IoT and Semantic Web technologies for event-detection in natural disasters

Journal:	*Concurrency and Computation: Practice and Experience*
Manuscript ID	CPE-18-0137.R1
Editor Selection:	Special Issue Submission
Wiley - Manuscript type:	Special Issue Paper
Date Submitted by the Author:	30-Apr-2018
Complete List of Authors:	GRECO, LUCA; University of Salerno, Dept. of Information and Electrical Engineering and Applied Mathematics (DIEM) Ritrovato, Pierluigi; University of Salerno, Dept. of Information and Electrical Engineering and Applied Mathematics (DIEM) Xhafa, Fatos; Universitat Politecnica de Catalunya, Computer Science Tiropanis, Thanassis; University of Southampton, Web and Internet Science Group, Electronics and Computer Science
Keywords:	Semantic Web, Data stream, Internet of Things, Disaster Detection.

Received 28 February 2018; | Revised ; | Accepted

DOI: xxx/xxxx

ARTICLE TYPE

IoT and Semantic Web technologies for event-detection in natural disasters

Luca Greco*[1] | Pierluigi Ritrovato[1] | Thanassis Tiropanis[2] | Fatos Xhafa[3]

[1]Dept. of Computer and Electrical Engineering and Applied Mathematics (DIEM), University of Salerno, Italy,
[2]Web and Internet Science Group, Electronics and Computer Science, University of Southampton, Southampton, UK
[3]Dept. de Ciències de la Computació, Universitat Politècnica de Catalunya, Barcelona, Spain

Correspondence
*Luca Greco. Email: lgreco@unisa.it

Present Address

Summary

Natural disasters cannot be predicted well in advance but it is still possible to decrease the loss of life and mitigate the damages, exploiting some peculiarities that distinguish them. Smart collection, integration and analysis of data produced by distributed sensors and services are key elements for understanding the context and supporting decision making process for disaster prevention and management. In this paper we demonstrate how Internet of Things and Semantic Web technologies can be effectively used for abnormal event detection in the contest of an earthquake. In our proposal, a prototype system, that retrieves the data streams from IoT sensors and web services, is presented. In order to contextualize and give a meaning to the data, semantic web technologies are applied for data annotation. We evaluate our system performances by measuring the response time and other parameters that are important in a disaster detection scenario.

KEYWORDS:
Semantic Web, Data stream, Internet of Things, Disaster Detection.

1 | INTRODUCTION

The ability to predict the occurrence of conditions that promote forest fires allows fire prevention teams to timely respond and manage targeted evacuations. In the same way, a smarter detection of mudslides, avalanches, earthquakes and other natural disasters can drastically help reducing reaction times, decrease the loss of life and mitigate the damages. In the context of disaster management, the Internet of Things (IoT) is gradually gaining attention, being able to provide affordable, lightweight and flexible solutions for early warning, notification, data analytics, knowledge aggregation, remote monitoring and victim localization [28]. In fact, natural disasters cannot be predicted well in advance but it is possible to exploit some peculiarities that allows to quickly reveal them by means of a proper combination of sensors, services and technological infrastructures [18]. For example, an earthquake generates two different seismic waves traveling at different speeds; being able to recognize them in some way allows to give an early warning (in seconds) so that people can take shelter in time. The weather condition can also be a problem for rescuers after an earthquake as well as the communication and the electrical power, that could be compromised because of power outages due to broken cables or other issues.

In such situations, it is important to guarantee access to services and continuous data collection.

The use of small single-board low power/low cost computers (lightweight nodes), which are capable of operating in conditions where source power [12] and communication bandwidth are reduced (such as during a disaster scenario), is a common trend nowadays. In order to easily access and integrate heterogeneous data, and make their retrieval and processing easier, standard technologies for data representation and management, such as Semantic Web technologies [15], can be helpful since they also allow to infer new knowledge from data through semantic reasoners: the inference process is crucial wherever there is a need to discover new relationships, analyze data, manage knowledge and discover possible inconsistencies in the data.

In this paper we propose an application of Semantic Web technologies for Internet of Things that allows to address event detection in the contest of an earthquake. In our proposal, a system retrieves the streams coming from web services collecting IoT sensors data to semantically annotate and store them in a triple store lying on a computer that communicates with lightweight nodes. The triples contain information about time, sensors position, weather condition and the earthquake. As a result, we evaluate our system performances by measuring the response time in two cases that show how the system is affordable in disaster detection scenario.

This work is organized as follows: section 2 discusses the specific problem we want to address; section 3 examines related works in the domain of IoT and earthquake detection, weather data retrieval and event detection; in section 4, we discuss the system design with reference to the four stage approach and then present the proposed architecture; in section 5 we provide some details about the implementation of the two main modules that compose our system: the Data Importer, that obtains weather and earthquakes data from appropriate services and transforms them in the triple format; the MQTT based notification Application, that provides communication between the triple store (hosted in a server) and the clients (lightweight computers). Finally, in Section 6, we discuss some results obtained during the testing stage.

2 | PROBLEM DEFINITION

Since we are interested in the development of an IoT system for earthquake detection and people alerting, our analysis starts from the following considerations:

- how could an earthquake be detected and which features can be used for early warning and alerting people in the affected areas?
- which information could be useful to rescuers (weather conditions or problems in the affected areas) so that they could organize the rescues in the shortest possible time, eventually trying to save people trapped under rubble and finding places where taking refuge?

