Big Data Risk Assessment the 21st Century approach to safety science

Dr. Coen van Gulijk, Dr. Miguel Figueres-Esteban, Peter Hughes
University of Huddersfield

SUMMARY

Safety Science has been developed over time with notable models in the early 20th Century such as Heinrich's iceberg model and the Swiss cheese model. Common techniques such fault tree and event tree analyses, HAZOP analysis and bow-ties construction are widely used within industry. These techniques are based on the concept that failures of a system can be caused by deviations or individual faults within a system, combinations of latent failures, or even where each part of a complex system is operating within normal bounds but a combined effect creates a hazardous situation.

In this era of Big Data, systems are becoming increasingly complex, producing such a large quantity of data related to safety that cannot be meaningfully analysed by humans to make decisions or uncover complex trends that may indicate the presence of hazards. More subtle and automated techniques for mining these data are required to provide a better understanding of our systems and the environment within which they operate, and insights to hazards that may not otherwise be identified. Big Data Risk Analysis (BDRA) is a suite of techniques being researched to identify the use of non-traditional techniques from big data sources to predict safety risk.

This paper describes early trials of BDRA that have been conducted on railway signal information and text-based reports of railway safety near misses and the ongoing research that is looking at combining various data sources to uncover obscured trends that cannot be identified by considering each source individually. The paper also discusses how visual analytics may be a key tool in analysing Big Data to support knowledge elicitation and decision-making, as well as providing information in a form that can be readily interpreted by a variety of audiences.

INTRODUCTION

Analysis of Big Data requires a suite of methods that process the large amounts that are present in modern system, such as the internet. Big Data is generally defined as very large sources of data (volume), that arrive at the input very quickly (velocity), from many sources (variety), with many different data types (value), and possibly with unknown accuracy (veracity). These qualities form the *five Vs* of Big Data; Figure 1 illustrates this concept within the GB Railways.

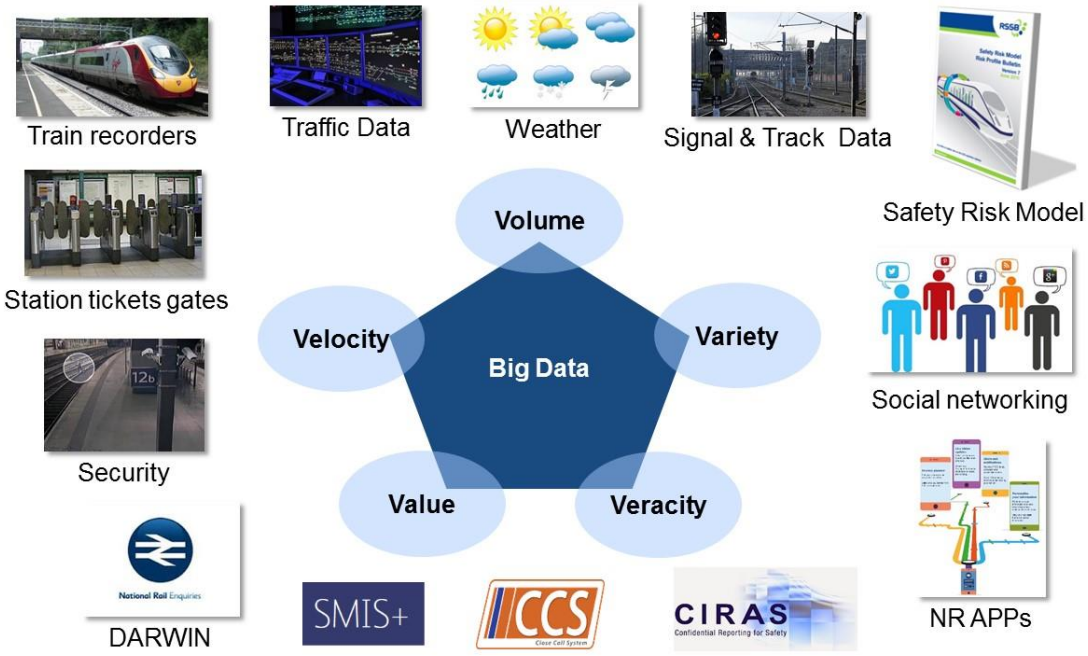

Figure 1: The five Vs of big data

BDRA is a step-change risk management because it provides the toolbox to perform statistical analysis of *all* relevant risk data rather than estimating probabilities from samples or limited datasets, and therefore reduces the need for risk-based models that rely on assumptions or simplifications to produce risk estimates.

