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

| | |
|---|---|
| Project | **FP6-619435/SPEEDD** |
| Deliverable | **D7.3** |
| Distribution | **Public** |

http://speedd-project.eu

# Interim Evaluation Report

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

Ivo Correira
(Feedzai)

Status: Revised

August 2016

**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 | D7.3 |
| Deliverable Title | Interim Evaluation Report |
| Contractual date of delivery | M13 (March 2016) |
| Actual date of delivery | August 2016 |
| Relevant Task(s) | WP7/Tasks 7.3 |
| Partner Responsible | Feedzai |
| Other contributors | UoB |
| Number of pages | 25 |
| Author(s) | C. Baber, S. Starke, N. Morar and I. Correira, |
| Internal Reviewers | A. Artikis |
| Status & version | Revised |
| Keywords | Evaluation, User Interface Design, Human Factors, Eye Tracking |

# Contents

# Figures

# 1. Executive Summary

This deliverable reports the evaluation of the second user interface design for the SPEEDD Credit Card Fruad use case. The initial prototype, described in D7.1, sought to reflect features that were common in user interface designs in the financial sector. This design was implemented in the first SPEEDD prototype. Following initial evaluation with analysts and laboratory trials (reported in D7.1 and D5.2) the user interface design has been revised. This report considers how these changes have affected perceived usability of the user interface. In addition, the report considers the manner in which a commercial product, produced by Feedzai, is used to support fraud analysis. A small trial, using eye-tracking, considered how analysts employ the Feedzai user interface. This provides further information towards defining baseline performance measures for evaluation of the SPEEDD prototype.

## 1.1 Document Structure

The document is divided in five main parts. Section 2 provides a short introduction to the report. In Section 3, the user requirements presented in D5.1 and revisited in D7.1 and D7.2 are reviewed. Section 4 contains the evaluation of the current user interface design. This involves an informal review by Feedzai personnel followed by the application of the Software Usability Scale (SUS) (Brook, 1988). This scale was used in D8.3 and D7.2. Section 5 considers the ways in which Feedzai employees interact with their current system. The final section reviews baseline metrics which might be beneficial for subsequent evaluation activity.

# 2. Introduction

## 2.1 History of the Document

| Version | Date | Author | Change Description |
|---|---|---|---|
| 0.1 | 22/02/2016 | Chris Baber | First version of the document |
| 0.2 | 25/02/2016 | Ivo Correia | First review from Feedzai |
| 0.3 | 30/03/2016 | Ivo Correia | Augmented evaluation analysis |
| 0.4 | 18/04/2016 | NCSR | Reviewed whole document. |
| 1.0 | 21/04/2016 | Ivo Correia | Final version. |
| 1.1 | 17/07/2016 | Ivo Correia | Updated with reviewers comments. |

## 2.2 Purpose and Scope of Document

The purpose of this document is to report an interim evaluation of the SPEEDD prototype in the credit card fraud management use case. In terms of evaluation, the aim is to show how the prototype should evolve and how it is going to be used by the fraud operators. The target audience of this document will be all parties involved in the implementation of the fraud use case.

## 2.3 Relationship with Other Documents

As noted in the previous section, this document is related to the following deliverables: 7.1 User Requirements, 7.2 Initial Evaluation Report, D5.1 Design of User Interface for SPEEDD Prototype, D5.2 Design of User Interface for SPEEDD Prototype (year 2).

## 2.4 Sources of Information

Information was gathered from the Feedzai personnel, who could provide insight due to their contact with several credit card fraud analysts, providing this way and indirectly, what is expected to find in a user interface built towards fraud management.

# 3. Requirements Revisited

## 3.1 Introduction

For the credit card fraud use case, there continues to be a tension between the need to fully automate (and hence remove) human analysts from the fraud detection process, and the need to support (human) analysts in resolving ambiguous, uncertain or problematic cases. The purpose of the user interface is, therefore, two-fold: to provide information to analysts on the activity of the automated system, and to support human analysts in their decision making as required.

In addition of having to support two different purposes, it is also useful to remember that there are several roles which can be defined as 'fraud analyst'. Each role has different information requirements because each role performs different tasks in different contexts.

Fraud analysts in banks seek to define fraud patterns which can be implemented in automated systems, or to check the reliability of the patterns which automated systems employ. This involves high-level review of activity, with a focus on analysing trends and on a detailed examination of a small number of cases. In contrast, call center agents check the validity of individual transactions and tend to have very high throughput (around 200 cases per day per agent). This involves no analysis beyond confirming details, although well trained staff are beneficial to keeping the fraud level down. In merchant-centred (businesses) analysis, the goal is to prevent charge-backs with the need to review each transaction before accepting it.

In each form of analysis, a suspiciousness score can be used to indicate whether transactions (either individually or collectively) exceed the thresholds defined by specific organisations (e.g., banks, credit card companies, merchants). The definition of thresholds (both in terms of value and in terms of which aspects of a transaction to consider) are closely guarded by the organisations and are not in the public domain. If a transaction exceeds a suspiciousness score, then its details could be displayed to the analyst. Details could include amount, location of transaction, date and time, type of purchase etc. How these details inform the analyst's decision making depend on the type of role they play. In D5.2, a set of these details was used to explore the decision strategies that could be applied.

# 4. Formative Evaluation and Transition from second to third prototype

## 4.1 Introduction

Following from the evaluation in D7.2 (Initial Evaluation) of the user interface for the fraud use case (shown in Figure 1), and number of revisions were made to the presentation of data.

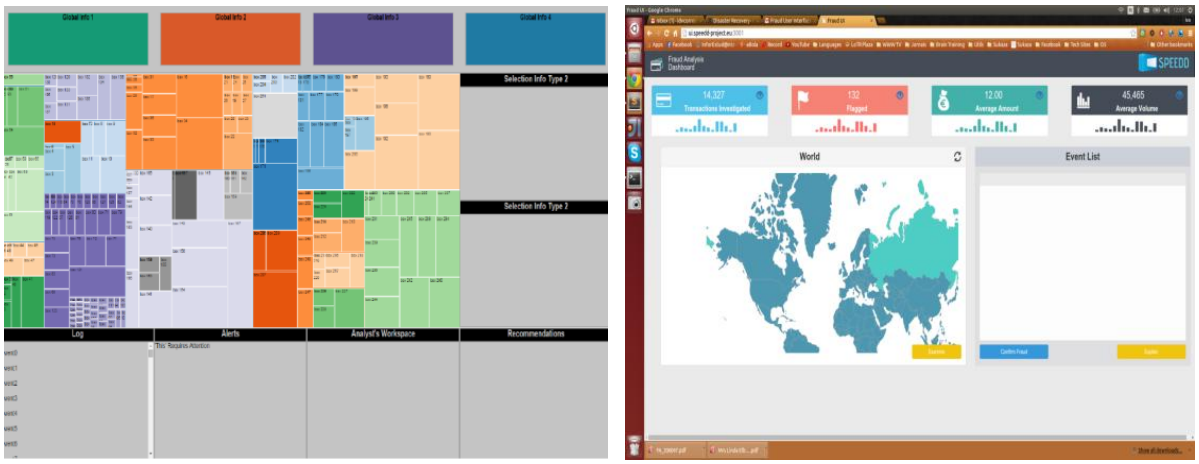


**Figure 1: User Interface designs for (left) version I and (right) version II of the SPEEDD Credit Card Fraud Prototype**

The SUS (System Usability Scale) showed an increase in perceived usability from version I to version II, with version II reaching a level deemed 'acceptable' (i.e., scores in excess of 65). Comments from users suggested that version I related to problems with understanding the scaling and colour used in the tree-map, e.g., it was not intuitive as to what size, colour or resizing indicated. Users seemed to prefer the use of a world map to indicate activity in countries, but were not sure what the bar charts in the top windows indicated (they indicate time in months). In both versions I and II, users asked about the interaction with the display, e.g., in terms of defining search criteria for different fraud patterns or specific transactions, in terms of filtering the event list, in terms of being able to rearrange panels and in terms of the relationship between changes in one panel and information in another. In order to address these comments, the user interface design was modified. The more recent design is shown in Figure 2.