Technically, an earthquake is defined as a shock of the ground caused naturally, as a result of the fracture and slippage of rock layers in the earth's crust. This natural phenomenon is determined by the sudden rupture of equilibrium of forces acting on the terrestrial layers, in areas characterized by instability. When tensions, accumulated in unstable areas for any reason (dislocation of the layers, chemical and physical phenomena with abrupt change of state, etc.), sharply prevail on the strength of the medium, fractures and flows are generated. At the same time, in the place of rupture, called hypocentre, elastic waves, longitudinal *P-waves* and transverse *S-waves*, are originated. The P-waves are the fastest and the first ones to be perceived: that's why they are called primary waves. P-waves are able to propagate through both rigid materials, water and air, therefore it is possible to hear a deep rumble when an earthquake starts. The S-waves instead cause shear deformations that change the shape but not the volume of the rock they come through. This kind of waves cannot propagate through fluids and are obstacled by oceans and the outer core of the Earth that is liquid.

An earthquake warning system is able to perceive P-waves thanks to a system of accelerometers and seismometers, and provide notifications of an earthquake while it is in progress. In particular, Earthquake Early Warning (EEW) systems use earthquake science and the technology of monitoring systems to alert devices and people when shaking waves generated by an earthquake are expected to arrive at their location. In this paper, we propose an IoT application for disaster management that leverages Semantic Web Technologies to address event detection in the contest of an earthquake. Here Semantic Web (SW) technology plays a fundamental rule: first it allows to give a "meaning" to all data provided by the different sensors in terms of geographical position, weather condition, city, etc... then, it also allows to organize all data and events in a machine understandable form, so that they can be easily accessed and analyzed by different kind of algorithms. Finally, leveraging on triple store performance, it is possible to define new analytics functions for earthquake using standard languages like SPARQL or infer new knowledge using reasoners and rule based systems.

3 | RELATED WORKS

The use of IoT technology for disaster detection [28] and management is a research area that is gaining more and more interest nowadays. In [7] a roadmap highlighting the possible use of next generation emerging technologies for enabling collective computational intelligence in managing disaster situations is presented. The authors discuss a relevant scenario to illustrate the model architecture and detail the proposed roadmap. Machine learning techniques for distributed event detection in Wireless Sensor Networks have been analyzed in [8], where performance and applicability for early detection of disasters (especially residential fires) are considered. The "Riskr" project [5] applies a low-technological solution for creating a disaster portal fed by social networking messages coming from Twitter. The results of users' tests suggest that the combination of online services and interoperability between disaster portals and social networks can enhance disaster management initiatives. The use of mobile ad hoc networks (MANETs) for a range of post-disaster circumstances has been investigated also in [16] where real-time data and information sharing, using a mobile app, has been expanded from individual point-to-point communication or 'hot-spots' to dynamically built environments, with virtual hub-and-spoke topology, referred as mobile ad hoc networks [17].

3.1 | IoT and earthquake detection

In terms of prevention from earthquakes, overhead shots have already proven to provide valuable information and data in the area of analysis at all levels, from satellite to plane or UAV (Unmanned Aerial Vehicle) shootings. An example of this is represented by the experiment of an Italian-English group in Iceland: they developed a research based on shooting with drone at low altitude and with high resolution so as to simulate very accurately the topography of the area and recover precious information about the events that had historically affected the territory and risk factors related to seismic events. The experiment included not only the development of a virtual reality solution but also the management of analytics applied to spatial data [21]. Another example is the solution developed by the Turin SysDev which allows the structural and environmental monitoring in buildings or infrastructures (i.e., stations, bridges, tunnels) to improve the control of the territory and its resilience in case of earthquakes or other dramatic events such as terrorist attacks. SysDev worked on the sensors integrated in building materials, being able to sense the vibration of the structure to which they are applied to and to transmit and process them in real time identifying possible troubles [22]. In terms of Earthquake Early Warning (EEW), *I.Co.*, an Italian company, created a seismic alarm that collects data about P-waves. In particular, when the sensor perceives the P-waves, it sends acoustic and optical alarm allowing to gain the longest possible time to be able to act [23]. A beacon is also available for earthquake and tsunami early warning. This sends earthquake data in real-time to the Brinco Data Center (BDC) from where it is possible to forward to chosen national and regional earthquake networks. The project is actually supported using *Indiegogo.com*.

An example of IoT system for EEW using a wireless sensor network is presented in [24]. The main purpose is to detect earthquake and alert people by immediately sending notifications through a smartphone. Components used for designing the network are: accelerometer, LCD display and Peripheral Interface Controllers. ZigBee is used for communications due to its low power consumption. The accelerometers are designed to measure the earth ground shake and can sense P-waves and S-waves. Sensors send the alert message as soon as they sense the P-waves. In [25] alerts about earthquakes are sent using Instant Messaging: the highest priority is given to IoT devices when they need to send message through the network. The system was tested with a MATLAB simulation and an Android application was developed to receive the alerts. In [26] smartphones are used as the main processing units for EEW; here a server performs spatial and temporal analysis and sends messages using Message Queue Telemetry Transport (MQTT) while a Control center manages the emergency.

When an earthquake occurs, it is also important to evaluate the seismic loss, considering building damage, casualties and so on. To this aim, the work in [27] presents a layered system with acceleration sensors that collect the ground surface acceleration and gateways that submit data to a remote monitor center. The monitor center provides the process, computation, analysis of the data and storage of the results into a database server. The data are classified into four categories: regional geographic information, seismic disaster estimation information, assistant decision making information and emergency response information. After an earthquake, it is important to manage the emergency. In [29] an ontology to model knowledge about sensors and actuator networks for earthquake emergency is designed starting from SSN. The goal is to show how the developed ontology can be effectively used to model the earthquake emergencies domain.

Some other interesting works have been proposed to provide highly resilient data transfer facility as in [11] where the topology independent bypass network-based NerveNet is discussed or in [6] where the SCALE (Safe Community Awareness and Alerting Network) framework is proposed for multiprotocol data transceiving. A management information system for earthquakes is described in [44] where RFID technology and positioning technology are exploited to help field team leader to manage effectively the resources on the earthquake field.