BIG DATA RISK ANALYSIS

The BDRA program at the University of Huddersfield aims to investigate whether and how rail safety and risk management could benefit from the new software tools that computer scientists develop today. In practice BDRA involves cloud-computing software tools that combine structured and unstructured data sources to support management decisions and is comprised of systems that:

- extract data from mixed data sources;
- process them quickly to infer and present relevant safety management information;
- combine applications to collectively provide sensible interpretation; and
- use online interfaces to connect the right people at the right time.

The outcome of this process is intended to provide better decision support for safety and risk management than would be possible by considering the data sources individually or only in small combinations. Our research to date has identified the following basic enablers of a BDRA system:

- data processing;
- data structure within an ontology;
- visualisation of raw data and derived results; and
- analytics.

These enablers have to be integrated to produce the emergent behaviour of BDRA, as illustrated in Figure 2. A suggested by its name: *data* is the basis of BDRA: modern technological systems in the GB Railways produce massive amounts of data. For instance, a useful data source is the data produced by supervisory control and data acquisition (SCADA) systems, which consist of coded signals that use internet communication channels to remotely control equipment. These signals typically include error messages and information about the state if a piece of equipment. For safety and risk purposes, dedicated databases are maintained where error messages are stored for further analysis. In addition, the Rail Safety and Standards Board (RSSB) maintains a database of millions of reportable safety incidents from the railway. In combination, these data sources can provide a powerful insight into how the operation and failure of equipment contribute to failures and accidents on the railway.

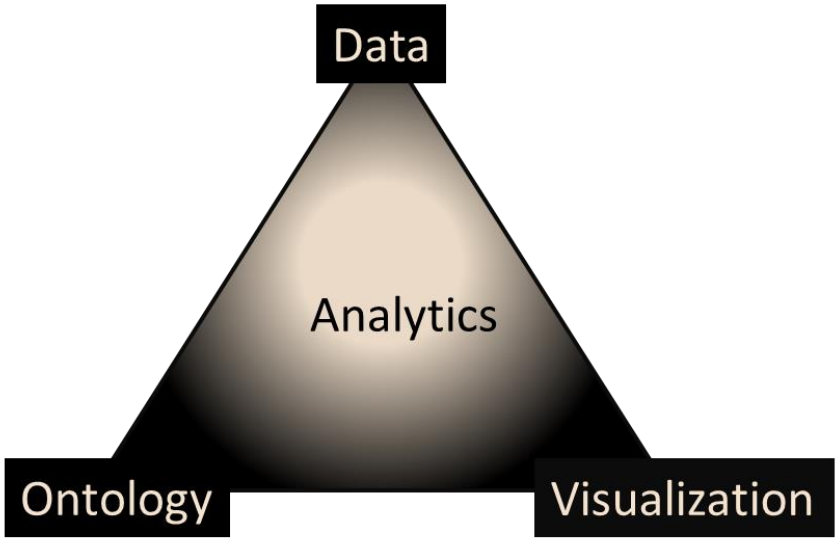

Figure 2: Basic enablers for BDRA

Ontology is the systematic classification of domain knowledge that supports the use of different databases in a meaningful way: it can be compared to a search engine which holds the right search keys to produce results that are relevant to the human operator. The search keys are based on a repository of concepts and words that represent the knowledge structure of a specific domain. In this case the domain is safety and risk for the GB railways, the concepts are the ways in which the components within the domain combine and interact to create the emergent behaviour of the overall system.