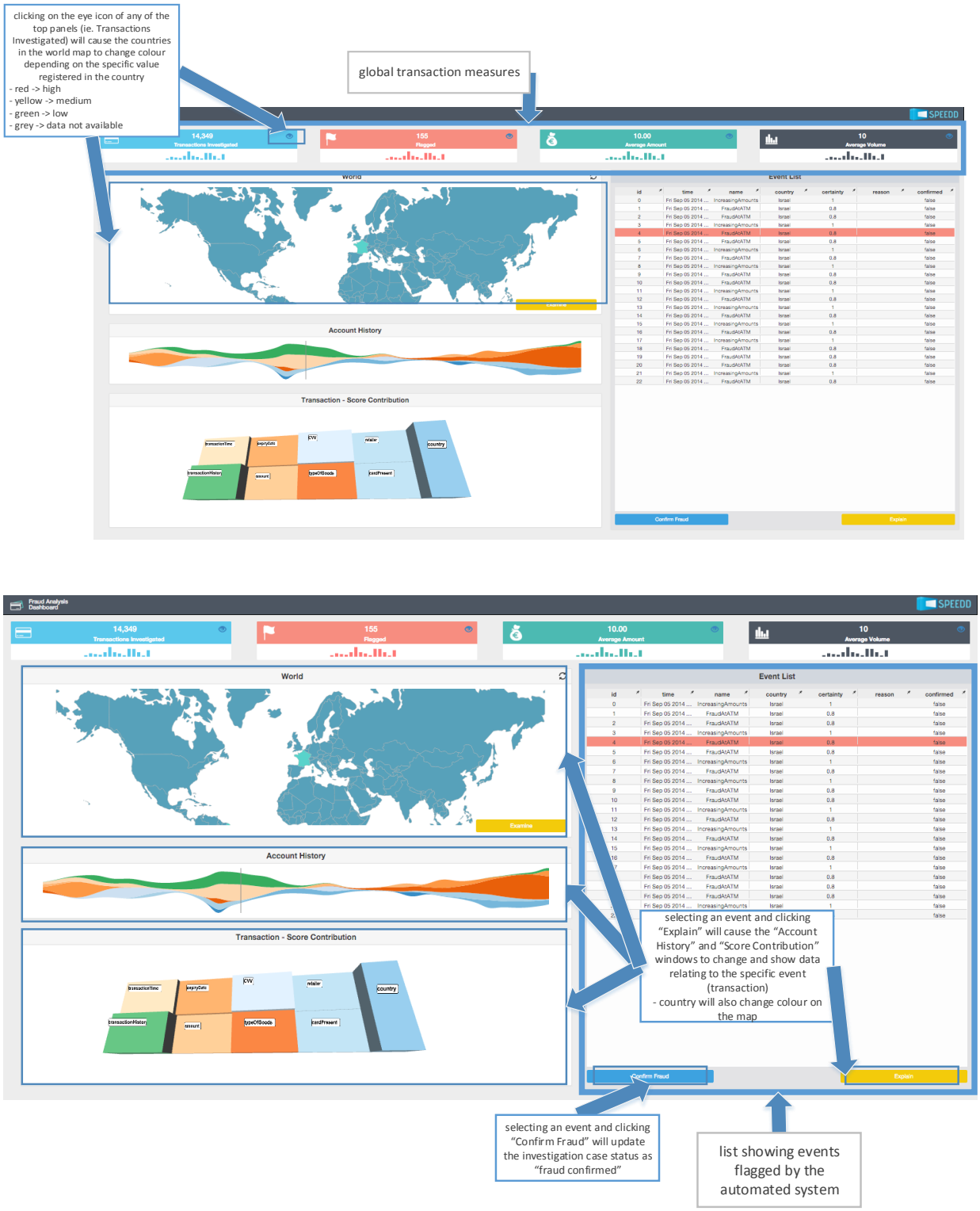


**Figure 2: Version III of User Interface design for SPEEDD Credit Card Fraud Use Case.**

## 4.2 Evaluating Version III User Interface

Interviews were conducted with 4 employees of Feedzai on 21st January 2016. The employees were made available by Feedzai in response to a request to provide staff who had experience and understanding of fraud analysis. All employees had knowledge of fraud analysis and one was specifically employed to analyse fraud patterns. While none of the participants were professional financial fraud analysts, it was felt that their knowledge of the domain provided sufficient experience to allow them to make informed evaluation of the prototype designs. Each user spent around 30 minutes interacting with the SPEEDD user interface.

As in D7.2 and D8.3, a usability evaluation of the UI design was performed using the Software Usability Scale questionnaire (Brookes, 1988). The questionnaire was translated into Portuguese (the English and Portuguese versions are in Appendix A). The SUS scale consists of 10 simple questions concerning the potential usefulness and benefit that users feel that the User Interface might provide them. Each statement is rated on a scale of 0 to 4. The scoring of responses then involves subtracting 1 from odd-numbered questions and subtracting scores of even-numbered questions from 5. This is because the questions alternate between positive and negative connotations. Scores are then summed and multiplied by 2.5, to give a final score out of 100. As a rule of thumb, scores in excess of 65 are deemed 'acceptable'. Figure 3 compares the evaluation of version III with the previous versions.

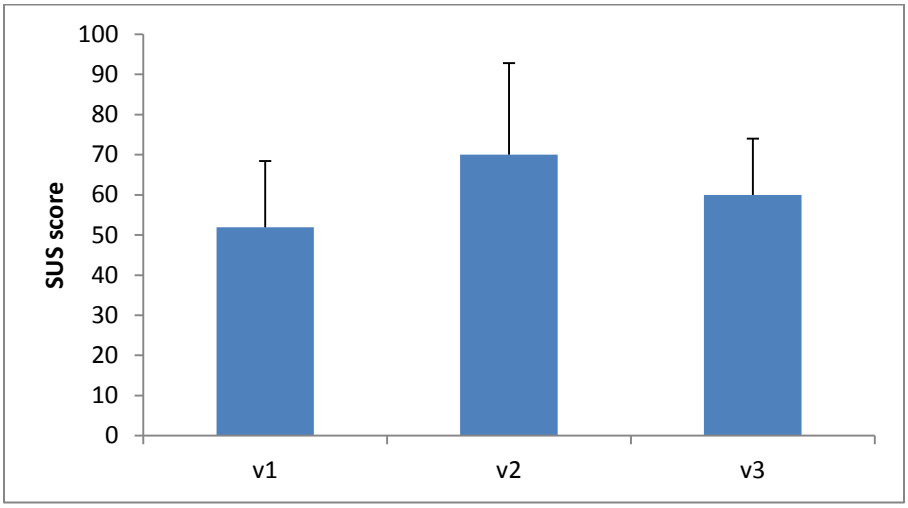


**Figure 3: SUS scores for the three versions of UI**

We note that the overall score on SUS is *lower* for version III than for version II. In each evaluation the variation in the scores reflected individual differences. This is illustrated by Table 1, where we have ranked the SUS scores from low to high for each version. It should be noted that the scores come from different respondents within and across versions. In all cases, it is apparent that some users provide much lower scores than others and version III, in particular, seemed to divide the responses into low (50 or less) or high (75 or more).

| Version I | Version II | Version III |
|---|---|---|
| 40 | 37.5 | 47.5 |
| 40 | 72.5 | 50 |
| 52.5 | 80 | 75 |
| 75 | 90 | 77.5 |

**Table 1: Rank order of SUS scores for the different versions of UI**

For version III, there were comments relating to the map, the timeline and the histogram. In general, users felt that the display was understandable. In response to version III, all participants described the workflow in two steps. First, look at the transaction queue (event list) and then, determine which needs to be processed next. From this, there was a requirement to allow the event list to be reordered in order to reflect different strategies, i.e., each strategy could reflect a different emphasis in the analysis and could involve ordering in terms of location, time, suspiciousness score etc. The histogram (bar chart) was confusing because its labels were not clear to the participants. Further the colour coding was not obvious to the participants, although they all recognised that the colour in the histogram matched those in the themeriver (timeline). Participants believed that the display would allow them to appreciate the contribution of each factor to the overall suspiciousness score although they also felt that many types of fraud could have a single factor which scored very high and other factors which would be less significant. For both the histogram and the themeriver, participants were not sure what the timescale was for the graphs, i.e., it was not clear whether the ordering was by date or sequence of transactions. The world map was not seen as vital to the decisions, although the fact that it changed shape was interesting and the use of colour coding was commented on positively. However, there was not agreement amongst participants as to what the colour or shape coding in the map should be used to indicate, i.e., it was not clear whether these properties would indicate overall trends, individual transactions or types of fraud. A common question related to the underlying models of fraud that the user interface was representing. This is interesting because, as a design concept, the user interface had no such models but the participants were interpreting aspects of the displays as if these reflected such models. The suggestion is that analysts take a two-pronged approach to their decision making which, on the one hand, considers the relevance (or value) of each item of information to their overall decision and, on the other hand, considers the overall decision in terms of plausible scenarios of fraud activity. While the former approach was considered in the experiments and models in D5.2, we have not considered in detail how the plausible scenario might develop or how this might be supported. In the next section, we consider the manner in which Feedzai personnel engage in fraud analysis using the Feedzai user interface.

# 5. Understanding the Fraud Analysis Process

## 5.1 Introduction

In D5.1, we presented an initial sketch of the strategy that we speculated could be followed by fraud analysts. This is shown in Figure 4. The aim was to presents a 'best-guess' description of how analysis might be undertaken.