3.2 | IoT and weather data retrieval

In our work, we are also interested in retrieving data about weather conditions that could be provided to rescuers. Such data can be nowadays obtained from a lot of sources (i.e. sensors on vehicles, weather stations, etc.) to be collected and further pooled into the cloud for the analysis. [30] In this context, several solutions [32] have been proposed to address problems in different domains (for example remote monitoring stations for agriculture purposes that measure atmospheric and ground conditions such as soil moisture). With IoT, it is quite simple to create a personal weather station for data acquiring and weather monitoring.

Different protocols can be used to send the data from the sensors to an application that can manage and elaborate them. In [33] a personal weather station is built to retrieve information about temperature, wind speed, humidity, rainfall, atmospheric pressure and air pollution using appropriate sensors and a Raspberry Pi. The TCP/IP protocol suite is used to send data to "SparkFun" cloud. These data can be read and shown to the users. Other examples of personal weather station can be found in [34]. In this case the focus is the power consumption so a restrict number of sensors are used: temperature sensor, CO2 sensor and humidity sensor. All data are sent using TCP/IP protocol suite to "ThingSpeak" website and plotted into charts to provide a better visualization. It is possible to use also indoor and outdoor sensors in a personal station to manage dangerous situations, such as fire or gas problems [35]. Data retrieved from weather station, both personal and public, can be also represented using ontologies. In [36] an ontology to represent data retrieved from AEMET, the Spanish Public Weather Service, is presented. In particular, the W3C Time ontology is used to model time knowledge and the Semantic Sensor network (SSN) ontology is used to model the network of sensors and weather stations. Moreover, information are extracted from the triple store using SPARQL queries and presented by means of a generic visualizer that considers the geo-position of the data.

3.3 | IoT for event detection

An event, according to the definition[1], is: "something that occurs in a certain place during a particular interval of time". It is possible to identify a lot of events in different contexts (i.e. a goal in the sport domain, a car that enters a parking area in the surveillance domain, an earthquake as a natural event). Studies about event detection using IoT have recently gained attention. Interesting applications vary from industrial safety and security to meteorological hazard, earthquake, and fire detection [37]. In the field of event detection, there are two major approaches: centralized and decentralized. In centralized methods the focus is on data-aggregation for reducing communication overhead. In decentralized methods the focus is on networking aspects. For example in [38] online and distributed event detection in the Wireless Sensor Network is discussed. Each sensor node is responsible for deciding whether an event occurs using a classifier or a neural network. Another approach, showed in [39], is to use a fuzzy logic for the event detection avoiding the false alarms. Compared to using a classifier, fuzzy logic allows the detection algorithm to maintain a high accuracy level despite fluctuations in the sensor values but the problem is the number of rules to be fired for event detection that can impact the temporal and spatial efficiency.

4 | SYSTEM DESIGN

In this section we discuss in detail the design of the proposed system. We start from the development approach, then we introduce technologies and present a high level architecture of the system.

4.1 | Four-stage approach

For designing the architecture, we adopted a four-stage approach [50]:

- Stage 1 consists of the network connected devices such as wireless sensors and actuators. Sensors collect data from the environment or object under measurement. Actuators can intervene if changing the physical conditions that generate data is needed. In an IoT architecture, some data processing can occur in each of the four stages. However, processing data at the sensor level is limited by the processing power available on each IoT device. The more immediate the need for information, the closer to the end devices the processing needs to be. For deeper insights that require more extensive processing, data should be sent into a cloud or data center-based system that can fuse several sources of data together.
- Stage 2 includes sensor data aggregation systems and analog-to-digital data conversion. The data from the sensors are typically in analog form. That data needs to be aggregated and converted into digital streams for further processing downstream. Data acquisition systems (DAS) perform these data aggregation and conversion functions. The DAS connects to the sensor network, aggregates outputs, and performs the analog-to-digital conversion. The Internet gateway receives the aggregated and digitized data and routes it over Wi-Fi, wired LANs, or the Internet, to Stage 3 for further processing. The pre-processing is important because the analog raw data streams that come from sensors create large traffic.
- Stage 3 is where the aggregated and digitalized data are ready to be analyzed before sending them to the data center. In the Stage 2 data are only pre-processed but they can still easily eat up network bandwidth and swamp the data center resources. Because of this, it's better to have systems at the edge capable of performing analytics as a way to lessen the burden on core IT infrastructure. With this staged approach, it is possible to pre-process data, generate meaningful results, and pass only those on.
- Stage 4 is where data are actually analyzed, managed and stored on traditional back-end data center systems. Data that needs more in-depth processing, and where feedback doesn't have to be immediate, gets forwarded to physical data center or cloud-based systems, where more powerful IT systems can analyze, manage, and securely store the data. It takes longer to get results when waiting until data reaches Stage 4,

[1] source: *www.dictionary.com*

but it is possible to execute a more in-depth analysis, as well as combine sensors data with data from other sources for deeper insights. Stage 4 processing may take place on-premises, in the cloud, or in a hybrid cloud system, depending on the data stream rate requirements.

In this scenario, semantic technologies play a central role at stage 3, for the annotation of collected data, and then at stage four where reasoning and triple storing take place.