Visualisation is a powerful, and arguably the only useful, tool for understanding large quantities of data. Hundreds of different visualisation techniques are available to provide output of data (e.g. geospatial risks), however modern visual analytics tools also provide the ability to interact with the data to modify how the data is structured, how queries are performed, and how different users of a system can learn from the experiences of others.

Analytics is the software and data processing engine that forms the backbone of any computer-based risk analysis tool. For BDRA the computer system is a cloud-based computer that can processes and store large amounts of data. Software for analysis on distributed systems is different from software packages that run on a single computer. For BDRA several different software tools have to run in parallel and the results of these tools have to be combined into a higher layer of the software hierarchy. At the top-most layer, the software can enable access of the data via the internet.

COMPONENTS

This section describes how each of the enablers depicted in figure 2 are implemented for practical implementation of BDRA. Respectively they are data, ontology, visualization and analytics. This paper provides only an overview of the key components, fully details are available in our previous work, for example Stow et al. 2015; Hughes & Figueres-Esteban 2015; Hughes et al. 2015, Figueres-Esteban 2015 et al. 2015, Van Gulijk et al. 2015.

Data (Close Call)

The first enabler is data. This section treats the procedures and some analysis of text-based data that is relevant for BDR: the Close Call database. A close call is a hazardous situation where the event sequence could lead to an accident if it had not been interrupted by a planned intervention or by random event (Gnoni et al. 2013). Network Rail workers and specific sub-contractors within the GB Railway industry are asked to report such events in the Close Call database. Close Call reports are freeform text reports where users can describe a situation that, in their view, could have led to an accident. Providing a free text format for data entry allows the reporter the ability to describe hazards in a rich way that would not be possible if data entry were constrained, for example by selecting hazard types from a pre-defined list. The Close Call Database contains approximately 200,000 entries that have been collected over a period of two years. Due to the large number of records, it is impractical to manually review the records and therefore computer-based techniques have been developed to extract safety relevant information from them.

Since the key information relevant to safety management is found in the free text, Natural Language Processing (NLP) techniques are is used. NLP has been an emerging area of study over the past two decades in the domains of road safety and medicine (Allen 1994, Wu & Heydecker 1998, Dale et al. 2000). One of the key problems in NLP is the inherent ambiguity in written language, including the use of jargon, abbreviations, misspelling and lack of punctuation. Processing of close call data by extracting information from free text involves five processes (Hughes & Figueres-Esteban, 2015):

- text cleansing, tokenizing, and tagging;
- ontology parsing and coding;
- clustering (creation of groups of records that are semantically similar);
- text analysis; and
- information extraction.

The exact procedure is described in Hughes et al. 2015. This paper highlights two results of the information extraction process.

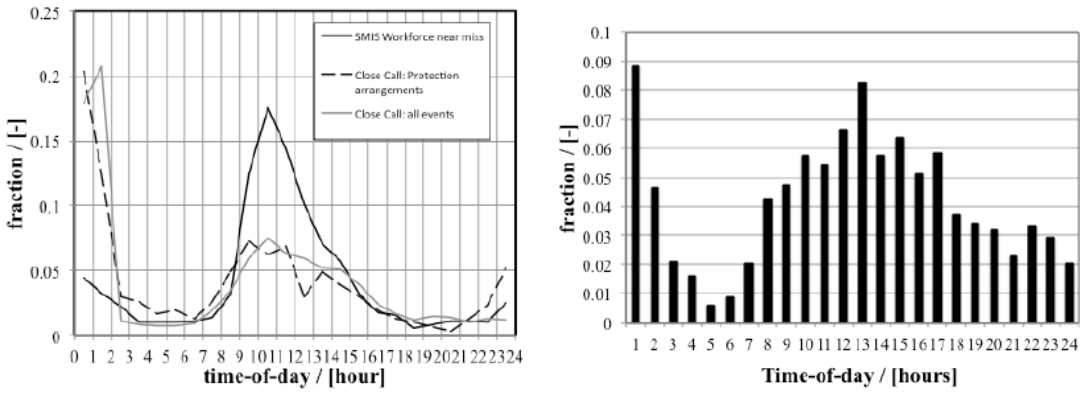

Figure 3. Frequencies of Workforce incidents in SMIS and Close Call.

