

Data Fusion of Georeferenced Events for Detection of Hazardous Areas

Sérgio Onofre, João Gomes, João Paulo Pimentão, Pedro Alexandre Sousa

▶ **To cite this version:**

Sérgio Onofre, João Gomes, João Paulo Pimentão, Pedro Alexandre Sousa. Data Fusion of Georeferenced Events for Detection of Hazardous Areas. 8th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), May 2017, Costa de Caparica, Portugal. pp.81-89, 10.1007/978-3-319-56077-9_7 . hal-01629570

HAL Id: hal-01629570

https://inria.hal.science/hal-01629570

Submitted on 6 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution| 4.0 International License

Data Fusion of Georeferenced Events for Detection of Hazardous Areas

Sérgio Onofre[1] , João Gomes[2], João Paulo Pimentão[2] and Pedro Alexandre Sousa[2],

[1] Holos SA, Caparica, Portugal, onofre@holos.pt
[2] Faculty of Sciences and Technology, NOVA University of Lisbon,
Monte de Caparica, Almada 2829-516 Portugal
jfn.gomes@campus.fct.unl.pt, {pim, pas}@fct.unl.pt

Abstract. When dealing with events in moving vehicles, which can occur over widespread areas, it is difficult to identify sources that do not derive from material fatigue, but from situations that occur in specific spots. Considering a railway system, problems could occur in trains, not because of train's equipment failure, but because the train is crossing a specific location. This paper presents a new smart system being developed that is able to generate geo-located sensor-data; transmit it for smart processing and fusing to the inference engine being built to correlate the data, and drill-down the information. Using a statistical approach within the inference engine, it is possible to combine results collected over long periods of time in a "heat-map" of frequent fault areas, mapping faulty events to detect hazardous locations using georeferenced sensor data, collected from several trains that will be integrated in these maps to infer high probability risk areas.

Keywords: Data-fusing, Smart System, Industry 4.0, IoT, Event Geographic position, Data Correlation, Forecast.

1 Introduction

The association of faults to geographic locations can be used to alert and provide support to incident intervention crews to deal with current issues and, especially to prevent forecast events in a specific location in a nation-wide (or trans-national) railway network.

Deciding on whether a location is faulty or not depends on the quotient between the number of trains that crossed the railway and how much breakdowns occurred around that location. A statistical approach is used where data is be auto correlated based on past events. To get the information needed to feed the system's databases it uses a sensor fusing system, where it combines and study the data collected from the trains that populate the railway network.

This system will be integrated in a multi-agent architecture [1] used in a distributed surveillance system (DVA[1]), where events, sensors and human resources are georeferenced. New features will be added to the system such as reasoning in terms of

[1] Advanced Surveillance System, for more information see: dva.holos.pt

what should be considered a correlated event (geographically speaking) and forecast of events using the knowledge acquired from previous events and data.

This paper starts by presenting a state of the art in data fusion architectures, Geolocation sensing, data correlation, event forecasting and finally presenting the foreseen method to integrate this system in the DVA for providing data correlation and forecasting functionalities.

2 Contribution to Smart Systems

The work described in this paper will encompass the collection of real-time raw data from sensors, which are then sent to a central processing unit that will perform reasoning and prediction based on the knowledge acquired data along a large period of time. In this paper a third-generation smart system [2], is presented, that performs human-like perception and acts autonomously, i.e without any human decision. Moreover, it is equipped with the capability to predict and adapt according to this readings.

In practice, one could argue that the current system encompasses all the components that are the base for the concept of "Industry 4.0", although applied to mobility: the cyber-physical systems that is achieved by the monitoring of the trains and the availability of the data through Internet protocols; the "IoTization" of trains allowing them to be remotely sensed and monitored, and finally the application of Cloud Computing and Big Data technologies and methodologies to both gather and process the high volumes of information that each train produces every second.

3 State of the Art

3.1 Data Fusion Architectures

Nowadays sensor nodes integrate multiple capabilities that include sensing, global positioning system (GPS), computing and communication. Fusing multiple simple sensor nodes, one can deliver scalable, reliable and more complex sensor based systems.