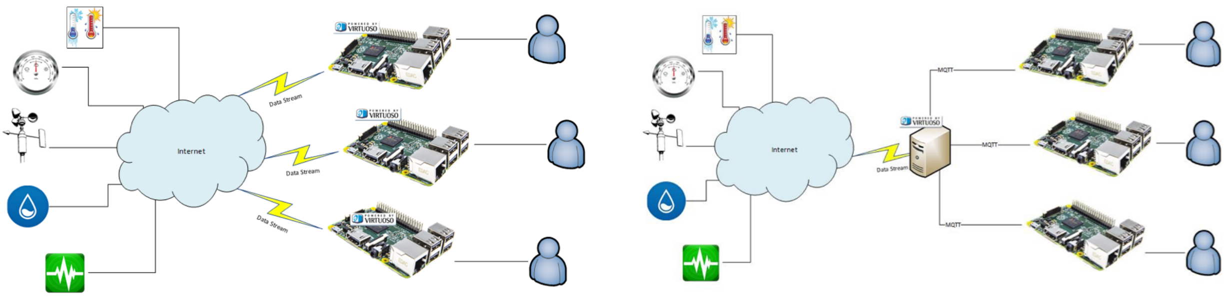

FIGURE 1 Proposed architectures: (left) no server (right) with intermediate server

4.2 | Proposed architectures

Two architectures (with and without an intermediate server) are proposed and compared (Fig. 1). In our case study, Raspberry Pi 2 model B is chosen as the system responsible for interacting with users (our lightweight node). The reason for using Raspberry is essentially due to its versatility, computational power (despite the reduced size) and low power consumption, that make it a viable solution also in IoT applications where scaling is a matter and some intelligence needs to be endowed inside the devices [14] [9] [10]. The server is a PC with Windows 10 x64 where the triple store runs. Triple store is a database where the records are organized into triples that, according to Semantic Web technologies formalism, are characterized by subject - predicate - object. Among the many triple stores available, Virtuoso has been chosen. Communications between server and Raspberry use MQTT protocol.

In the first architecture, sensors for weather conditions and sensors for earthquakes send their data to the servers. Then, using the Raspberry Pi, these streams are retrieved and stored in Virtuoso as triples. A user can further query the triple store to obtain information needed. The Raspberry Pi is able to send alerts about weather condition changes or about earthquakes directly to the user. In the second architecture, Raspberry Pi doesn't retrieve the data stream directly from the Internet but a server provides the required information and stores them into Virtuoso. On the Raspberry Pi, a user can retrieve data from the server sending a SPARQL query and waiting for the response. Using this approach, the user can have a quick response from the system instead of retrieving all the available data. The streams are then stored into Virtuoso as triples, just like in the previous case. On the server, a Java application is in charge of retrieving the streams and store them. Anyway, in both architectures, the streams are supposed to be retrieved from the Internet. The only difference lies in the presence or the absence of the server as an intermediate operating entity between the Internet and the Raspberry Pi. Of course, the server allows to perform more operations given a time slot with respect to a Raspberry Pi. Moreover, in a server based architecture, the Raspberry Pi can give to the user a quick answer without having to store the triples. The quick response could help users to predict and manage natural disaster in time to reduce the loss of life or mitigate the damages.

5 | IMPLEMENTATION

After discussing the design and the architecture, we provide some details about the modules that compose our system.

5.1 | Data importer

The aim of this module is to retrieve data from available web services and store them as triples. There are several Internet weather services that provide forecasts for several days. The further the time described by a forecast lies in the future, the more inaccurate the forecast becomes [40], [41]. The period of 24 hours has been chosen as a compromise between the deteriorating accuracy of weather forecasts over time and the time period the project requires weather data for. A *weather state* is determined by measuring values of certain elements such as temperature, relative and absolute humidity, dew point temperature, wind speed, wind direction, precipitation, cloud coverage, and others [42]. Apart from weather elements, Internet weather services often provide some information that are called *weather condition*. Generally speaking, the weather condition is a one-word description of the current weather situation. Examples for the general weather condition are Sun, Cloudy, Rain, and Fog. Some weather conditions can be split up into several conditions, e.g. it is overcast and raining into Overcast and Rain. Source for weather data is weather web service. There is a tremendous amount of services providing weather data over the Internet. However, only a small number of these services is suitable for usage in this context. The services differ regarding the way data is provided, the data format, the area being covered, the terms of use etc. The access to Internet weather services in this project is implemented in Java, hence an important question about a certain weather service is whether it can easily be accessed from a Java program. For the evaluation of Internet weather services, the following aspects were examined: coverage area, data format, data access, access restrictions, terms of use, documentation, stability, weather elements, time frame, weather updates. Based on these aspects and on the assumption that JSON format had to be used, two alternatives were selected: Weather Underground[2] and World Weather Online. Finally, World Weather Online has been chosen for its better stability when retrieving the data.

In order to identify which kind of earthquakes data were necessary for our aims, we identified a set of guiding questions:

- Which principal city is the closest?
- What is the magnitude?
- What are the exact coordinates (latitude and longitude)?
- When has it occurred?
- What is the depth where it started?

Based on these requirements, we found a couple of services suitable for the project: USGS Earthquake and INGV Service. Our final choice was INGV Service because of the covered area, since our case study is based in Italy.

The two services use different format to present their data. World Weather Online use JSON format, while INGV use plain text format. This led us to develop two different modules for the two services. For World Weather Online, the JSON is parsed using an external library [48]; every field of interest is parsed and the triples are created and stored as N-Triple format into a file. The data are retrieved for six principal cities in the center of Italy: Macerata, Ascoli Piceno, L'Aquila, Rieti, Pescara, and Perugia. The data stored into triples are expressed using the metric system and time is expressed using UTC. For INGV, the plain text is retrieved and the information needed are saved as triples (same format of the weather triples) into a file. Once the dataset with the triples has been built, it has been added to a triple store.

5.2 MQTT Application

In this section we give some details about the implementation of our MQTT application. MQTT stand for Message Queue Telemetry Transport[3] and it is a lightweight protocol for messaging, based on the pattern publish/subscribe.

In this protocol three actors are involved:

- *Publisher* connects to the Message Broker and publishes contents;
- *Subscriber* connects to the Message Broker and subscribes to interested contents;
- *Message Broker* provides the distribution of published contents to correct subscribers.

An important feature of MQTT is its possibility to provide three Quality of Service levels:

1. *Fire and forget*: A message is sent once and no acknowledgement is required.
2. *Delivered at least once*. A message is sent at least once and an acknowledgement is required.
3. *Delivered exactly once*. A four-way handshake mechanism is used to ensure the message is delivered exactly one time.