The first information extraction process was the identification of incidents with track workers. The database of railway report incidents database shows that incidents with track workers take place more frequently between the hours of 11:00 and 15:00. An analysis was performed to investigate whether the same pattern is present in the close call database. An automated search query was programmed to retrieve the protection / possession arrangements events in the close call database as function of time-of-day. The results were compared with track worker near miss events in the incident database as function of time-of- day and with all events in the close call database as a function of time-of-day.

The relative distributions of these events by time of day are shown in Figure 3, which shows that the incident database and close call reports follow similar trends during the day. However further examination shows that the times at which reports are made for all close calls are similar to the times reports are made for protection arrangements; suggesting that there may be a reporting bias that interferes with the actually hazard report. The high fraction of close call events between 00:00 and 01:00 is likely to be due to a default of the reporting system that sets the time-stamp to 00:00 when the time of the incident is not entered by the person making the entry.

A similar problem was investigated in relation to trespassing to address the question: do trespasses take place at certain times of the day or do they take place with equal probability throughout a 24 hour period? Figure 3 suggests that trespass does not occur with equal frequency: the trend seems that they occur more frequently during working hours. What causes this trend is as yet unexplained but similar to the possession entries, reporting bias may play a role.

Ontology

The second enabler is ontology, which captures interrelating concepts in risk systems. Text parsing, tokenizing and tagging for Close Calls are based on ontologies (explained in the prior paragraph). Dahlgren (1995) suggests that concept-based ontologies are well suited for computer-based systems that attempt to provide a "world view". In this context a world view is a naive description of *what there is in the world* and how these components should be classified. Naive means that conceptualisation and classification take place on relatively shallow or common-sense assumptions in a dominant belief system. Guarino (1997) illustrates naive ontology building based prior work by Genesereth & Nilsson (1987). Consider five blocks (numbered a, b, c, d and e) in two separate piles on top of a table (Figure 4). A possible conceptualization can be given by:

<{a, b, c, d, e}, {on, above, clear, table}>

Where {a, b, c, d, e} is the universe under consideration: the blocks we are interested in. The set {on, above, clear, table} describes the relevant relations where *table* entails the concept of an object holding the blocks, *clear* entails the concept of not touching, and *on* and *above* are relations between objects and concepts. This naive conceptualization enables all possible combinations of blocks in two different stacks on the table without the need to create new rules. So different instances (e.g. instance a, b and c, d and instance a, c, e and d, b) are both allowed with the same conceptualization which reduces the complexity of the description of the world view. This reduction of complexity is of great value in the computer science domain and is also sufficient for most analyses since humans normally use a shallow layer of knowledge to describe meaning and intentions: Dahlgren (1995) states that nearly 80% of common-sense reasoning is based on the naive approach. Papers by Smith & Welty (2001), Noy & McGuiness (2001) and the RDF-documentation of the

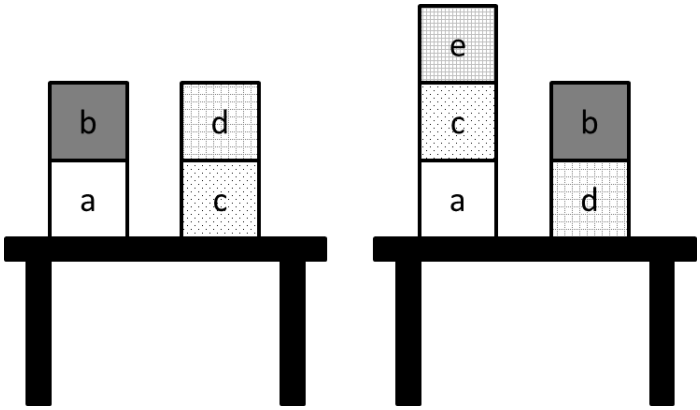

Figure 4: One naive conceptualization, 2 real world instances. Incredibly powerful in computer science.