Multiple Architectures have been proposed for sensor fusing. First example is the one proposed in Fig 1 (a) where static sensors nodes are wired to a central gateway that handles all the requests and processing. An implementation of this type is [3] where all the equipment in an office (printers, thermometers, ventilation, etc..) are connected to a central gateway. All the clients that require information connect to the gateway that proceeds to service the requests. Fig 1 (b) is used when the data collected volume is large and therefore needs some local processing before sending the compressed/processed version to the server. Finally, when different layers of processing are responsible for different aspects of data manipulation there is a hierarchical processing as presented in Fig 1 (c), in which the collected data passes

through several layers of data processing and manipulation before reaching the central server.

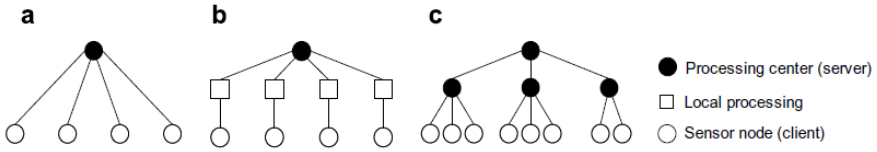


Fig. 1. Typical Sensor Fusing Architectures. [4].

Recently there has been a more hybrid approach where mobile agents are used in combination with static nodes [5]. An example of this is the Interoperable-agent-mode for sensor network [4] that is FIPA standards compliant[6], [7], which considers static wireless nodes distributed in an area and a mobile agent that goes physically to the sensor nodes to collect the data. The mobile agent is responsible for making his own itinerary around the sensor nodes and when the battery runs low he returns to the central station.

As another example, the DVA surveillance system [8] is a human-machine collaborative distributed system where static nodes are in place and in case of alert the mobile agents (be it policeman, security, firefighter or civil protection) are called in to deal with the occurrence.

3.2 Geolocation Relation

The next question that arises is how one should relate events, since the goal is to associate locations to breakdowns. It makes sense to define a radius around said events so that it is possible to search for previous nearby occurrences (since location is not 100% accurate). Speed (Velocity) of the train is an important aspect to consider in defining said radius since the faster the train goes the more difficult it is to track down the exact location of the breakdown, therefore, the radius should be wider.

Several approaches have been developed in terms of geolocation radius but in different contexts such as [9] where automatic safety envelopes (that can be both dynamic or static) are defined around different construction equipment based on their width, length and velocity, the goal here is to alert the workers that can, without their knowledge, be putting their safety in jeopardy by going too near dangerous equipment. These envelopes radius are created according to the equation (1).

$$R = \sqrt{(\frac{L}{2} + r + \frac{vt}{2})^2 + (r + \frac{w}{2})^2} \text{ where } \quad v = |\vec{v}| \tag{1}$$

Where L,W,v are the length, width and velocity (both in X and Y axis) respectively of the moving equipment.. The r represents the Safe distance of the equipment (which means the distance it will go before completely stopping), t represents the time it will take to break.

3.3 Event Processing

To make the system reliable and robust it is very important not to have false positives that can mislead the analysis. Therefore, there is a need to implement a layer between the sensors and the interface, so that it can analyze the raw sensor data and do the reasoning so to differentiate between false alarms and real events. Model-Based approaches have been proposed in [10] applied to supply chains where the authors propose a set of resources, orders, specifications and a set of milestones throughout the supply chain. The milestones are then monitored and agents actively look for deviations of the expected models. Fuzzy logic has also been widely used in the literature to solve these kinds of problems.

In [11] the authors define multiple arrays of input variables (Vi), their domains (DDV), and their constraints (Φ) the values that the system receives from each individual set of input variables are classified in terms of "normality" (Ncn) which is a scalar between [0,1] being 1 the most normal situation and the 0 the most abnormal. Several solutions have been proposed to make decisions towards this "normality factor". One simple solution as presented in [11] is finding the minimum value of normality (Ncn) and consider only this value to classify the situation, however, this is limiting because it is only considered one set of input variables while the others are discarded, which means that there is the possibility that values with information important for the classification are being ignored. Choquet Integral, as defined in [11] proposed, to define specific weights **(μ(Ai))** to normality concepts and correlate them. Equation (2) it is shows how Choquet Integral correlates the normality factors.