[2] https://www.wunderground.com/
[3] http://mqtt.org/

Although MQTT runs on TCP, it is designed to have low overhead compared to other TCP-based application layer protocols. Moreover, the publish/subscribe architecture is suitable for IoT because messages need response. This means lower network bandwidth and less message processing that actually extends the lifetime of battery-run devices. To ensure security, MQTT brokers may require username/password authentication which is handled by TLS/SSL. Since the last quarter of 2014, MQTT has become an ISO standard (ISO/IEC PRF 20922)[4]. In our system, we make use of MQTT to handle communication between the server with Virtuoso and the clients. The MQTT module was developed in Java and the relative project is composed by two main class: one for the server and the other for the client. The project presents a generic class for a generic MQTT client, that could be publisher, subscriber or both. Two classes allow to manage what happens when a message is received. For the server part, the class *MQTTCallbackServer* allows to receive the SPARQL query, execute it and send the results to the client. For the client side, instead, the class MQTTCallbackClient allows to show the results of the query. The class Utils is a helper class that contains methods and constant used into the project. The application depends on the Apache Jena framework, JSON library, and Eclipse Paho for Java. MQTT Application comes with a build script for Apache Ant that provides target definitions for compiling, using, and running the application in order to generate three JAR files: the executable, the sources, and the Javadoc.

6 | EXPERIMENTAL STUDY

The developed application has been tested first with JUnit and then onboard to validate the reliability of the operation, the correct interaction with the triple store and its SPARQL endpoint and the protocol between client and server. Classes to perform analyses about the response times of the system have been also included. In particular, we estimated the time needed by the server to receive a message from the client, elaborate and send a response. This approach does not consider delays occurring when a message is sent or received from a host, since MQTT generally allows to send many messages that are stored into a queue and then processed one by one. The approach has been validated using the PC as server with Virtuoso and the Raspberry Pi 2 Model B as client. It has been also simulated the case where more clients want to retrieve information from the triple store. The results, for each client, have been saved into different files and the response times have been analyzed to evaluate the entire system, comparing with the case of a client running on a PC. To evaluate the system, a list of 36 different SPARQL queries is sent to the server; for each query the answers from the triple store are retrieved and sent to the client. Fig. 2 shows an extract from the list of the 36 sample queries used for the experimental study. The client sends the same queries three times, using different topics. In this way, we could also observe data caching influence on performances. In Fig.3 we show the response times for each SPARQL query, when the client application is running on a PC and on a Raspberry. Globally, we can observe a little deterioration in response time when using a Raspberry client with respect to PC. However, in our worst case (most complex query with ID 25), the response time does not exceed 17 seconds.

```
SELECT DISTINCT ?event ?timeT ?mag ?cond ?cityT WHERE { ?event rdf:type earthres:Earthquake; earthprop:city
?cityT; earthprop:mag ?mag ; earthprop:dateTime ?timeT . ?weather rdf:type weatherres:Weather;
weatherprop:city ?cityW ; weatherprop:dateTime ?timeW ; weatherprop:condition ?cond . FILTER (?cityW=?cityT
&& year(?timeW)=year(?timeT)&& month(?timeW)=month(?timeT)&& day(?timeW)=day(?timeT)&&
hours(?timeW)=hours(?timeT))}
SELECT DISTINCT ?event ?timeT ?mag ?cond ?cityT WHERE { ?event rdf:type earthres:Earthquake; earthprop:city
?cityT; earthprop:mag ?mag ; earthprop:dateTime ?timeT . ?weather rdf:type weatherres:Weather;
weatherprop:city ?cityW ; weatherprop:dateTime ?timeW ; weatherprop:condition ?cond . FILTER (?cityW=?cityT
&& year(?timeW)=year(?timeT)&& month(?timeW)=month(?timeT)&& day(?timeW)=day(?timeT)&&
hours(?timeW)=hours(?timeT) && ?mag>2)}
SELECT DISTINCT ?cityT ?timeT ?mag ?cond WHERE { ?event rdf:type earthres:Earthquake; earthprop:city ?cityT;
earthprop:mag ?mag ; earthprop:dateTime ?timeT . ?weather rdf:type weatherres:Weather; weatherprop:city
?cityW ; weatherprop:dateTime ?timeW ; weatherprop:condition ?cond . FILTER (?cityW=?cityT &&
year(?timeW)=year(?timeT)&& month(?timeW)=month(?timeT)&& day(?timeW)=day(?timeT)&&
hours(?timeW)=hours(?timeT))}
SELECT DISTINCT ?cityT ?timeT ?timeW ?mag ?cond WHERE { ?event rdf:type earthres:Earthquake; earthprop:city
?cityT; earthprop:mag ?mag ; earthprop:dateTime ?timeT . ?weather rdf:type weatherres:Weather;
weatherprop:city ?cityW ; weatherprop:dateTime ?timeW ; weatherprop:condition ?cond . FILTER (?cityW=?cityT
&& year(?timeW)=year(?timeT)&& month(?timeW)=month(?timeT)&& day(?timeW)=day(?timeT)&&
hours(?timeW)>hours(?timeT))}
```

FIGURE 2 Sample SPARQL queries.

The results obtained are encouraging if evaluated in the context of the temporal resolution of organizational sensor systems for disaster management, which is systematically described in the work[4].