World Wide Web consortium (W3C) demonstrate that this is the standard approach in the computer sciences today.

Visualisation

Visualization is the third enabler. In order to test value of visualisation for BDRA a visual-analysis was performed of Close Call text. We undertook this analysis in accordance with the data extraction methodology proposed by Paranyushkin (2011). Since the objective was to assess whether groups of similar hazards could be identified through visual analysis, a pre-constructed dataset was used. A sample of 150 records was constructed from selecting the 50 data records from the Close Call database classified as *trespass*, *slip / trip hazards on site* and *level crossing*. These records were cleansed using the natural language toolkit in Python (Bird et al. 2009) and the tagging and tokenization processes described in Hughes et al. (2015) were applied to create two types of text for visualising. The entire process of cleansing, tagging and tokenization is illustrated in the Table 1.

The visual analysis was performed by constructing the two-word-gap and five-word-gap networks and representing the networks with the Gephi software (Paranyushkin 2011). The networks obtained from the text were composed of nodes related to tags (e.g. geo_place, elr_code or distance_tag), tokens (e.g. level_crossing_, road_vehicle_, access_ or network_rail_) and words (e.g. location, trespasser or pedestrian). In order to gather knowledge from the networks the size of the nodes were scaled to represent the connections between individual words and tokens. The Louvain method for community detection with enough resolution was applied to represent large clusters from the networks (Blondel et al. 2008). The Louvain community detection algorithm detected four clusters from the five-word-tokenised text network with a resolution of 1.5. The result gives a modularity of 0.6 (Paranyushkin 2011). Figure 5 represents one of the clusters detected.

The cluster in Figure 5 shows high degree nodes with a high betweeness (cross_, geo_place, distance_tag and location). In addition we can also find a great quantity of high and medium degree nodes related to level crossings (e.g. elr_code, level_crossing, road, box_signal_ or signaller_). Moreover, the clusters present important differences regarding nodes related to people and type of terms used, for example nodes related to technical staff (*network_rail_, operative, member_of_staff or signaler)* and terms such as *signal box, cctv_, elr_code, cess, not working, approach_, clear_, main_, line_, dn_, up_ or downside*. The visual analysis provided insights into the data that were obscured when only the textual data was shown, these insights allowed for further refinement of the tokenisation process (removing stopwords and stemming plurals or verbs).

Original record
Emailed report from LOM   Date: 08/09/13 Time: 1900   ELR: LEN3 59m 14ch   Issue - Trespasser on the line in the Hartburn Junction area. Trains cautioned, reported all clear by MOM @ 1930 Action - Fencing to be checked 09/09/13   DU: <city></city><place></place> Newcastle   <!-- RICH TEXT -->
Cleaned record
Emailed report from LOM Date: 08/09/13 Time: 1900 ELR: LEN3 59m 14ch Issue - Trespasser on the line in the Hartburn Junction area. Trains cautioned, reported all clear by MOM @ 1930 Action - Fencing to be checked 09/09/13 DU: Newcastle
Cleaned and tagged record in lowercase
emailed report from local operation manager date _date_ time _time_ elr_code distance_tag issue trespasser on the railway line in the geo_place junction area trains cautioned reported all clear by mobile operations manager _time_ action fencing to be checked _date_ geo_place
Cleaned, tagged and tokenized record without stopwords and in lowercase
email report from local operate_ manager_ date _date_ time _time_ elr_code distance_tag issue trespasser on railway_ line_ in geo_place junction_ area_ train_ warning_ reported all clear_ by mobile_operations_manager_ _time_ action fence_ check_ _date_ geo_place

