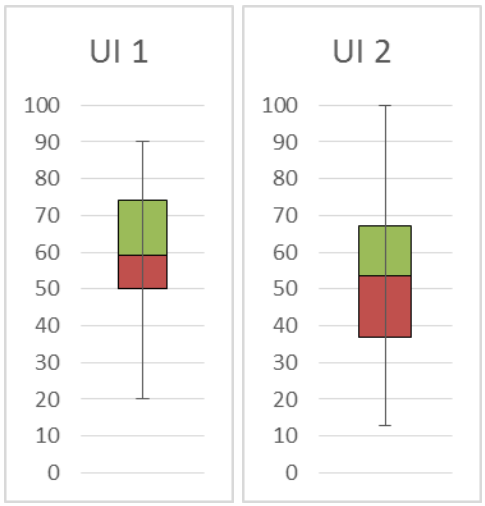


Figure 15: Comparison of workload ratings for the two dashboards

3.4 Discussion

The analysis of the mean decision time exposed a large difference between the two user interface versions. However, in order to explore this effect further, we look at the two components of total decision time (i.e. time to act and time to submit), an interesting effect can be spotted. Time to submit is similar across all conditions (see section 3.3.3), but there was a significant difference between dashboards in terms of time to act (see section 3.3.2), with a difference between means of approximately 5 seconds. This suggests that the two intervals (act and submit) relate to two distinct stages in operator decision-making. The first one, the act interval, is the information gathering stage, while the submit interval, the final checking and response submission stage. Assuming that this is what is actually happening, then dashboard 2 speeds up the process of gathering information. In comparison with the nominal baseline data (section 2) which suggested that current practice takes around 15s to 20s to respond to an alert, the total decision time (section 3.3.1) is 13s to 15s for dashboard 1 and 9.8s to 11.1s for dashboard 2. This suggests that dashboard 1 could provide response times comparable with current practice, and dashboard 2 could lead to faster response times. While this implies a benefit of the design of dashboard2, one needs to remember that speed of response is not a necessary or sufficient criterion for operators. It is more important to ensure that the decisions are accurate and complete.

User performance seems to improve as automation reliability increases. Section 3.3.4 shows that users were able to interpret computer reliability to a high degree of accuracy (even when there was no specific indication to user about reliability). Participants achieve mean match levels of 24.02% (std. error = 2.47), 48.67% (std. error = 2.32) and 78.25% (std. error = 1.85) for the low, medium and high reliability condition, respectively. This illustrates that participants are able to accurately determine whether they should follow the computer recommendation, considering that the computer's reliability level was set at 20, 50 and 80% for the low, medium and high condition, respectively. Section 3.3.5 shows that decision accuracy was comparable across dashboards, but was affected by the computer reliability. The significant main effect of reliability on decision time could point to the fact that users are able to distinguish between the different reliability levels, resulting in a more cautious approach to decision-making in the low and medium reliability conditions.

4 Quantitative evaluation of other SPEEDD components for the Traffic use case

In this section, we give some results related to the quantitave evaluation of the remaining components of the SPEEDD prototype.

4.1 Evaluation of patterns learning

We applied OSL (see Deliverable D3.3) to traffic management using real data from the magnetic sensors mounted on the Grenoble South Ring, consisting of approximately 3.3GiB of sensor readings (one month data). Annotations of traffic congestion are provided by human traffic controllers, but only very sparsely. To deal with this issue, we also used a synthetic dataset generated by the traffic micro-simulator. The synthetic dataset concerns the same location and consists of 6 simulations of one hour each (≈ 18.6MiB).

A set of first-order logic functions is used to discretize the numerical data (speed, occupancy) and produce input events such as, for instance, HappensAt(fastSlt55(53708), 100), representing that the speed in the fast lane of location 53708 is less than 55 km/hour at time 100. The total length of the training sequence in the real data case consists of 172,799 time-points, while in the synthetic data it consists of 238 time-points. The evaluation results were obtained using MAP inference [Huynh and Mooney 2009] and are presented in terms of F_1 score. In the real dataset, all reported statistics are micro-averaged over the instances of recognized CEs using 10-fold cross validation over the entire dataset, using varying batch sizes. At each fold, an interval of 17,280 time-points was left out and used for testing. In the synthetic data, the reported statistics are micro-averaged using 6-fold cross validation over 6 simulations by leaving one out for testing.

Fig. 16 presents the experimental results on the real dataset. We compare OSL against the AdaGrad online weight learner [Duchi et al. 2011] that optimizes the weights of a manually constructed traffic congestion definition.