Finally, some considerations can be made to support the proposed architecture and emphasize its strenghts:

[4] https://www.iso.org/standard/69466.html

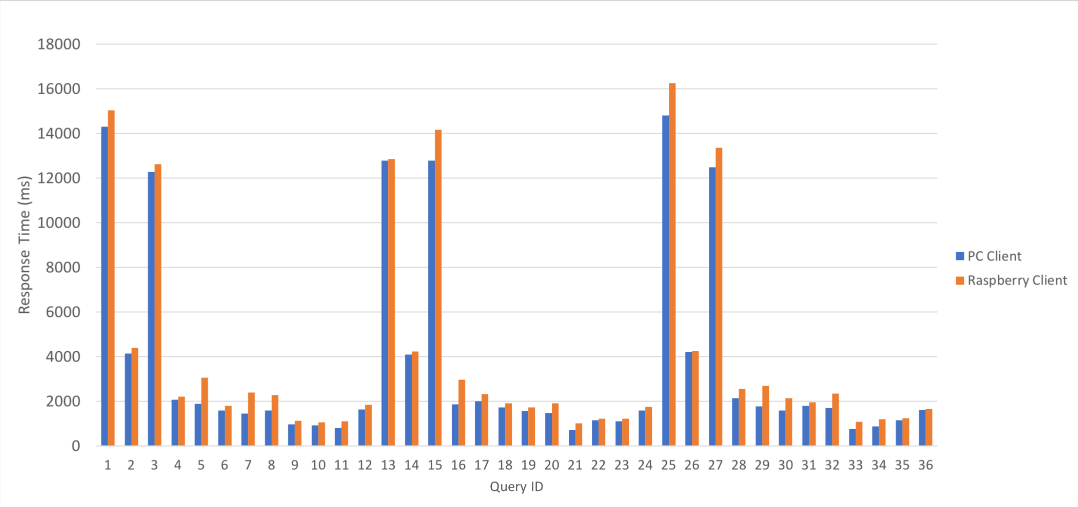

FIGURE 3 Response times comparison between PC based client and Raspberry based client for each performed query.

- thanks to the adoption of semantic web technologies, it becomes easy to add reasoning and specific rules to improve alerting functionalities, for example by defining SPARQL queries (e.g. starting from the registration of a specific event - time, zone, magnitude, etc.- alert messages can be forwarded automatically to other devices located in a geographical position close to registered event).
- the advances in the production of powerful embedded computing systems and the availability of reliable open platforms make implementation and deployment costs much reduced with respect to former approaches.
- the distribution of the workload among different lightweight nodes also contributes to improve system's resilience and robustness to fault.

Moreover, the modularity of the architecture makes it easily adaptable to other contexts where monitoring and early warnings are an issue: water pollution, public transportation systems, power plants and so on [2 1] .

7 | CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an approach to abnormal event detection, in the contest of an earthquake, that leverages Internet of Things and Semantic Web technologies. The proposed system is able to retrieve the streams from web services collecting IoT sensors data and semantically annotate them. The obtained triples are then stored in a triple store that can be easily queried using SPARQL language. System performances have been tested by measuring the response time for each defined query in two cases: PC based client and Raspberry based client. Results prove that the proposed situation is suitable in disaster scenarios where low power consumption and short response times are a concern. In future, we plan to integrate stream computing capabilities over the triples according to [51] for enabling real-time data analysis for event detection.

ACKNOWLEDGMENT

The authors would like to thank the student Carmine Di Gruttola for his contribution to the development of the application components during his Master Degree activity carried out jointly with the University of Salerno and the University of Southampton.

References

1. Finazzi, Francesco and Fassò, Alessandro, *A statistical approach to crowdsourced smartphone-based earthquake early warning systems* Journal of Stochastic Environmental Research and Risk Assessment, volume 31 No. 7 (2017) pp. 1649-1658, ISSN 1436-3259.
2. Jean Kumagai, *A Brief History of Earthquake Warnings* in IEEE Spectrum, posted on-line 1st June 2007.

3. Allen, Richard M., and Hiroo Kanamori, *The potential for earthquake early warning in southern California*. Science 300.5620 (2003): 786-789.

4. Farzad Alamdar, Mohsen Kalantari, Abbas Rajabifard, *Understanding the provision of multi-agency sensor information in disaster management: A case study on the Australian state of Victoria*, In International Journal of Disaster Risk Reduction, Volume 22, 2017, Pages 475-493, ISSN 2212-4209,

5. Jess Farber; Trina Myers; Jarrod Trevathan; Ian Atkinson; Trevor Andersen *Riskr: a web 2.0 platform to monitor and share disaster information* International Journal of Grid and Utility Computing 2015 Vol. 6 No. 2 98-112

6. K.Benson,*Enabling resilience in the Internet of Things*, in Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), Mar. 2015, pp. 230-232.

7. Nik Bessis, Eleana Asimakopoulou, Fatos Xhafa *A next generation emerging technologies roadmap for enabling collective computational intelligence in disaster management* International Journal of Space-Based and Situated Computing 2011 Vol. 1 No. 1 76-85

8. Majid Bahrepour; Nirvana Meratnia; Mannes Poel; Zahra Taghikhaki; Paul J.M. Havinga *Use of wireless sensor networks for distributed event detection in disaster management applications* International Journal of Space-Based and Situated Computing 2012 Vol. 2 No. 158-69

9. Choi, Jeong Woo, Xuanjun Quan, and Sung Ho Cho. *Bi-Directional Passing People Counting System based on IR-UWB Radar Sensors.* IEEE Internet of Things Journal (2017).

10. He, Jianhua, et al. *Multi-tier fog computing with large-scale IoT data analytics for smart cities* IEEE Internet of Things Journal (2017).

11. M.Inoue, Y.Owada, K.Hamaguti,and R.Miura,*Nerve Net: A regional- area network for resilient local information sharing and communications* in Proc. IEEE 2nd Int. Symp. Comput. Netw., Dec. 2014, pp. 3?6.

12. Liao, W., Salinas, S., Li, M., Li, P., Loparo, K. A. (2017). *Cascading Failure Attacks in the Power System: A Stochastic Game Perspective.* IEEE Internet of Things Journal.