$$C_{\mu}(N_{c_1}^{j}, N_{c_2}^{j}, .. N_{c_n}^{j}) = \sum_{i=1}^{n} (N_{(i)} - N_{(i-1)}) \times \mu(A_i) \quad \textbf{(2)}$$

This means that once the normality concepts are defined (Nc1, Nc2…Ncn), it is needed to relate them by defining weights. The conclusions are made based on the final value of the equation. The OWA aggregation operators presented in [12], use the normality factors times a weight this makes a junction between different criteria (normality factors) using weights. In (3) is presented how the OWA aggregators are calculated.

$$OWA(a_1, a_2, .. a_n) = \sum_{j=1}^{n} w_j . b_j \quad \textbf{(3)}$$

Where (a1,a2,…,an) represents the values that are collected to make the decision, wj represents the weights and bj is the ordered set of (a1,a2,..an).

3.4 Event Prediction (Forecasting)

In [12] it is presented a way to predict events in real time streams of data. The authors look for event signatures (what happened before the event) and make sequences of readings that lead to the event. The readings are then compared with the sequences in the database that are known to cause breakdown events. Since there can be a lot of events signatures in the database it is not feasible to compare every sensor reading with the values in the database, the authors, propose to look for triggers. These triggers are then used in forward rules which means, for example, that If Trigger 1

and Trigger 2 are found THEN event C will happen. In the paper it is also proposed to once a new signature is found to generate readings (real or unreal) that lead to this event. The idea is to have a model of the event.

4 Architecture

The DVA system as explained in [13] is "is a Geo-referenced multi-agent surveillance system, composed by several agents: Sensor agent – provides sensor information; Processor agent – transforms sensor information into parameters; Inference agent – uses parameters in rules for event detection; Action agent – executes predefined actions for each event; Backup agent - stores all the system information; Interface agent – shows (in maps) the values of the sensors, events, actions and system status; Mobile agent – Associated with a human, equipped with a mobile device who is responsible to perform events' actions, such as confirming the event or handling the event; Monitor agent – monitors all system's agents, ensuring correct system performance." In this paper, two new agents to be added to this architecture are presented: "**correlate agent**" (that makes links between events) and the "**forecast agent**" (that looks for current sensor readings and predicts problems that might arise in the near future).

4.1 Proposed Architecture

The Fig 2 represents the proposed changes (in red) to the DVA architecture. The "Correlate" Agent will be activated every time a new event is detected. It will look for previously events nearby (stored in the Backup Agent) and if it detects a similar event in the area, it will create a new type of event (Correlated Event) that will be shown on the interface. The Forecast Agent is receiving the data from the processor and previously readings stored in the Backup Agent to make predictions about potential problems.

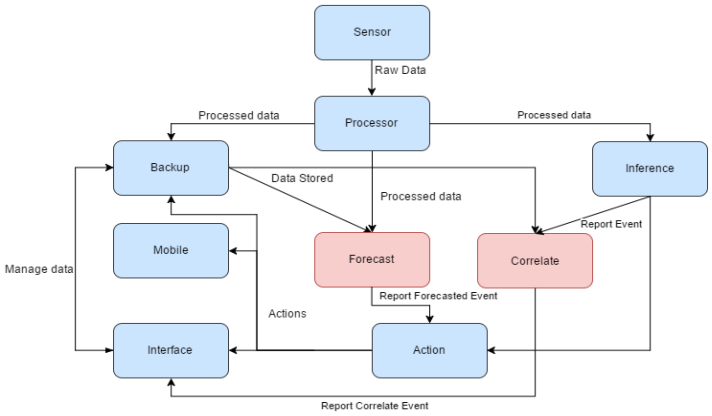


Fig. 2. Architecture Proposed.

It will also generate a new event type (Forecasted Event) in the system so it can make decisions to prevent damage.

4.2 Proposed Changes to the Architecture

Correlated Event: The correlated event, links separate events to geographical areas. It is described by: location of the **center of the event** (since the goal is to map these events, it is need a geographical center given by the center of mass of the composing events), a **geographical radius** (distance between the center of mass and the further away event), a **time of occurrence** (when it was detected), **number of occurrences** (how many events it is linking) and a **vector with the events correlated**.