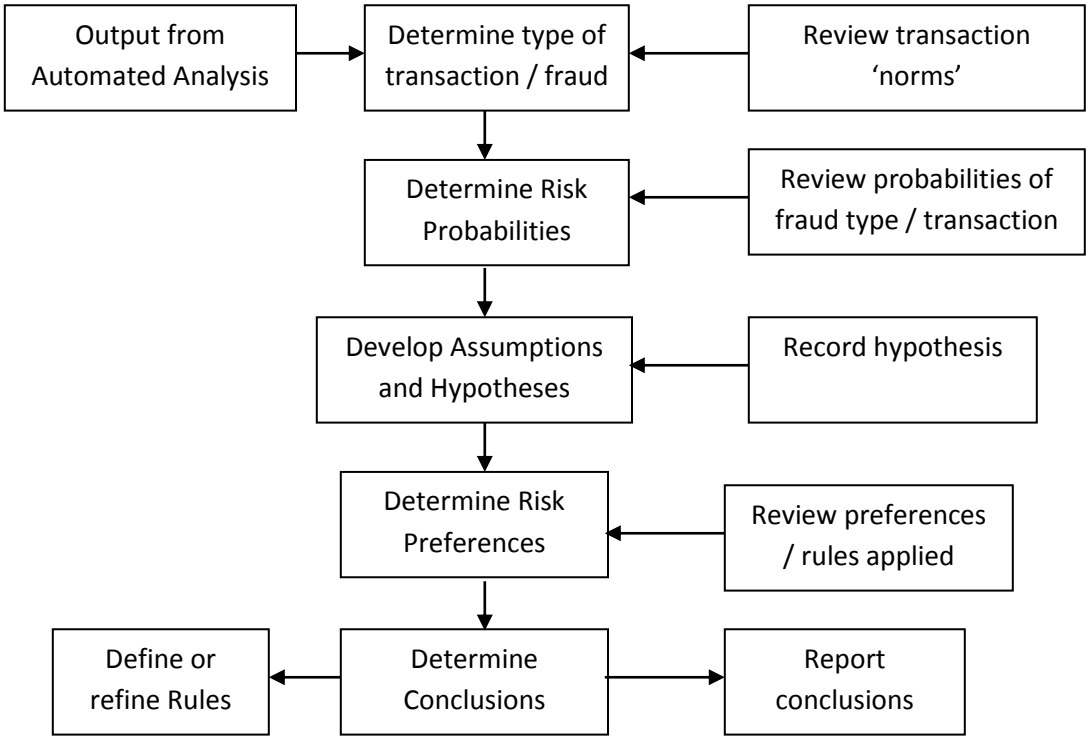


**Figure 4: Possible strategy for fraud analysis**

## 5.2 Using Fraud Analysis Software: case studies conducted at FeedZai

The Feedzai user interface is shown in Figures 5 and 6. On the Landing page (Figure 5), there is an overview which shows a summary (count of suspicious, safe and all transactions), daily statistics for a specific time frame and a data browser listing all transactions, including search filters, order options and option to reveal / hide reviewed transactions. The events list, on the left hand side of Figure 5, shows transactions to be investigated. Selecting one of these transactions takes the user to Figure 6. On this screen, there are customer contact details (i.e., email), the transaction amount, and the

suspiciousness (risk) score. From this page, the user can access pages with full customer information or full transaction details. The user will decide whether to block or allow the transaction.

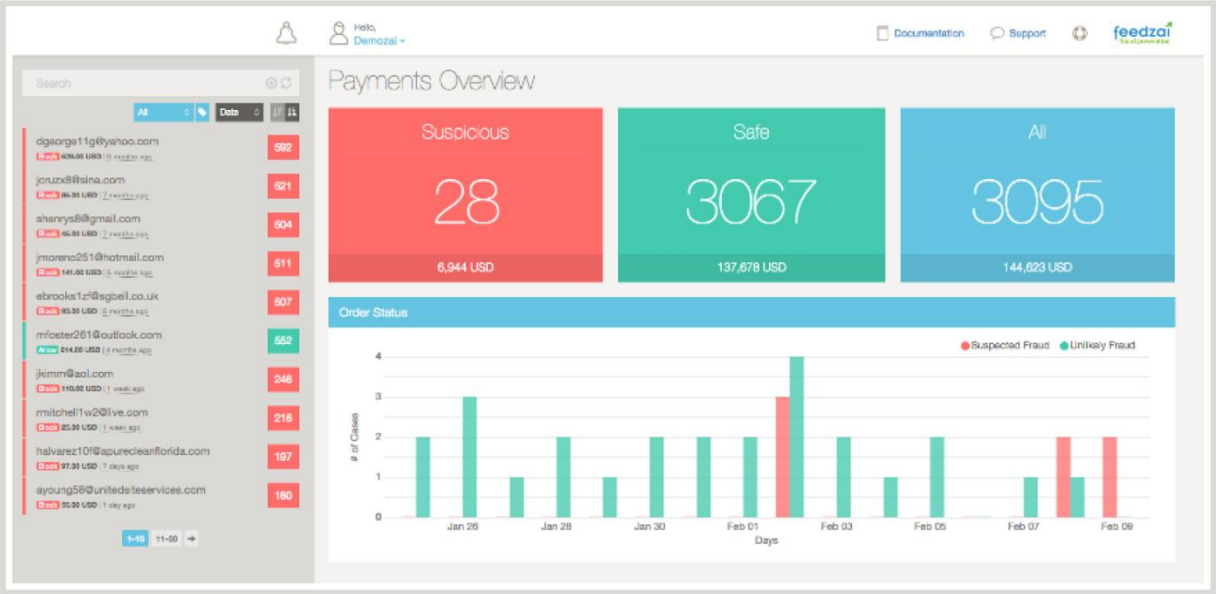


**Figure 5: Feedzai Landing Screen**

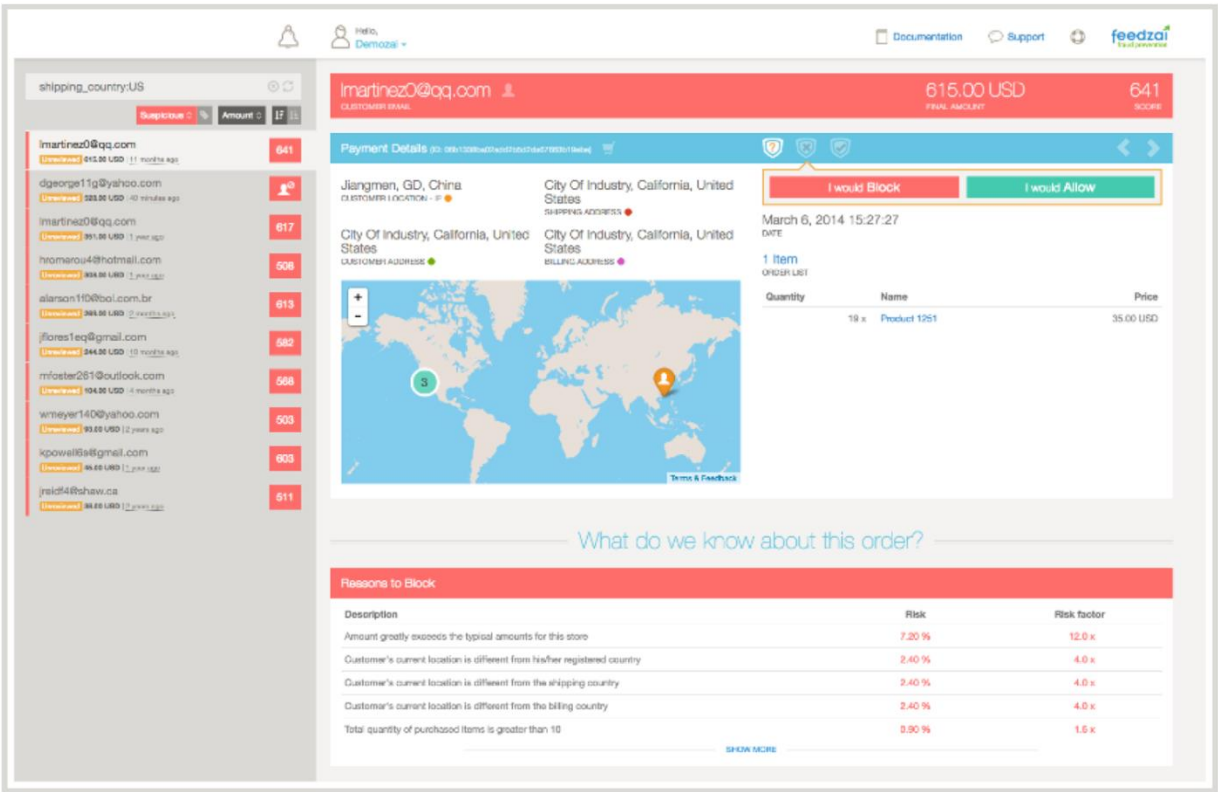


**Figure 6: Feedzai Transactions screen**

### 5.2.1 Eye-tracking with the Feedzai system

Four Feedzai analysts, who were unfamiliar with previous tests conducted in SPEEDD, volunteered to participate in a set of studies. One study explored the current version of the user interface, one study considered how analysts use the current Feedzai user interface and the third study considered how analysts react to automated support.

Given the user interface to the Feedzai system (Figures 5 and 6) we asked participants to work through a few cases in 20 minutes (the actual number range from 5 to 15). We asked participants to only use the information on the screen and to not make notes on paper.