13. Fagui Liu, Ping Li, and Dacheng Deng, ?Device-Oriented Automatic Semantic Annotation in IoT,? Journal of Sensors, vol. 2017, Article ID 9589064, 14 pages, 2017. doi:10.1155/2017/9589064

14. Lu, H., Li, Y., Mu, S., Wang, D., Kim, H., Serikawa, S. (2017). *Motor anomaly detection for unmanned aerial vehicles using reinforcement learning.* IEEE Internet of Things Journal.

15. Maarala, Altti Ilari, Xiang Su, and Jukka Riekki. *Semantic Reasoning for Context-aware Internet of Things Applications.* IEEE Internet of Things Journal 4.2 (2017): 461-473.

16. Patricia Morreale; Allan Goncalves; Carlos Silva *Mobile ad hoc network communication for disaster recovery* International Journal of Space-Based and Situated Computing 2015 Vol. 5 No. 3 178-186

17. Quang Tran Minh; Shigeki Yamada *Evaluations of commodity WiFi-based multihop access network for disaster recovery* International Journal of Space-Based and Situated Computing 2016 Vol. 6 No. 13-11

18. Stankovic, John A. *Research directions for the internet of things.* IEEE Internet of Things Journal 1.1 (2016)

19. L. Greco, P. Ritrovato, A. Saggese, M. Vento, *Abnormal Event Recognition: A Hybrid Approach Using Semantic Web Technologies*, no. 1, 2016.

20. M. Bellini, *Big Data, IoT, sensori e Mobile per la prevenzione antisismica e per l'assistenza.*[Online]. Available: http://www.internet4things.it/iot-library/iot-big-data-droni-nuovi-materiali-mobile-per-prevenire-e-ridurre-le-conseguenze-dei-terremoti/.

21. M. Bellini, I droni, IoT e realtà virtuale in Islanda per prevenire i terremoti. [Online]. Available: http://www.internet4things.it/iot-library/i-droni-iot-e-realta-virtuale-in-islanda-per-prevenire-i-terremoti/.

22. *Sysdev, startup IoT per monitorare case e ponti, raccoglie 500mila euro.* [Online]. Available: http://www.economyup.it/startup/4333_sysdev-startup-iot-per-monitorare-case-e-ponti-raccoglie-500mila-euro.htm.

23. B. D'Amico, *Perché ho fatto l'allarme antisismico per tutti.* [Online]. Available: http://www.economyup.it/made-in-italy/1727_perche-ho-fatto-l-allarme-antisismico-per-tutti.htm.

24. Alphonsa A and Ravi G, *Earthquake Early Warning System by IOT using Wireless Sensor Networks* , 2016.

25. T.-Y.Chi,C.-H.Chen,andH.-C.Chao,*An efficient notification service algorithm for earthquake early warning system* in Proc. IEEE Int. Conf. ICT Converg. (ICTC), Sep. 2011, pp. 282-287.

26. A. M. Zambrano, I. Perez, C. Palau, and M. Esteve, *Technologies of Internet of Things applied to an Earthquake Early Warning System* Futur. Gener. Comput. Syst., 2016.

27. Z. Chen, Z. Li, Y. Liu, J. Li, and J. Chen, *Quasi real-time evaluation system for seismic disaster based on internet of things* Proc. - 2011 IEEE Int. Conf. Internet Things Cyber, Phys. Soc. Comput. iThingsCPSCom 2011, 2011.

28. Ray, Partha Pratim, Mithun Mukherjee, and Lei Shu. *Internet of Things for Disaster Management: State-of-the-Art and Prospects.* IEEE Access 5 (2017): 18818-18835.

29. L.Spalazzi,G.Taccari,andA.Bernardini,*An Internet of Things ontology for earthquake emergency evaluation and response* in Proc. IEEE Int. Conf. Collaboration Technol. Syst. (CTS), May 2014, pp. 528?534.

30. A. Mahendra, *How Internet of Things will help in weather forecasting* [Online]. Available: http://iotworm.com/internet-of-things-technology-weather-forecasting/.

31. Z.-L.Tanand C.-L.Zhang, *Construction of the safety management system for urban underground business district with the application of IoT* in Proc. 10th IEEE Int. Conf. Service Syst. Service Manage., Jul. 2013, pp. 743-746.

32. P. Thibodeau, *Internet of Things brings new era of weather forecasting.* [Online]. Available: http://www.computerworld.com/article/3018390/internet-of-things/internet-of-things-brings-new-era-of-weather-forecasting.html.

33. M. Sudhakar and V. Swathi, *Real Time Weather Data Acquisition and Monitoring From a Remote Location Using ARM-11 Processor and IoT*, Int. J. Innov. Technol. Res., no. 4, 2016.

34. K. S. S. Ram and A. N. P. S. Gupta, *IoT based Data Logger System for weather monitoring using Wireless sensor networks*, vol. 32, no. 2, 2016.

35. M. Sowjanya and M. Ameenuddin, *A reconfigurable smart sensor interface for weather monitoring with iot environment*, Int. J. Eng. Sci. Res. Technol., vol. 5, no. 7, 2016.

36. G. Atemezing et al., *Transforming Meteorological Data into Linked Data*, Semant. Web, vol 1, 2011.

37. C. T. Vu, R. A. Beyah, and Y. Li, *Composite Event Detection in Wireless Sensor Networks*, 2007.

38. M. Bahrepour, N. Meratnia, and P. J. M. J. M. Havinga, *Sensor fusion-based event detection in Wireless Sensor Networks*, Mob. Ubiquitous Syst. Netw. Serv. MobiQuitous, 2009. MobiQuitous ?09. 6th Annu. Int., pp. 1-8, 2009.

39. K. Kapitanova, S. H. Son, and K. Kang, *Event Detection in Wireless Sensor Networks.Can Fuzzy Values Be Accurate?*, Lect. Notes Inst. Comput. Sci. Soc. Informatics Telecommun. Eng., 2009.