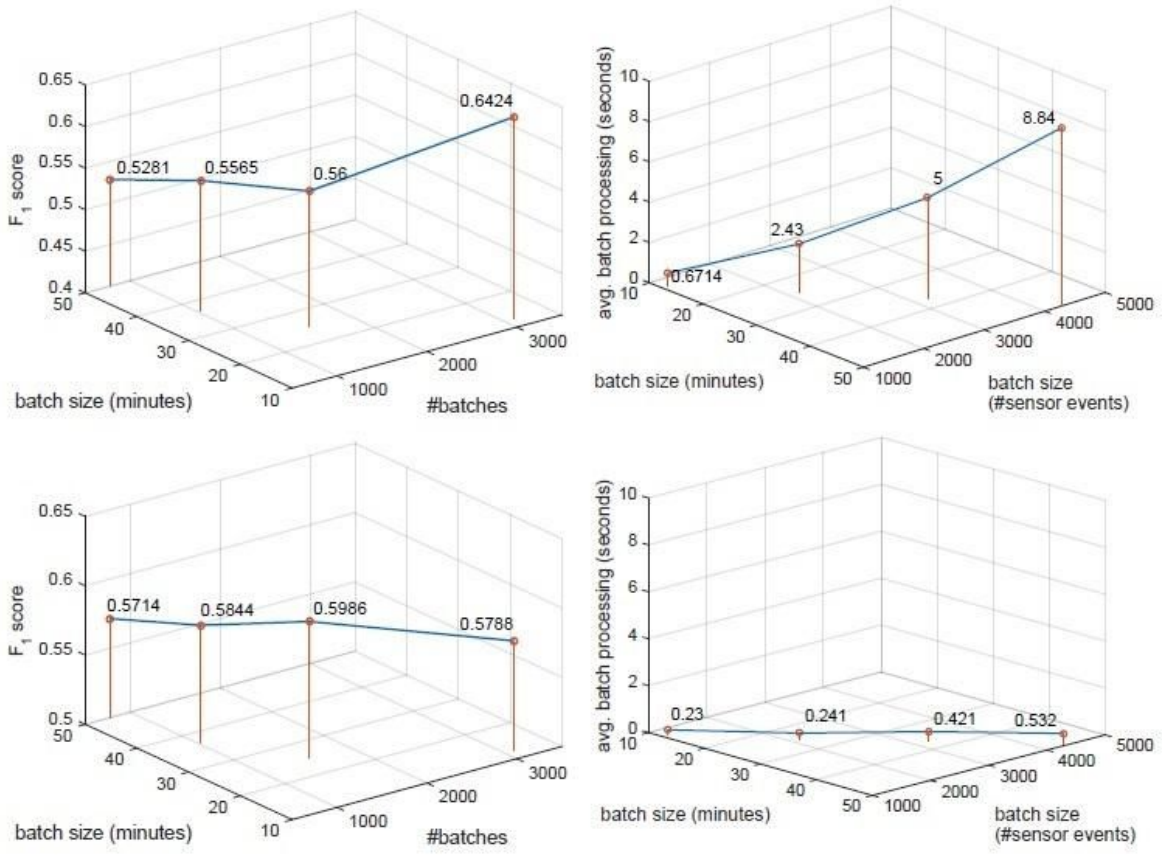


Figure 16: Real dataset: F_1 score (left) and average batch processing time (right) for OSLα (top), and AdaGrad operating on manually constructed traffic congestion rules (bottom). In the left figures, the number of batches (see the *Y* axes) refers to number of learning steps.

The predictive accuracy of the learned models, both for OSL and AdaGrad, is low. This arises mainly from the largely incomplete supervision. In OSL , the predictive accuracy increases (almost) monotonically as the learning steps increase. On the contrary, the accuracy of AdaGrad is more or less constant. OSL outperforms AdaGrad in terms of accuracy. (OSL achieves a 0.64 F_1 score, while the best score of AdaGrad is 0.59.) This is a notable result. The absence of proper supervision penalizes the hand-crafted rules, compromising the accuracy of AdaGrad that uses them. OSL is not penalized in this way, and is able to construct rules with a better fit in the data, given enough learning steps. For some locations of the highway, OSL has constructed rules with different thresholds for speed and occupancy than those of the hand-crafted rules. With respect to efficiency (see the right diagrams of Fig. 9), unsurprisingly AdaGrad is faster and scales better to the increase in the batch size. At the same time, OSL processes data batches efficiently, much faster than their duration. For example, OSL takes less than 9 sec to process a 50-minute batch including 4,220 sensor readings.