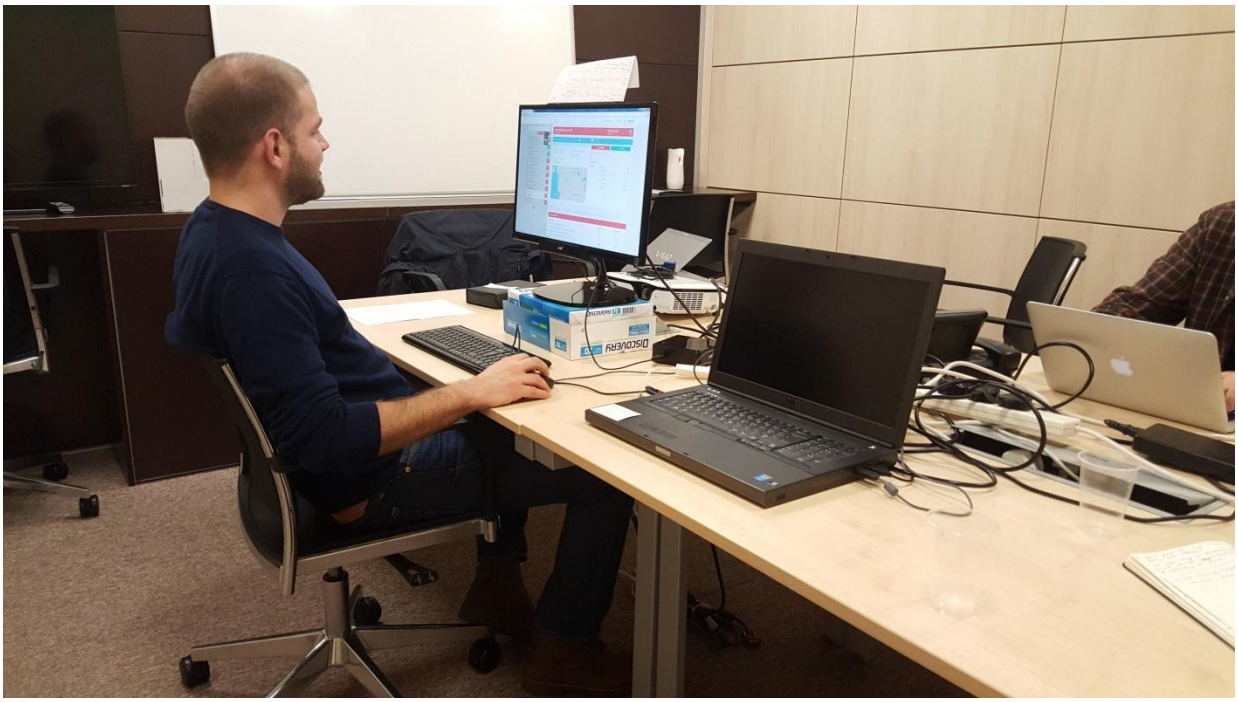

**Figure 7: Participant interacting with Feedzai system**

Each of the 4 participants had a different structure for using the analytics software. Emphasis varied between calling up customer information, transaction information, or just looking at the summary.

- Revisits to information that had already been attended to, e.g. calling up customer information several times between consulting other information
- Guidance through the analytics suite can be called up and introduces all components, so that the user gets a fast overview over the UI

## 5.3 Using Automated Support

In the second study we asked the Feedzai personnel to consider the possible utility of cueing decision relevant information. In D5.2, the computer model of human decision making that is being developed in SPEEDD assumes that salient information will be more likely to be attended to, than less relevant information. The experiment reported in D5.2 and the results of the model point to the suggestion that decision makers select relevant information in an efficient manner. We wondered whether it might be possible to encourage such selection by cueing those features which are relevant to defining a specific instance of fraud.

This raises several questions that WP5 is addressing, in terms of the Human Factors of automation and human decision making. For example, given that the automation will make suggestions to the human,

- Do people take computer suggestions on board?
- How do people interact with an automated support system?
- Do people look at the regions which are highlighted by the computer?
- Do people make decisions according to the computer suggestion?

The study (which we present here as a pilot for a more detailed experiment) employed the information sources from D5.2. This provides a simplified and abstracted version of fraud analysis which is sufficiently tractable to solve quickly while containing sufficient dimensions to have some uncertainty. We would suggest that the task is analogous to the decisions made by call centre operators rather than the other types of fraud analyst (see Section 3).

As illustrated by Figure 7, there are 9 information sources arranged in a matrix. The task is to determine whether fraud has occurred, on the basis of simple patterns across these sources. There were 4 types of fraud that could be identified on the basis of these information sources:

- Card usage close to expiry
- Transactions in far-away locations in a short time period
- Multiple small transactions
- Other fraud patterns

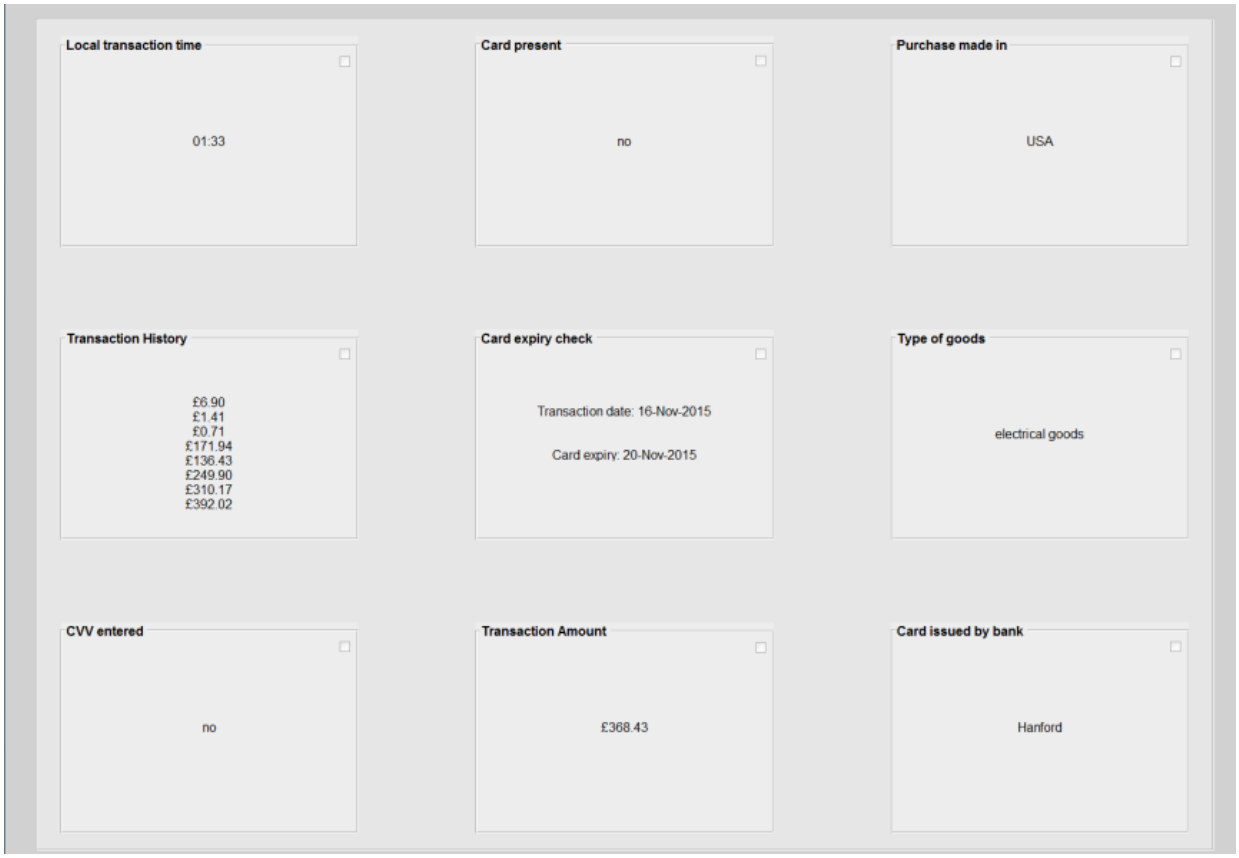


**Figure 8: User interface used in experiment**

The procedure involved each participant classifying 15 transactions twice: once without computer support, using their own experience and understanding of the patterns; and once with computer support.

The computer support provided cues to the user by colour coding three information sources:

- One region suggested as most important = **blue**
- Two regions suggested as also important = **grey**

For this task, the computer system was correct 100% of the time. This means that the most efficient strategy would be to look at the suggested regions and make the decision accordingly.

Participants sat facing the screen. A Tobii X2-60 eyetracker (mounted on the bottom of the screen) was used to track eye-movements. The definition of fraud types (in terms of information sources) was provided on a crib-sheet (on top of the screen).

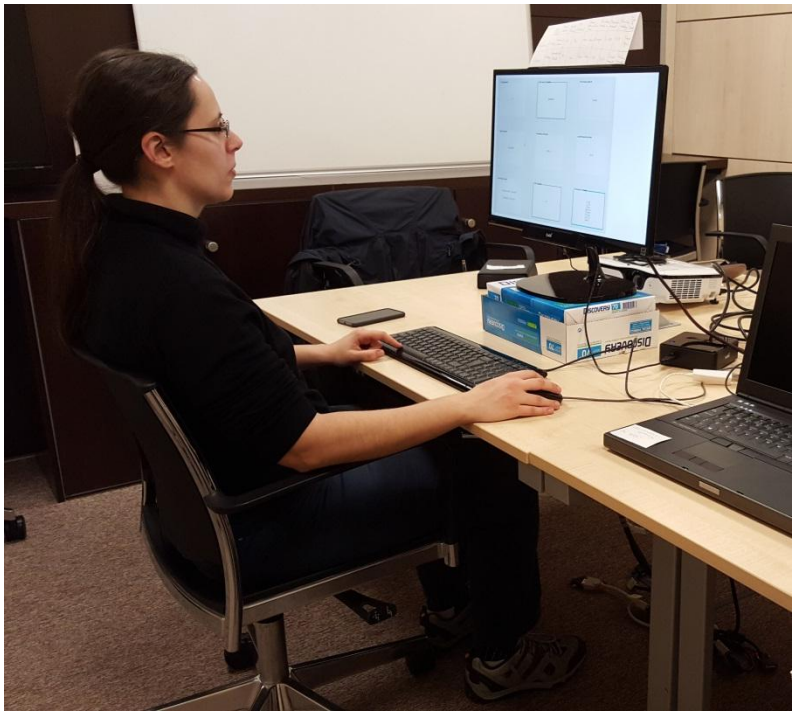

**Figure 9: Participant using eyetracking during experiment**

### 5.3.1 Information seeking in the first 3 seconds

Figures 9 and 10 show the pattern of eye-movements in the first 3 seconds for both conditions. It should be noted that the information sources moved between trials (so the screen layouts are not identical in these figures). It is suggested that, without the computer support, participants tend to focus on 2 or 3 regions before making their decision. This is illustrated by the constellations of dots being concentrated in specific regions in Figure 9. With the computer support, participants are more likely to focus on the 3 highlighted (blue and grey) regions (as shown in Figure 10). However, interestingly, participants will *also* attend to regions which are not highlighted. This suggests that the colour cueing is sufficient to attract their attention in the first 3 seconds, but that they may also seek other sources to check.