To link the geographical events an adaptation is made from the equation (1) considering that the width as well as the breaking time (time to full stop) of the train are irrelevant since the events will be detected while the train is still moving.

$$R = \frac{L}{2} + \frac{v}{\delta} \text{ where } \quad v = |\bar{v}| \tag{4}$$

Where L represents the length of the train, and the v (composed by its X and Y component) represents the velocity of the train at the time of the fault detection. Delta (δ) is a parameter, between 0 and 1 that allows the adjustment of the radius according to what is verified in practice. For the time being it is considered to be 0.25s which at a standard train speed of 100km/h will produce a length of roughly 110m which is half the usual length of a train. Events are therefore correlated if the circumferences centered at the location of each event with their respective radius (R) intersect within a timeframe (the timeframe being a parameter that will be adjusted as more and more information is collected).

Forecasted Event: The Forecast Agent is actively looking for readings that can lead to breakdown, for example, if the value of a variable has been steadily rising to a point that if its growing rate continues soon it will become a problem, the agent, generates an Forecasted event with a **description** of the of the predicted event as well as the **readings relevant** and finally the **measures** that train should do to avoid damage. These are the "triggers" that were previously presented in section 3.4.

Correlation Agent: In the next figure it is shown the correlation agent's flow chart (**Fig. 3.**) where it is explained in more detail its operation. The agent is activated once a new fault event is detected. It's assumed that it is not dealing with false positives since it is already filtered using the techniques described in section 3.3. Afterwards it's defined a geolocation radius around the event as presented in section 4.2. Once the area is defined the agent looks in the event database for previous occurrences in the vicinities. Subsequently two things can happen: either it finds no other events in the area therefore the agent shuts down until another fault event occurs or it finds other events (at least 2) and generates a new Correlation Event. If the new Correlation is defined it is then sent to the Interface Agent that flags this area in a form of heat map that has the warm colors to show the areas of found malfunctioning.

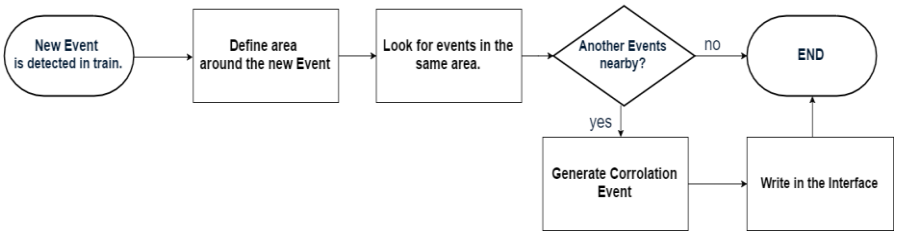


Fig. 3. Flow Chart Correlation Agent.

Forecast Agent: The Forecast Agent looks for patterns in the readings before breakdown events (called event signatures), in brief, every time a fault is detected it will make a history of readings just before the event. These readings will then be used as comparison for the active trains and if a pattern that previously led to breakdown is a found in an active train this agent will generate Forecasted Events.

There are critical points in the railway grid where power sub-stations are changed, these points usually are associated with high voltage differences (more than 5kV). These variations tend to cause peeks of currents/voltage in the trains' electrical systems, this can eventually become a source of malfunctioning or breakdown.

The forecasting needs to be done on multiple active trains in the grid, this means that it needs threads to take care of these multiple tasks. Due to the high volume of data that it is collected and sent (at a rate of 3000bps), the Apache Hadoop's Map-Reduce Paradigm (from big data analysis) is used. This gives the ability to separate trains into different parallel tasks. The Hadoop then processes all the information sent by the trains and looks for patters that can be known to be troublesome. The results are shown in the interface with a delay associated with the receiving the information, the processing and the actual writing in the interface, however, this delay is not critical since the goal of this system its not to control the train in real time, but to give insight to the maintenance crews of potential problems within the electrical grid and/or with the trains. Every time there is a forecasted event of breakdown this agent produces a report of how many critical points did the train endure before. The idea is to give an average of the number of how many critical points the trains can endure before needing a maintenance team to check its electrical protections. More detailed information about Apache Hadoop can be found in [14]. To extract knowledge the data mining techniques presented in section 3.4 are used.