To test the behavior of OSL under better supervision, we made use of a synthetic dataset produced by the traffic micro-simulator of GTL. Fig. 17 presents the experimental results. Not surprisingly, the

predictive accuracy of the learned models in these experiments is much higher as compared to real dataset. Moreover, the accuracy of OSL and AdaGrad is affected mostly by the batch size: accuracy increases as the batch size increases. The synthetic dataset is smaller than the real dataset and thus, as the batch size decreases, the number of learning steps is not large enough to improve accuracy. The best performance of OSL and AdaGrad is almost the same

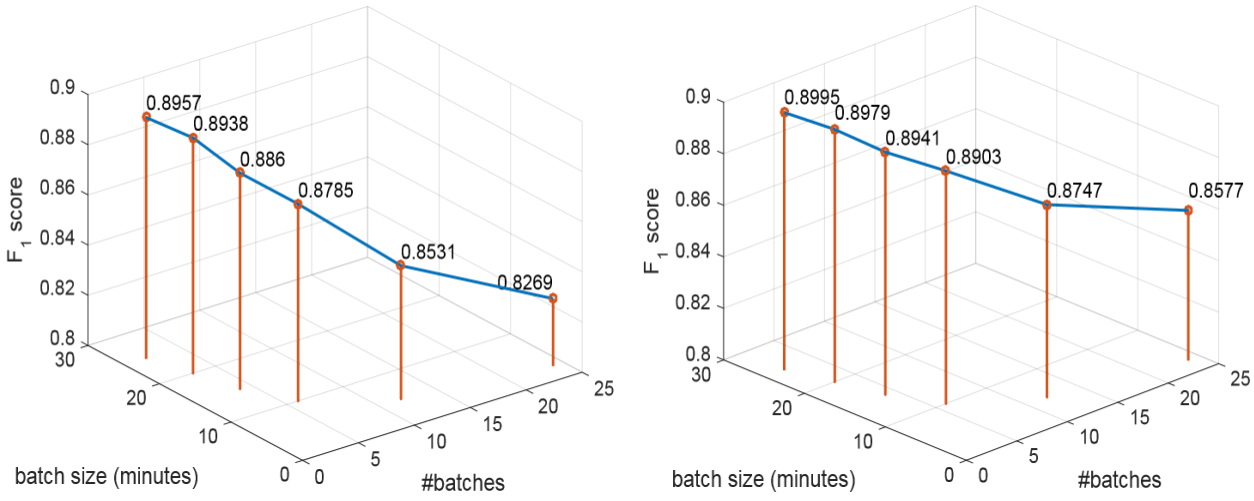


Figure 17: Synthetic dataset: F1 score for OSLα (left) and AdaGrad operating on manually constructed traffic congestion rules (right).

(approximately 0.89). In other words, OSL can match the performance of techniques taking advantage of rules crafted by human experts. This is another notable result.

4.2 Event forecasting

In order to explore the quality of our CEP module, we ran a test comprising of 20 simulations generated by the traffic micro-simualtor of GTL along with annotations of congestions. The annotations of congestions include the location and the time the congestion is detected. First, we evaluated the quality of our Congestion pattern against the annotated data. We checked the proportion of detections by our EPA that were annotated in the data as congestions (precision) and second, the proportion of congestions we were able to detect out of all the annotated congestions (recall). In all our simulations our precision was 100%, while the average recall over all the simulations was 72%. This can be easily explained: the rule implemented has been given to us by the domain expert, who is the one to identify the congestions in the simulations, thus giving a perfect precision. However, when implementing the pattern we applied a “stricter” criterion for the rule than the one in the simulator: we took into account not just the average speed critical thresholds, but also density thresholds, therefore we have a less success rate in the recall of the results, i.e., there were annotations of congestion in the data that we “missed”.

As a second step, we aimed at checking a more interesting question, that is, whether the inclusion of uncertainty aspects enables us to predict a congestion in the highway before it reaches critical thresholds, as opposed to detecting it once it happens. We addressed this question by having two EPNs, once including uncertainty aspects and the other one without uncertainty, i.e. deterministic;