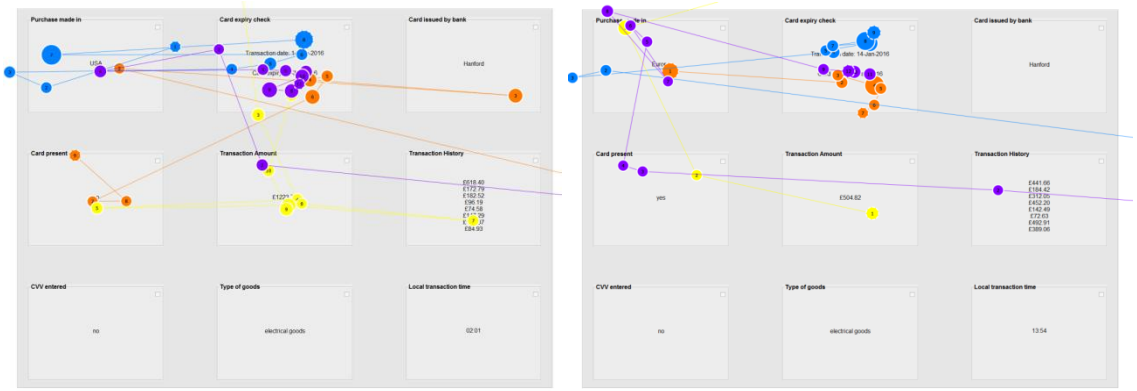


**Figure 10: Eye movements in first 3 seconds for two participants without computer support**

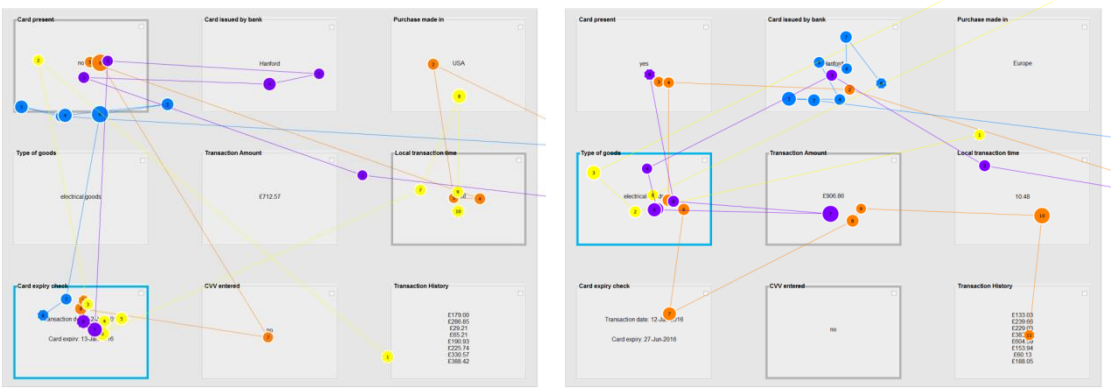


**Figure 11: Eye movements for two participants using computer support**

### 5.3.2 Analysis of Eye Tracking Data

Figures 12 and 13 compare the gaze duration for the different information sources, without and with computer support. In each graph, the relevant sources are highlighted (in blue and grey) to indicate which sources ought to be attended to by the participant. We had expected a difference between the two conditions, in that the computer support could be assumed to have fewer but longer gaze durartions on the highlighted sources. This is not the case. Indeed, the distribution of gaze duration across information sources is remarkably similar between the two conditions. This suggests that the cueing did not have an overt impact on strategy (for this participant, and this is the case for the other participants as well).

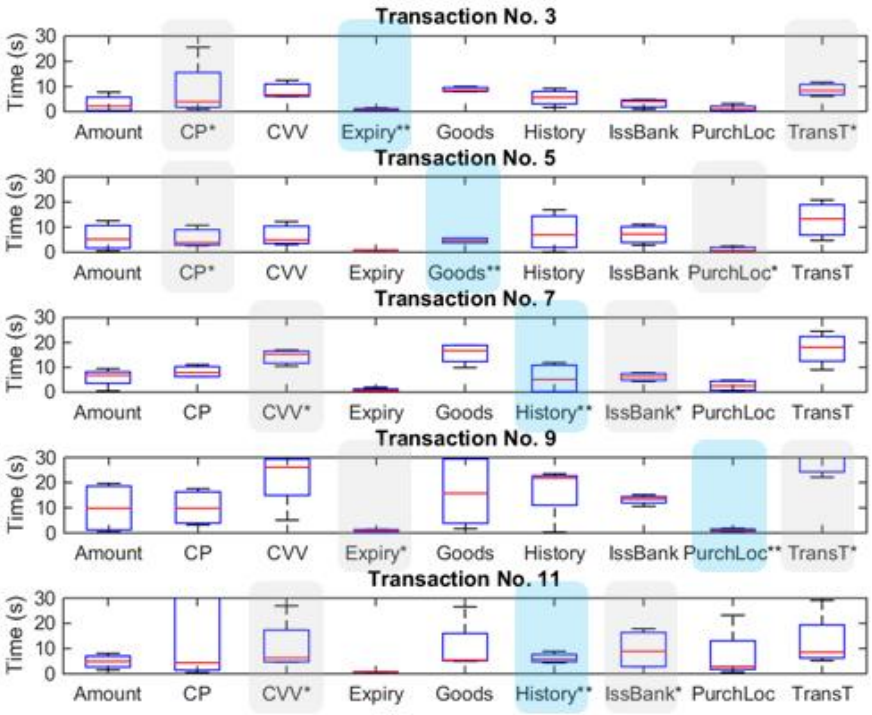


**Figure 12: Without computer support**

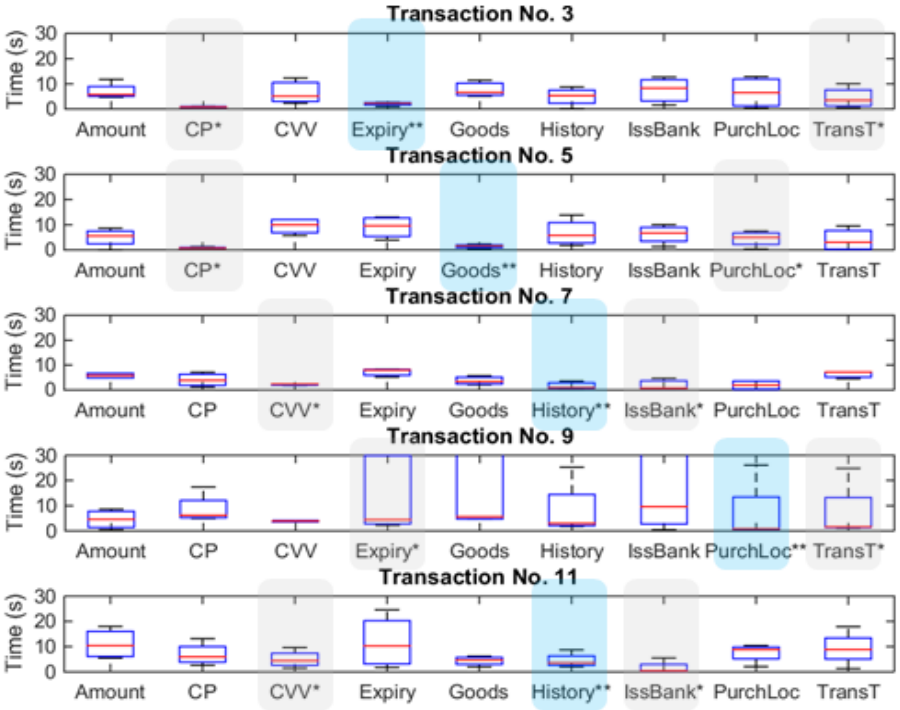


**Figure 13: with computer support**

Figure 13 shows the time from the start of the task to fixation on the first information source in a trial (averaged across participants and trials for each condition). It can be seen that the time to fixate on the first information source is faster and more consistent (i.e., lower interquartile range) in the computer support (rec) conditions when that source is highlighted in blue. When the source is highlighted in grey, the time to the first visit is, again, faster with computer support (although consistency is similar). When there is no highlighting, there is no difference between conditions. Thus, highlighting makes the information source conspicuous and this, in turn, is more likely to attract the user's visual attention. This is what we had intended to occur.