5 Conclusion

DVA is a developed and tested system that protects both people and goods from all sorts of dangers (both natural and human caused) using statically nodes spread out across vast areas. In the presented approach, some of its behavior had to be adapted because we are now considering moving trains, which makes the sensors' positions change over time. So, no static location relations to identify events are now available. And this was a cornerstone for the inference engine of DVA.

In this new adaptation, the DVA architecture had to be adjusted so that it could correspond to the specific needs of the current problem, one of these was to link

multiple events to specific geographical areas. The other big need was to embed the DVA with the capability of predicting, this brings an advantage since the protection of the trains starts even before the incidents occur giving more time to warn the right mobile agents (Maintenance crews) and the right measures. The correlation brings an advantage in detection of specific dangerous locations, for example, areas that are problematic (i.e. areas where often events are triggered) that require more attention from the involved agents and be more often checked. The correlation also has an impact in how the maintenance crews are spread across their working, for example, after a careful study of the common problem areas of the agents (maintenance crews) should be closer to where the problems usually appear. This architecture is currently under implementation and results are expected to be presented soon.

References

1 N. R. Jennings, "An agent-based approach for complex software systems." Commun. ACM, vol. 44, no. 4, pp. 35–41, 2001.
2 T. Gessner, M. Messe, and F. Gmbh, "Smart Systems Integration and Simulation," Integr. Vlsi J., pp. 5–22, 2016.
3 J. Suarez, J. Quevedo, I. Vidal, D. Corujo, J. Garcia-Reinoso, and R. L. Aguiar, "A secure IoT management architecture based on Information-Centric Networking," J. Netw. Comput. Appl., vol. 63, pp. 190–204, 2016.
4 P. K. Biswas, H. Qi, and Y. Xu, "Mobile-agent-based collaborative sensor fusion," Inf. Fusion, vol. 9, no. 3, pp. 399–411, 2008.
5 Y. Xu and H. Qi, "Distributed computing paradigms for collaborative signal and information processing in sensor networks," J. Parallel Distrib. Comput., vol. 64, no. 8, pp. 945–959, 2004.
6 "FIPA97 Foundation for intel. physical agents spec. Part 1 Agent Management," Quality, 1997.
7 "FIPA 97 Specification Part 2: Agent communication language," 1997.
8 S. Onofre, P. Sousa, and J. P. Pimentao, "Multi-sensor geo-referenced surveillance system," in 10th Int. Symposium on Mechatronics and its Applications (ISMA), 2015, pp. 1–6.
9 Z. Li, S. Zhou, S. Choubey, and C. Sievenpiper, "Failure event prediction using the Cox proportional hazard model driven by frequent failure signatures," IIE Trans., vol. 39, no. 3, pp. 303–315, 2007.
10 E. Fernández, C. M. Toledo, M. R. Galli, E. Salomone, and O. Chiotti, "Agent-based monitoring service for management of disruptive events in supply chains," vol. 70 OP - I. p. 89, 2015.
11 J. Albusac, D. Vallejo, J. J. Castro-Schez, C. Glez-Morcillo, and L. Jiménez, "Dynamic weighted aggregation for normality analysis in intelligent surveillance systems," Expert Syst. Appl., vol. 41, no. 4 PART 2, pp. 2008–2022, 2014.
12 R. R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking," vol. 18, no. 1. pp. 183–190, 1988.
13 B. Dias, B. Rodrigues, J. Claro, J. P. Pimentão, P. Sousa, and S. Onofre, "Architecture and message protocol proposal for robot's integration in multi-agent surveillance system," Lect. Notes Comput. Sci., vol. 8536 LNAI, pp. 366–373, 2014.
14 M. Liroz-Gistau, R. Akbarinia, D. Agrawal, and P. Valduriez, "FP-Hadoop: Efficient processing of skewed MapReduce jobs," Inf. Syst., vol. 60, pp. 69–84, 2016.