Table 1: Cleaning, tagging and tokenization processes in one trespass record

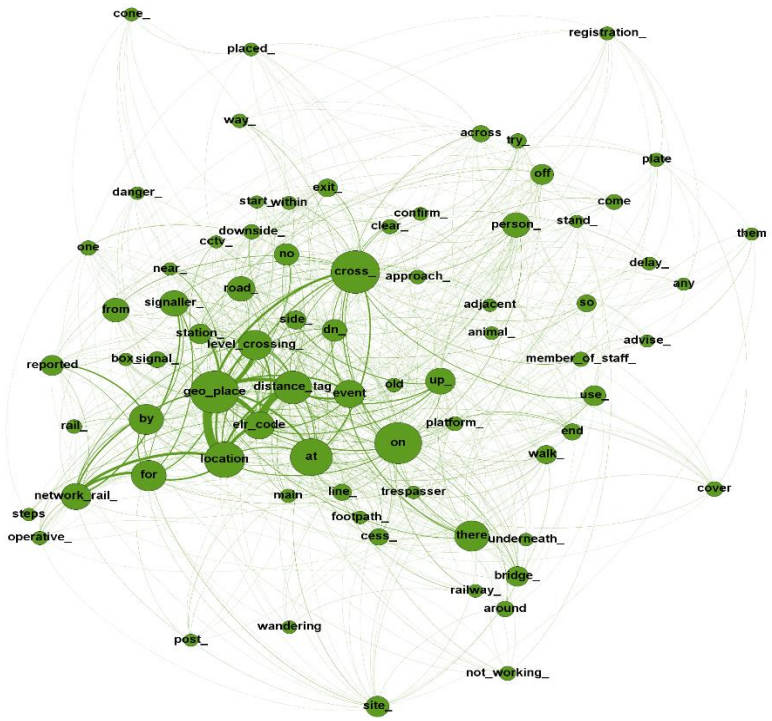

Figure 5. Network graph of Close Call word associations filtered the nodes larger than 20 degree.

Figure 6: RAATS graphical user interface (GUI).

Analytics (RAATS)

The last, and arguably most important enabler is analytics. Analytics turn data into information. The RAATS project does exactly that. An event where a train passes a signal showing a red (stop) aspect without authorization is known as a signal passed at danger (SPAD). SPADs can range from minor incidents where a signal is passed by only a few meters, to events that results in train collisions: following a SPAD at Ladbroke Grove in 1999, a collision occurred which resulted in 31 fatalities (HSE, 2000). Since that time, the GB rail industry has made significant efforts to reduce the rate of SPADs.

Traditionally, SPAD risks are analysed using a process which examines the potential consequences of passing a particular signal at danger. A weakness in the traditional analysis was that it is unknown how many times trains approach a signal when it is displaying a red aspect. The RAATS project addresses that shortcoming by analysing live data from signalling systems. The source of the information used in the RAATS software is train describer data (NR, 2015). A train describer is an electronic device connected to each signalling panel which provides a description of each train (its 'headcode') and which section of track (or 'track section') it currently occupies. RAATS software reads the train describer live-feed, stores the data, and most importantly calculates which trains actually approach a red aspect. The red approaches to a single signal can be analysed over a period from a single day to a period of a year. Alternatively the user can choose to analyse all signals in an area or even all the signals in the database.

Figure 6 shows the RAATS graphical user interface. This image shows the results for a single signal ET776 which is located on the up Cowdenbeath line at Redford. The figure shows that 23% of trains approached the signal at red in the period of the 17 August to 13 October 2014, which is a high percentage compared with the average for the network. In this way, RAATS software analytics provide intricate details about the number of trains approaching a signal at danger. RAATS adds value to safety on the GB railways by analysing a large live data feed in a way that supports risk analysis.

CONCLUSIONS

The GB Railways are a source of large amount of data from a variety of data-sources, producing data very quickly. The BDRA research programme at the University of Huddersfield aims to investigate whether and how rail safety and risk management could benefit from the new software tools that computer scientists develop today. BDRA is a step-change in the accuracy of risk estimation because it provides the toolbox to perform statistical analysis of *all* relevant risk data rather than estimating probabilities from samples or limited datasets, and therefore reduces the need for risk-based models that rely on assumptions or simplifications to produce risk estimates.