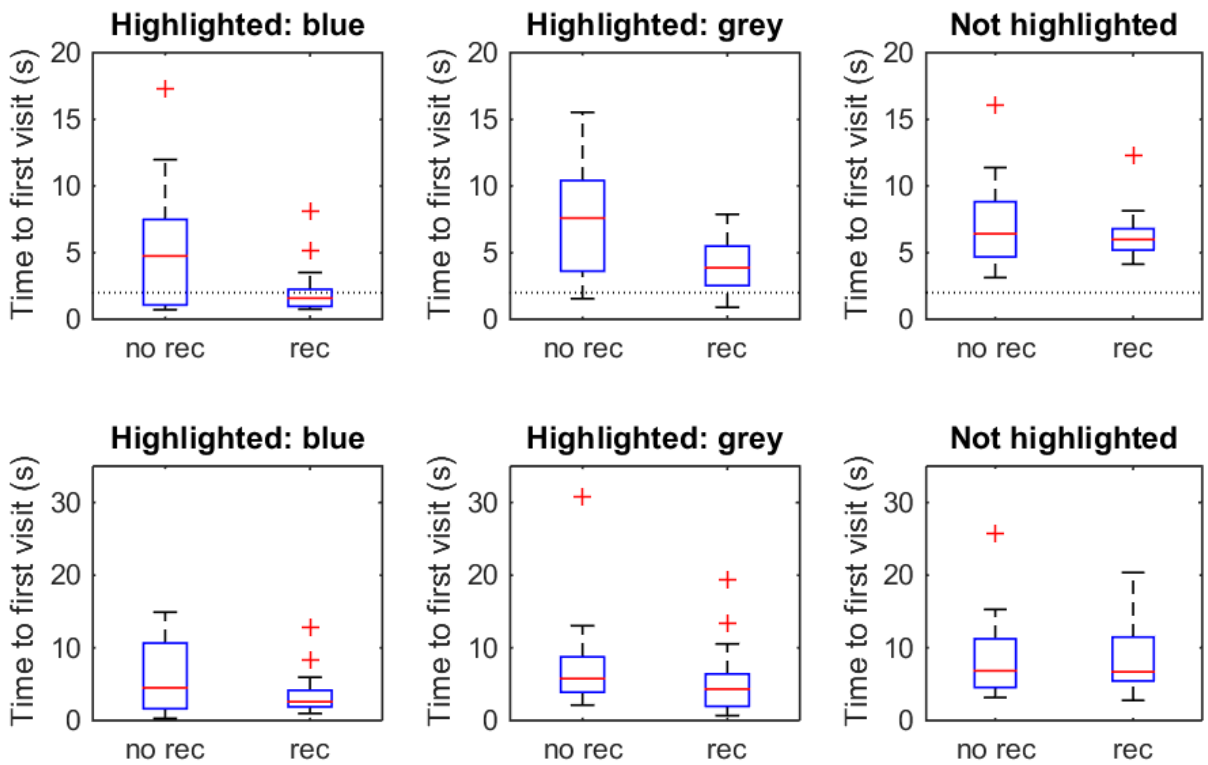


**Figure 14: Time to first visit (median time, top; interquartile range, bottom)**

### 5.3.3 Information Seeking across the whole trial

The analysis so far as concentrated on the initial response to the displayed information. We have shown that the participants respond more quickly to the highlighted information sources but they do not necessarily treat these are more important in their information search. Comparing performance across all trials (Figures 14 and 15), it is clear that participants tend to treat the information sources similarly across trials.

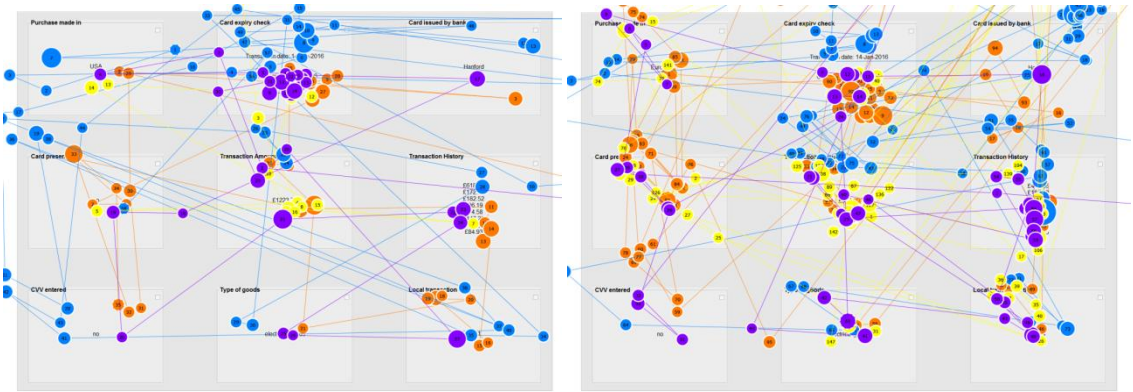


**Figure 15: Eye movements of two participants across whole trial, not using computer support**

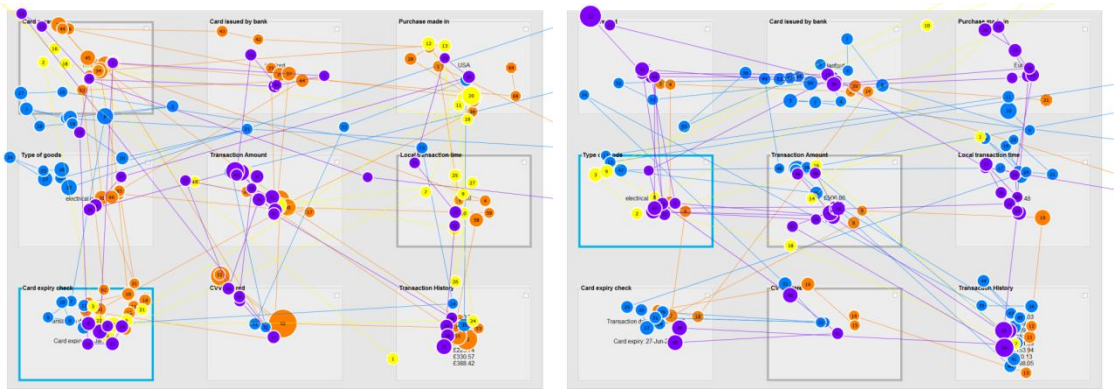


**Figure 16: Eye movements across the whole trial for two participants using computer support**

### 5.3.4 Conclusions and Observations

The notion of using computer support, presented using cued information sources, received a mixed response from the participants. On the positive side, they suggested that this could reduce the time taken to make decisions by highlighting key places to search, and that this could give an insight into the 'belief' of the automated system. Some of the participants felt that the approach could increase their own decision accuracy (although their estimates of how accurate the system was ranged from 10% to 60%). On the negative side, the system seemed to highlight information that the user felt was not relevant which meant that there was some concern over the reliability and trustworthiness of the system, and also some confusion over what rules the system was applying. Thus, despite the use of highlighting of specific regions drawing visual attention very fast, users often chose to ignore the attended information as it did not fit their own decision heuristics. For example, a popular strategy was to count the number of factors which agreed with the definition of fraud on the crib-sheet, and to assume a majority verdict for the decision, i.e., if more than 5 factors pointed to fraud then accept this. While this was the description offered by participants, the eye-tracking data suggests that they were looking for confirmation from fewer factors. Thus, the eye-tracking implies that participants were

tending to select around 3 factors before making their decision. This could be interpreted in terms of the Take-The-Best strategy (discussed in D5.2). An alternative strategy was to concentrate exclusively on specific sources, e.g., location, amount, expiration date, and use these to inform the decision. This could reflect a particular view of what consistutes fraud or which are the key factors. A problem with such an approach is that it could lead to missing examples of fraud which are not defined by these sources. In some instances, the approach was to create a story and use this to test the possibility of fraud, e.g., "Card present at home location to buy electrical goods in the middle of the night seems weird".

## 5.4 Laboratory Pilot Study

The study in Feedzai suggested a clear separation between attending to information and making a decision. This counters previous work reported in D5.2. We believe that an explanation of this difference arose from the manner in which the Feedzai analysts interpreted the 'rules' and the 'stories' that they constructed around these rules. It was apparent that there was a discrepancy between the rules that had been used to define the system's selections and those that were being applied by the analysts. Consequently, we conducted a further pilot study in which participants (N = 5) were required to strictly apply the rules that had been defined.

The workflow required participants to check the information sources against the rules (Figure 16, 1), then select a decision option (Figure 16, 2). Following this decision, the computer support presented its recommendation (Figure 16, 3) and the participant could change or confirm their decision (figure 16,4).


| | Card expiry check | Card issued by bank | Card present | CVV entered | Local transaction time | Purchase made in | Transaction amount | Transaction history | Type of goods |
|---|---|---|---|---|---|---|---|---|---|
| Normal | ≥ 5 days | NorthWest | Yes | Yes | 6:00 – 20:00 | USA | ≤ 500 | N/A | Electrical goods |
| Fraud | ≤ 4 days | Hanford | No | No | 20:00 – 6:00 | Europe | ≥ 510 | 3 small amounts | Travel agent |
| Usefulness | 60% | 55% | 60% | 70% | 60% | 55% | 65% | 85% | 55% |

**Figure 17: Workflow for laboratory pilot study**

Figure 17 shows the participants (P1..P5) attention to the 3 recommended information sources at their initial selection (blue), after the computer support had make its recommendations (in red) and in the next trial (in orange).