40. H. Stern, *The accuracy of weather forecasts for Melbourne, Australia*, Meteorol. Appl., vol. 15, no. January, 2008.

41. T. N. Palmer, *Predicting uncertainty in forecasts of weather and climate*, Reports Prog. Phys., vol. 63, no. 2, 2000.

42. T. S. Glickman, *Glossary of Meteorology.* 2000.

43. D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)* 2006. [Online]. Available: https://tools.ietf.org/html/rfc4627.

44. N.-N. Wu, Y. Ma, and H.-S. Huang, *Research on the management information system of earthquake field based on the Internet of Things* in Proc. IEEE Int. Conf. Inf. Syst. Crisis Response Manage. (ISCRAM), Nov. 2011, pp. 538-542.

45. NOAA, National Digital Forecast Database (NDFD) Simple Object Access Protocol (SOAP) Web Service.[Online]. Available: http://graphical.weather.gov/xml/.

46. World Weather Online, [Online]. Available: https://www.worldweatheronline.com/.

47. Norwegian Meteorological Institute, [Online]. Available: http://api.met.no/#english.

48. S. Leary, *JSON Java*. [Online]. Available: https://github.com/stleary/JSON-java.

49. D. J. Ward, *Latitude/Longitude Plotting Tool*. [Online]. Available: http://www.darrinward.com/lat-long/?id=2559471.

50. J. Fuller, *The 4 stages of an IoT architecture.* [Online]. Available: http://techbeacon.com/4-stages-iot-architecture.

51. Llaves, Alejandro, Javier D. Fernández, and Oscar Corcho. *Towards Efficient Processing of RDF Data Streams* Proceedings of the 3rd International Workshop on Ordering and Reasoning, collocated with the 13th International Semantic Web Conference (ISWC 2014). 2014.

AUTHOR BIOGRAPHY

Luca Greco is an Assistant Professor and member of the Artificial Vision Research Group at the University of Salerno. He received in 2008 the Laurea Master degree (cum laude) in Electronic Engineering, and in 2013 the Ph.D. in Electronic and Computer Engineering, both from the University of Salerno, Italy. His interests cover the areas of image and video analysis, machine learning and semantic technologies with applications like intelligent video surveillance, information retrieval and sentiment analysis. He authored more than 40 research papers in international journals, conference proceedings and book chapters.

Pierluigi Ritrovato received the master degree in computer science in 1992 and from 2016 is Associate Professor in "Information Processing Systems" at the University of Salerno teaching courses of Computer Architecture, Internet of Things and Semantic Enterprise Systems at the bachelor and master degree in Computer Engineering. In 2014, he joined to the MIVIA (Intelligent Machine for Video, Image and Audio Analysis) laboratory leading research in the area of Cognitive Vision. In the last ten years, he has been focusing his scientific research on Knowledge Management, Semantic Web, Adaptive Learning Management Systems, Grid middleware for Service Oriented Architecture. In 2001 he has been appointed by the European Commission as Locomotors, together with the United State counterpart for the "eLearning Futures and the Learning GRID" Working Group established in the frame of EU-US Cooperation programme on Science and Technologies for Learning. He has been member of the Steering Committee of the NESSI (Networked European Software & Service Initiative - from 2007 to 2009) European Technology Platform. He has been scientific coordinator of several research projects funded by the European Commission in the frame of fifth, sixth and seventh framework program.

Thanassis Tiropanis is an associate professor with the Web and Internet Science Group, Electronics and Computer Science, University of Southampton and is interested in distributed linked data infrastructures, linked data for higher education, social networks and social machines. Prior to that he was assistant professor with the Athens Information Technology Institute (AIT) in Greece working on Web technologies and e-learning. Prior to AIT, Thanassis was a research fellow with University College London (UCL) where he worked on network and service management research for telecommunication services. He holds a PhD in computer science from UCL, and a DipIng in computer engineering and informatics from the University of Patras, Greece. He is a senior member of IEEE, a chartered IT professional with BCS, a fellow of the Higher Education academy in the UK and a member of the Technical Chamber of Greece.

Fatos Xhafa received his PhD in Computer Science in 1998 from the Department of Computer Science of the Technical University of Catalonia (UPC), Barcelona, Spain. Currently, he holds a permanent position of Professor Titular at UPC, Barcelona Tech. He was a Visiting Professor at Birkbeck College, University of London (UK) during academic year 2009-2010 and Research Associate at Drexel University, Philadelphia (USA) during academic term 2004/2005. Prof. Xhafa has widely published in peer reviewed international journals, conferences/workshops, book chapters and edited books and proceedings in the field (http://dblp.uni-trier.de/pers/hd/x/Xhafa:Fatos). He is awarded teaching and research merits by Spanish Ministry of Science and Education, by IEEE conference and best paper awards. Prof. Xhafa has an extensive editorial and reviewing service. He is editor in Chief of International Journal of Grid and Utility Computing and International Journal of Space-based and Situated Computing from Inderscience and member of EB of several International Journals and Guest Editors of Special Issues. He is Editor in Chief of the Elsevier Book Series ?Intelligent Data-Centric Systems? is actively participating in the organization of several international conferences and workshops. He is a member of IEEE Communications Society, IEEE Systems, Man & Cybernetics Society and Emerging Technical Subcomm of Internet of Things. His research interests include parallel and distributed algorithms, massive data processing and collective intelligence, optimisation, networking, P2P and Cloud computing, security and trust worthy computing, machine learning and data mining, among others.

How to cite this article: Greco L., Ritrovato P., Tiropanis T., and Xhafa F. (2018), IoT and Semantic Web technologies for event-detection in natural disasters, *Concurrency Computation Practice and Experience, 2018;00:1–6*.