Four enablers have been identified for BDRA: data, ontology, visualisation and analytics. To date, two data sources have been analysed to extract safety information from signals and free text reports of railway hazards. Visualisation techniques have been applied to support the data analysis. A relatively simple form of ontologies has been used in the text analysis. More complex ontologies can be used to classify the concepts that allow

combining the information obtained from the different data sources. Analytics have been used to assess how many trains approach red aspects. Developing small software applications that combine the four enablers have shown that the overall BDRA function has been successful in our research to date.

Whilst this initial work shows how BDRA could benefit safety and risk management for railway safety in Britain, it has also demonstrated that implementation and integration of automated data-analytic techniques for safety and risk is not straightforward. The BDRA process requires novel risk analysis techniques, semantic techniques, interactive visualisation techniques for performing data analysis and dedicated computer systems; many of which have to be researched in dedicated collaboration projects between safety scientists, information technologists, software developers and railway engineers.

ACKNOWLEDGEMENT

RSSB is gratefully acknowledged for co-funding this work through the Strategic Partnership MoU of the 8th of August 2013

REFERENCES

Allen JF (1994) Natural language processing. Encyclopaedia of Computer Science, New York: John Wiley and Sons Ltd.

Bird, S., Klein, E. & Loper, E., (2009) Natural Language Processing with Python, Available at: http://www.amazon.com/dp/0596516495.

Blondel, V.D. et al., (2008) Fast unfolding of community hierarchies in large networks. Networks, pp.1–6.

Dahlgren K (1995) A linguistic ontology, Int. J Human-Computer Studies 43: 809 – 818.

Dale R, Moisl H & Somers H (2000) Handbook of natural language processing , New York: CRC Press.

Figueres M, Hughes P & Van Gulijk (2015) The role of data visualization in Railway Big Data Risk Analysis, in: proceedings of the 25th European safety and reliability conference, ESREL 2015, Zurich, Swizerland, 7-10 September 2015.

Genesereth MR & Nilsson NJ (1987) Logical foundation of artificial intelligence, Morgan Kaufman, Los Altos CA.

Gnoni MG, Andriulo S, Maggio G & Nardone P (2013) Lean occupational safety: an application for a near-miss management system design, Safety science 53: 96ı 104.

Guarino, N (1997) Understanding, building and using ontologies, Int. J. Human-Computer Studies 46: 293 – 310.

HSE Books. (2000) Ladbroke Grove rail enquiry, Part 1 , London: Her Majesty's Stationary Office.

Hughes P & Figueres-Esteban M (2015) Learning from close call events , Report: IRR 110/89, Huddersfield: IRR.

Hughes P, Figueres-Esteban M & Van Gulijk C (2015) Learning from text-based close call data, in: proceedings of the 25th European safety and reliability conference, ESREL 2015, Zurich, Switzerland, 7-10 September 2015.

Noy NF & McGuinness DL (2001) Ontology Development 101: A Guide to Creating Your First Ontology, KSL report KSL-01-05/SMI-2001-0880.

NR (2015) http://www.networkrail.co.uk/data-feeds/, accessed 10 April 2015.

Paranyushkin, D., (2011) Identifying the Pathways for Meaning Circulation using Text Network Analysis. Nodus Labs, pp.1–26.

Smith B & Welty C (2001) Ontology: Towards a New Synthesis, in: Second Formal Ontology and Information Systems, October 17-19, Ogunquit, Maine, USA: Iii – ix.

Stow J, Zhao Y & Harrison C (2015) Estimating the frequency of trains approaching red signals – a key to improved understanding of SPAD risk, (submitted manuscript to Journal of Engineering).

Van Gulijk C, Hughes P & Figueres-Esteban M (2015) Big Data Risk Analysis for rail safety?, in: proceedings of the 25th European safety and reliability conference, ESREL 2015, Zurich, Switzerland, 7-10 September 2015.

W3C (2015) http://www.w3.org, accessed 10 April 2015.

Wu J & Heydecker BG (1998) Natural language understanding in road accident data analysis, Advances in engineering software 29: 599-610.