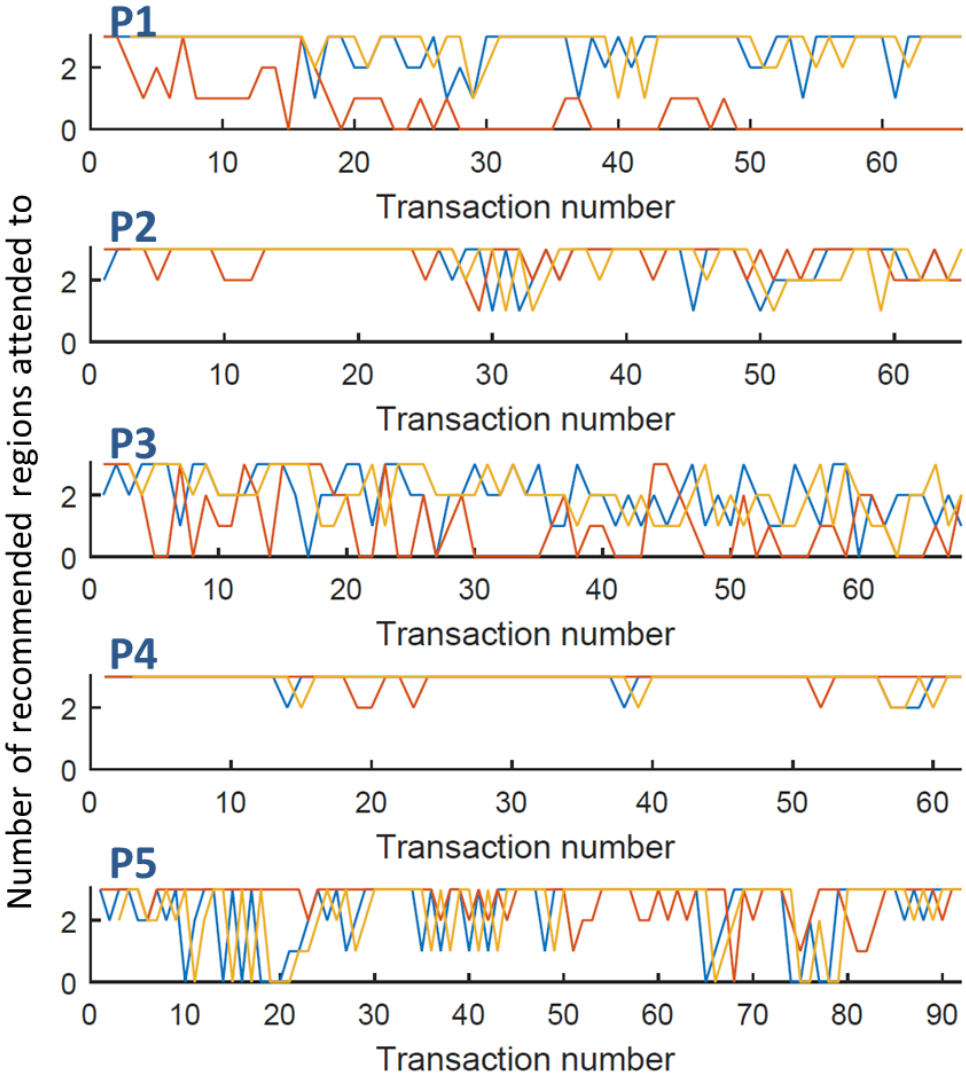


**Figure 18: Attention to recommended information sources**

As with the Feedzai trial, participants did not change the number of information sources that they checked, in the presence of the computer support. There seems to be large individual differences in the use of the computer support. However, participants reported that the computer support seemed to make the task easier. Overall, accuracy was over 70% for 4 participants (and around 50% for the fifth) with little difference between the two conditions.

# 6. Evaluation Using Patterns

Besides the visual evaluation of the prototype, SPEEDD also evaluates its system on terms of detection performance. While the visual evaluation of the prototype has the goal of helping the fraud analysts detecting more correctly the presented transactions, the evaluation over the detection performance has the goal of reducing the number of alerts that actually reach the analysts.

The definition of this work resulted from the collaboration between IBM and Feedzai to define a set of considered patterns. Once the initial group of patterns was set, IBM implemented them in PROTON, tested and handled the code to Feedzai, for testing on real data. The results were then passed back from Feedzai to IBM for further iterations.

This section starts by defining the conditions where the initial evaluation took place, then analyse the initial results and draw a set of conclusions for future work. A description of the implemented performance improvements is also described, including a description of what was done and how they influence the results obtained. A more in depth analysis of this theme can be found in deliverable D3.2.

## 6.1 Evaluation Datasets

The very first evaluation was done over a sample of the data, with 10.000 heavily anonymized transactions. This sample allowed IBM to create the first version of the rules and proceed to internal testing and debugging. Due to the heavy anonymization, no conclusive results could be drawn out of this data set.

The first real evaluation on the Credit Card Fraud Management use-case was accomplished using a subset of 1 million transaction, which spawned for a 1 month of data. This subset of data contained 122 fraudulent transactions and as it corresponds to real data, had to be evaluated inside Feedzai facilities. The evaluation was executed in a single node and no hardware performance was recorded, focusing solely on trying to achieve qualitative results over fraud analysis.

## 6.2 Initial Results

Given the defined patterns, none of the 122 fradulent transanctions was caught by the system in the first iterations of the evaluation process. Due to the need of running the patterns on Feedza premises, the evaluation, debugging and fixing cycle was considerable slower, which did not help getting better results.

Another impediment on the progress of the fine-tuning process is the fact that in the dataset, transactions are only tagged as fraudulent or genuine. This does not allow to know which was the pattern or behaviour behind the fraud tagging, which increases the hardness of the problem.

## 6.3 Performance evaluation

Latency is defined as the elapsed time between the detection time of the last input event required for a pattern matching and the corresponding detection time of the output event. For example: assuming that the (derived) event D is defined as a sequence of events (E1, E2, E3) then the latency is measured as the time period since an instance of E3 arrives into the system till the emission of the corresponding instance of D.

Given that besides precision and recall, one of the main constraints in the credit card fraud detection use case is the response speed, benchmarks regarding system latency were made, where performance issues were detected. The system was responding with an average latency of around 12 seconds, reaching maximum values of 140 seconds. That naturally demanded action to tackle this performance issue.

Figure 19 depicts the evaluation process. PROTON is fed recurring to input files, containing a list of events to be processed. A module called *Analyzer* records all the events – both input and derived ones. For every event, the analyzer registers the detection times. The output of the analyzer is a test log which is processed later by the *Stats* utility, which returns the registered latency.

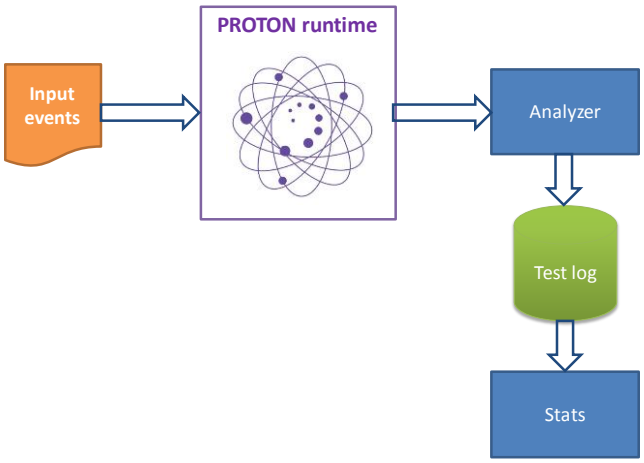


**Figure 19: PROTON performance testing conceptual overview**

The dataset consisted of 30K events injected at a non-uniform rate with peak of 100 events/sec, randomly distributed over the context of credit card PAN.

The performance testing was executed on a machine with the following configuration:

- CPU: Intel® Core ™ i7-4800MQ @ 2.70GHz -- Cores : Dual Core
- RAM: 16.0 GB
- OS: Windows 7 (64-bit)

## 6.4 Performance test results

Figure 20 shows the evolution of latency, in milliseconds. While the initial processing latency is low; it constantly increases over time until it reaches the values of **140,000 milliseconds** with an average latency of **11,186.4 milliseconds**.

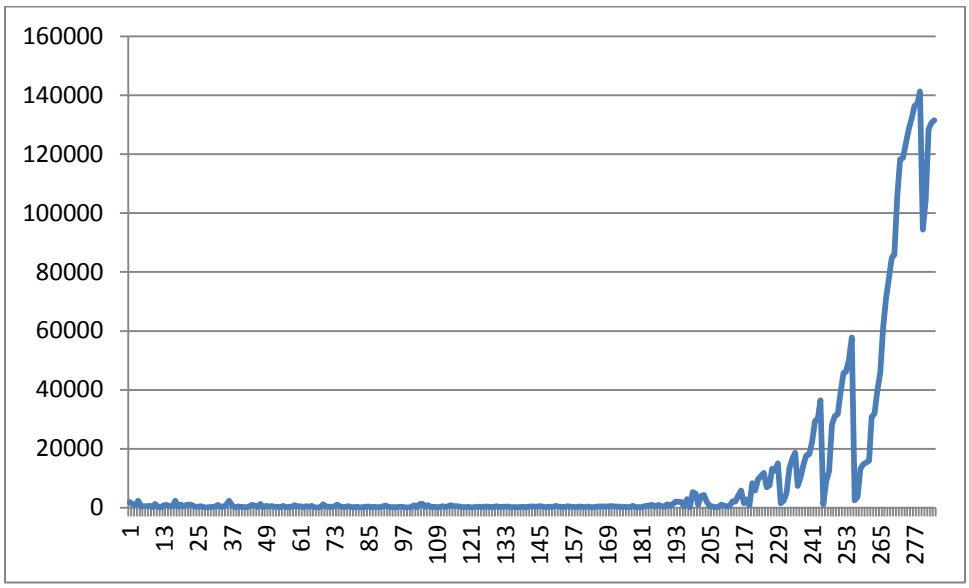


**Figure 20: End-to-end latency in milliseconds.**

## 6.5 Analysis of results

In the general case, PROTON implements its operators in an incremental way, meaning that for event aggregations, only intermediate calculations are kept. This means that the step of adding new events to the state can be achieved in constant time. This stands true for all the operators in the testing network and therefore, the high latency does not stem from the event processing agents gathered state, as these only perform incremental calculations.