and running the tests twice, one time for each EPN (with and without uncertainty). This is a common approach in CEP engines dealing with uncertainty (see for example in [Cugola et al. 2014]). The deterministic case served as baseline, as we knew at this stage that all our congestions have been detected correctly. The precision of our results indicates the proportion of congestions we were able to predict (in other words, PredictedCongestion pointed out correctly to a congestion), whereas the recall indicates the proportion of congestions we were able to detect out of all the annotated congestions (in other words, PredictedCongestion pointed out correctly out of all congestions). We used a threshold of 0.6 in the certainty attribute to determine whether to consider PredictedCongestion as a congestion. In other words, only PredictedCongestion alerts with a certainty value larger than 0.6 were considered in our calculations of precision and recall. In these tests, the average precision was 91% and the average recall was 75%. Furthermore, PredictedCongestion event is emitted 3 to 4 minutes before a Congestion is detected, thus enabling the system to take proactive actions in order to alleviate these congestions. The recall average indicates that there are other situations that cause congestions which are not detected by our pattern. Further analysis shows that these situations are characterized by "jumping data", meaning, the values of speed and density tend to jump thus not satisfying the increasing build-up which is required in our pattern. We are currently investigating these "jumping" cases to see if we can identify some common behavior/pattern.

4.3 Decision making

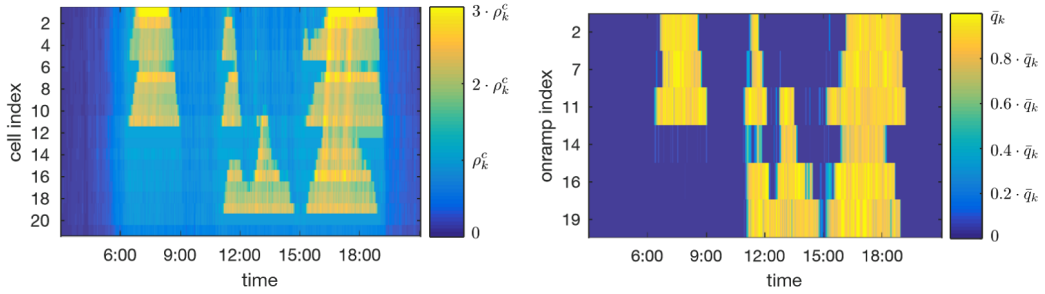
To quantify the benefits of ramp metering, we use the Total Time Spent (TTS), a standard metric defined as the sum of the travel times of all cars for a certain day. We perform three types of simulations. First, we simulate traffic without ramp metering to obtain a baseline performance, TTS_{ol}. Second, simulations using local ramp metering as described in Section 4.2 are performed, but no coordination between ramps is used. The corresponding travel time is denoted TTS_{cl}. Third, we employ coordinated ramp metering with the coordination along the lines of Section 4.3 and denote the corresponding total time spent as TTS_{co}. The parameters of the coordination are chosen as $_3$ = 0.1 and $_4$ = 0.2. For the five-week period, we obtain relative savings of

$$\frac{\mathrm{TTS}_{ol} - \mathrm{TTS}_{cl}}{\mathrm{TTS}_{ol}} = 9.9\% \quad \text{and} \quad \frac{\mathrm{TTS}_{ol} - \mathrm{TTS}_{co}}{\mathrm{TTS}_{ol}} = 13.6\%.$$

Benefits of coordination tend to increase as traffic demand increases, while conversely, no benefits are obtained on days with no or only light congestion for an uncontrolled freeway. However, TTS does not only quantify time wasted in congestion and in onramp queues, but vehicles traveling at free-flow velocity contribute significantly as well. Ramp metering cannot provide any benefits during times at which the uncontrolled freeway is not congested. Therefore, we define the Total-Free-flow-Time TFT as the travel time accumulated by all vehicles on a hypothetical freeway, that is always uncongested, i.e. all vehicles travel at free-flow velocity at all times. The relative savings in terms of time wasted in congestion and in on-ramp queues for all days amount to

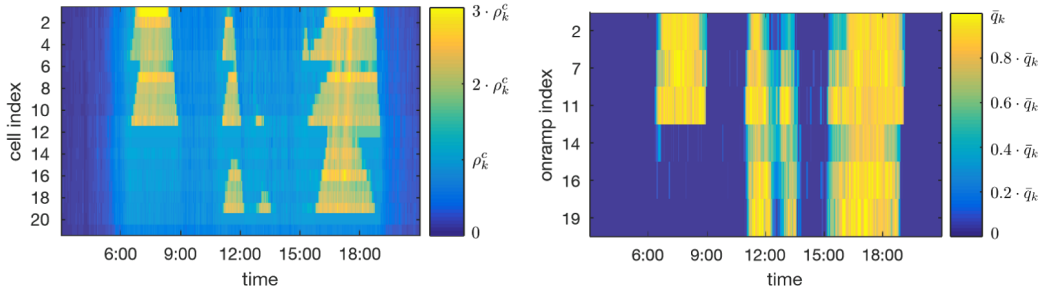
$$\frac{\mathrm{TTS}_{ol} - \mathrm{TTS}_{cl}}{\mathrm{TTS}_{ol}} = 9.9\% \quad \text{and} \quad \frac{\mathrm{TTS}_{ol} - \mathrm{TTS}_{co}}{\mathrm{TTS}_{ol}} = 13.6\%.$$