Monitors tools were then used to look for bottlenecks at execution time, as all pointed that this was the root cause for the performance issues. Indeed, as shown by Figure 21, most of runtime threads were spending most of their time in the blocked state, represented by the red bars in the graph. This happened due to synchronization calls in the evaluations of context segmentation expressions.

A higher injection rate naturally leaded to a higher bloackage, as more threads are created for handling input events which then later, concurrently tried to gain access to the evaluator of the context segmentation expressions, turning this into a contended resource.


**Figure 21: Thread monitoring of the application during run**

In the credit card use case, the segmentation context is based on the credit card PAN (i.e. the first six digits of the credit card number). Given that this context was shared by all the elements in the network, while context service is busy evaluating an expression for a single input event, no more events can be processed. This caused major blocking of all threads partitioning the input events into context subgroups. We have seen that with time, all the threads were blocked in this particular lock 96% of time. The solution to this issue was to define each evaluator as thread-local. Given that now each thread manages its own context, there is no longer the need for synchronizing the resource.

However, the synchronized usage of this resource was not the only problem contributing to the increase of the latency. As soon as the correction was applied to the synchronized context evaluator, two additional minor issues were detected, namely logging overhead and overhead resulting from writing to a console as explained next.

The problem with the logging overhead was caused by creation of additional strings even when the debugging level was not set. This problem was corrected by checking before if the debugging level is active or not before creating the logging messages. As for the overhead writing to the standard output, this method also results in additional blocking of threads trying to write at the same time. It was solved by using only the logging framework.

## 6.6 Performance improvements

After applying the described corrections, latency has improved significantly (see Figure 22) – moving to an average of **139.8 milliseconds** (with an average matching set size of 3.2 events) and max peak latency of **2,000 milliseconds** at peak injection time (around 100 events per sec). This shows **80 times**

improvement in the average latency and **70 times** improvement in peak latency comparing to our previous version.

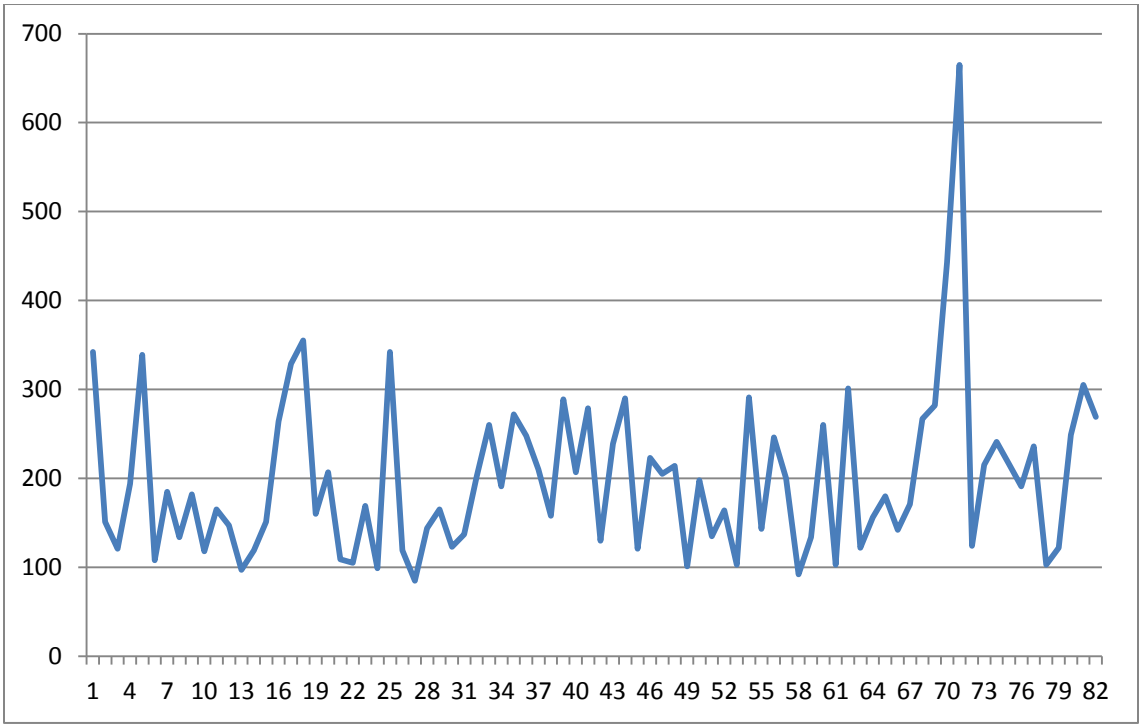


**Figure 22: End-to-end latency in milliseconds after changes in PROTON's code base**

## 6.7 Future Work

Given these results, SPEEDD has naturally several tasks to achieve during the next evaluation in order to get close to the desired results.

First, the problem regarding the lack of information behind the reason why the transactions were marked as fraud must be solved. Because there is no other dataset which could be matched to get those reasons, some sort of manual process must be found to complete the dataset. The process will likely result in trying to infer the patterns given other fields present in the dataset.

Second, the evaluation must occur within a cluster and not in a single machine. Moving to a cluster does not only fulfil one of the requirements of SPEEDD, regarding scalability, but will allow to accomplish the third need for the Credit Card Fraud Management, which is to collect detection performance results in the entire dataset and not just in subsets of it.

These steps should be fully addressed in order to achieve the goals proposed for the project for the third and last prototype evaluation.

Finally, some performance issues were detected, the major ones related to synchronization of resources and the minor related to debugging and logging messages. All of them were addressed and solved, resulting in the improvements visible on Table 2.

| | Before (millisecs) | After SPEEDD developments (millisecs) |
|---|---|---|
| Maximum latency | 140,000 | 2000 |
| Average latency | 11,186 | 140 |

**Table 2: Latency performance before and after corrective actions**

# 7. Conclusions

Credit card fraud management can be evaluated in two perspectives. In the perspective of the analyst, who has to evaluate a set of alerts, and in the perspective of the machine, which has to generate those alerts. It is always a balance between the two. Machines are not still perfect to the point of making fully autonomous fraud detection, while analysts are not quick enough to handle the growing number of transactions and alerts processed each day. In this document, the SPEEDD prototype was reviewed once more in the perspectives of visual evaluation and detection performance evaluation.

For the first, the evolution of the SPEEDD graphical user interface was shown and how it influenced the analysis from fraud experts. It showed how the disposition of the information items could help an analysts addressing the decision process better, how analysts reacted to the suggestions constructed by the computer and which visual elements were given the most importance.

As for the decision evaluation, it was described as the process used to access the real data and perform the analysis, the description of the datasets used and the results extracted from these analysis. Given those same results, some conclusions were drawn regarding the numbers (none of the 122 fraudulent transactions out of 1 million transactions) and how they can be improved.

Finally, the document also presented an overview of what is ahead, namely the direction where the user interface will grow and the defined plan to tackle the problems detected during the decision performance evaluation, in order to accomplish the goals stated by the consortium for this use-case.

**References**

Brooke, J., 1996, SUS: a quick and dirty usability scale. In: P.W. Jordan, B. Weerdmeester, B.A. Thomas and I.L. McLelland (eds) *Usability Evaluation in Industry*, London: Taylor and Francis, 189–194

# 8. Appendix 1 – SUS questionnaires in English and Portuguese

***System Usability Scale***



| | Strongly disagree | | | | Strongly agree |
|---|---|---|---|---|---|
| 1. I think that I would like to use this system frequently | 1 | 2 | 3 | 4 | 5 |
| 2. I found the system unnecessarily complex | 1 | 2 | 3 | 4 | 5 |
| 3. I thought the system was easy to use | 1 | 2 | 3 | 4 | 5 |
| 4. I think that I would need the support of a technical person to be able to use this system | 1 | 2 | 3 | 4 | 5 |
| 5. I found the various functions in this system were well integrated | 1 | 2 | 3 | 4 | 5 |
| 6. I thought there was too much inconsistency in this system | 1 | 2 | 3 | 4 | 5 |
| 7. I would imagine that most people would learn to use this system very quickly | 1 | 2 | 3 | 4 | 5 |
| 8. I found the system very cumbersome to use | 1 | 2 | 3 | 4 | 5 |
| 9. I felt very confident using the system | 1 | 2 | 3 | 4 | 5 |
| 10. I needed to learn a lot of things before I could get going with this system | 1 | 2 | 3 | 4 | 5 |

***Escala de usabilidade do sistema***

| | Discordo muito | | | | Concordo muito |
|---|---|---|---|---|---|
| 1. Acho que usaria este sistema frequentemente | 1 | 2 | 3 | 4 | 5 |
| 2. Achei o sistema demasiado complexo | 1 | 2 | 3 | 4 | 5 |
| 3. Achei o sistema facil de usar | 1 | 2 | 3 | 4 | 5 |
| 4. Achei que precisaria de assistência de um técnico para | 1 | 2 | 3 | 4 | 5 |
| 5. Achei as diversas funções do sistema bem integradas | 1 | 2 | 3 | 4 | 5 |
| 6. Achei que há demasiadas inconsistências no sistema | 1 | 2 | 3 | 4 | 5 |
| 7. A maioria das pessoas irá apreder a usar este sistema | 1 | 2 | 3 | 4 | 5 |
| 8. Achei o sistema muito complicado de usar | 1 | 2 | 3 | 4 | 5 |
| 9. Senti-me confiante a usar o sistema | 1 | 2 | 3 | 4 | 5 |
| 10. Preciso de aprender muitas coisas antes de conseguir usar este | 1 | 2 | 3 | 4 | 5 |