The results are visualized in Figure 18 for one day that provides average savings. It should be noted that both the time spent in congestion and the relative savings of coordination seem to be sensitive to the traffic demands. In a non-monotonic setting, small changes in demands may cause large differences in open- or closed-loop behavior. Coordination tends to provide larger relative savings for more severe congestion.



(a) Density for local control.

(b) Queue occupancies for local control.



(c) Density for coordinated control.

(d) Queue occupancies for coordinated control.

Figure 18: Simulation results for traffic demands of April 19th, 2014. Coordination distributes vehicles among on-ramps, thereby reducing traffic on the mainline and increasing the bottleneck flows, in particular for cell 11 and 19.

5 Expert evaluation of SPEEDD prototype for the Traffic use case

SPEEDD prototype was presented to Mr Philippe MANSUY, Director of operations of DIR CE, which is in charge of the Grenoble south ring. The functionalities of the prototype was presented and the performance of the overall system tested. Here after, we reproduce the comments of Mr. Mansuy.

The outcomes of SPEEDD are particularly relevant for the traffic use case for two reasons: efficient design for traffic monitoring interface and proactivity for decision making support. In my opinion, SPEEDD has achieved results that can be helpful to improve the current state of traffic control rooms in terms of dashboard / display design, and opens a new paradigm for decision making in traffic control by introducing proactivity while taking uncertainties into account.

In the DIRCE control room, two profiles of traffic operator behaviour can be found: in one, people focus on CCTV cameras, and in the other people focus on a map which uses colour coding to show on traffic speed. CCTV provides a timely and efficient local view of the infrastructure, but does not give the overall picture of the traffic situation across the road network. The color-map provides a view of the full infrastructure but it has an inherent delay due to data aggregation (3 minutes for our current setup). I'm particularly attached to the dashboard developed by SPEEDD since it combines the advantages of both approaches, providing a complete view of the infrastructure, in a timely manner, and allowing to zoom on local views where needed. I am also interested in the suggestion that SPEEDD could make predictions concerning congestion some 4 minutes from the current situation.

Currently, we are deploying ramp metering to regulate traffic on the Grenoble south ring. The adopted process will be fully automated reacting to variations of traffic density. The task of the operator will be restricted to switching off and on the control module. The SPEEDD project has presented an improved decision making process based on predicted events such as upcoming congestions. I believe that such predictions could be also very useful in other scenarios both for improving fully automated systems and for helping human operators. However, the impact of this approach could be even more important if we consider the following two scenarios.

The first example concerns traffic control using variable speed limits (VSL). We made experiments where operators were asked to act on VSL panels in order to prevent congestions. Unfortunately, decisions were taken too late. By being able to predict the occurrence of congestion sufficiently in advance, SPEEDD could be helpful as a decision making support for operators in such a case.

The second example concerns access control of critical infrastructures. For instance, in our network we have a road leading to ski resorts, where it is critical to avoid congestion in some portions of the road, namely a tunnel and two road stretches bordered by dangerous cliffs. The dilemma is to ensure free flow traffic while optimizing the use of the infrastructure whatever the demand. In contrast to the south ring, here having human operators in the loop is necessary. SPEEDD could help operators by

suggesting optimal actions to increase or decrease the light rates while taking into account uncertainties due to split ratios between two main ski resorts.

In conclusion, our participation in the SPEEDD project has helped with our thinking about how decision making in traffic control can be supported. It has done this by showing how to improve the presentation of information to operators and by demonstrating a novel paradigm for proactive decision making in traffic control which is able to take uncertainties into account.

These positive comments confirm the relevance of SPEEDD prototype and suggests that there is scope here for creating useful products to support road traffic control.

6 References

[1.] Hart, S. G., & Staveland, L. E. (1988) Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research, *Advances in psychology*, *52*, 139-183.
[2] G. Cugola, A. Margara, M. Matteucci, and G. Tamburrelli (2014). Introducing uncertainty in complex event processing: model, implementation, and validation. Computing (2014), 1–42.
[3] J. Duchi, E. Hazan, and Y. Singer. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research 12 (July 2011), 2121–2159.

/