

## 3D-Monitoring Big Geo Data on a seaport infrastructure based on FIWARE

Pablo Fernández<sup>1</sup>  · José Pablo Suárez<sup>1</sup> · Agustín Trujillo<sup>2</sup> · Conrado Domínguez<sup>3</sup> · José Miguel Santana<sup>2</sup>

Received: 30 August 2015 / Accepted: 1 March 2018 / Published online: 22 March 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

**Abstract** Many organizations of all kinds are using new technologies to assist the acquisition and analysis of data. Seaports are a good example of this trend. Seaports generate data regarding the management of marine traffic and other elements, as well as environmental conditions given by meteorological sensors and buoys. However, this enormous amount of data, also known as Big Data, is useless without a proper system to organize, analyze and visualize it. SmartPort is an online platform for the visualization and management of a seaport data that has been built as a GIS application. This work offers a Rich Internet Application that allows the user to visualize and manage the different sources of information produced in a port environment. The Big Data management is based on the FIWARE platform, as well as “The Internet of Things” solutions for the data acquisition. At the same time, Glob3 Mobile (G3M) framework has been used for the development of map requirements.

---

✉ Pablo Fernández  
pablo.fernandez@ulpgc.es

José Pablo Suárez  
josepablo.suarez@ulpgc.es

Agustín Trujillo  
agustin.trujillo@ulpgc.es

Conrado Domínguez  
conradodt@gmail.com

José Miguel Santana  
josemiguel.santana@ulpgc.es

<sup>1</sup> Division of Mathematics, Graphics and Computation (MAGiC), IUMA, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain

<sup>2</sup> Imaging Technology Center (CTIM), University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain

<sup>3</sup> University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain

In this way, SmartPort supports 3D visualization of the ports scenery and its data sources.

**Keywords** GIS · Seaport · Smart port · FIWARE · G3M · 3D visualization · Big Data · Georeferenced data

**JEL Classification** O14 (Industrialization, Manufacturing and Service Industries, Choice of Technology) · O33 (Technological Change: Choices and Consequences, Diffusion Processes) · Y10 (Data: Tables and Charts)

## 1 Introduction

A seaport is an environment that combines natural and human elements. This infrastructure is extremely complex, dynamic and of great economic and social importance for coastal areas. Numerous activities of diverse natures take place in it, such as goods and travelers transportation or fishing as well as maintenance operations, rescue and protection of the natural surroundings.

Management of large infrastructures is a task that can be accomplished through collaborative GIS tools taking as example the works of Dragičević and Balram (2004) and Boroushaki and Malczewski (2010). In this case, the port authority is the body entrusted with the decision making within the boundaries of the port, regarding all its activities and available resources. Therefore, it is of vital importance for this organization to have a system that allows the management, visualization and analysis of the elements present at the port just like all the natural ambient factors. Other examples of this kind of platforms can be found in Georgas and Blumberg (2010), regarding the data analysis of New York Harbor and the environmental control of the Gdansk seaport conducted by Kaminski et al. (2009). There are also books in the bibliography about this specific subject such as Claramunt et al. (2007) which lists several related projects and Wright and Yoon (2007) that presents a survey of many data types and their specific treatments.

There are numerous sources of available data within a seaport area. These sources can be subdivided into two groups, depending on their nature. The first group is composed of human-made resources and port activities, including all the human elements within the port. These elements are important for the port authority that has to manage them and for the users of the port that could make use of them. Some are:

- port infrastructure, buildings and maritime signals;
- vessels activities and positions, routes and schedules;
- available piers;
- transported goods and passengers;
- nearby locations of interest: transportations, hospitals;
- possible oil/trash spills;
- emergency management assets;
- road traffic within the seaport.

The second group of sources is related to the natural environment, including both static and dynamic parameters of the surrounding like:

- surrounding topography and model of terrain;
- bathymetric model and composition of the seabed;
- water levels and tides;
- ocean currents;
- water salinity;
- wave movements, direction, frequency and height;
- marine biosphere (presence of fish, cetaceans, seaweed...);
- protected natural spaces;
- meteorological measurements.

The information is gathered from many sources. Personnel and other users will report the human activity, as well as vessels sensors gather it. However, environmental data is recorded in many ways as direct observation, field studies and mainly through sensors located on the ports surroundings.

The amount and variety of the recollected data make necessary the use of Big Data technologies. Big Data is a new concept that involves the storage of massive data and its scalability to its subsequent analysis. This kind of data responds to the comprehensive definition given by the categories volume, velocity and variety. SmartPort fulfills the requirements due to its scalable nature as well as the variety of its data sources. Regarding support systems for Big Data analysis, other works on the area (Zhang et al. 2012) can be of great interest. In particular, the visualization of environmental data extracted of maritime areas has been the target of other studies (McCann 2004) as there are many tools used in the bibliography (Talukder and Panangadan 2009).

Likewise, there are a great number of possible roles for the users of the proposed application, such as sailors, port authorities, emergency management personnel, search and rescue teams, coastal engineers, coastal scientists and oceanographers. For instance, sailors could be interested in tides, currents and maximum ship drafts, while fishermen could need to know zones where fishing is allowed. The changing needs of any particular user make essential that an integral solution should be customizable.

The present work develops a visualization and management system for the data of Las Palmas de Gran Canaria Port. This seaport, also known as “Puerto de la Luz”,<sup>1</sup> is one of the main ports of Spain and the first of the geographical area of West Africa. With more than 16 km. of docks, this port serves as a crossroad between Europe, Africa and America.

The main idea is to display the current state of all the sensors available on the ports surroundings (nowcast) as well as their historical evolution. In addition, the system shows the geographical location of all the resources available to the port authority and the rest of users of the port and the routes of the nearby ships. Finally,

<sup>1</sup> <http://www.palmasport.es/web/guest/autoridad-portuaria>.

SmartPort should support control tasks, through an alert management that monitors the value of many sensors.

The first fundamental pillar of this project is the Future Internet platform of the European Community (FIWARE) architecture, a platform that allows the validation of new concepts, technologies, business models, applications and services in a big scale (FIWARE 2016). The University of Las Palmas de Gran Canaria (ULPGC) as a partner of the FIWARE project has the goal of developing innovative projects using this new technology. This platform belongs to the Program Future Internet Public Private Partnership (FI-PPP), which is a program of public–private cooperation in the field of Future Internet technologies funded by the European Commission involving more than 152 European companies and organizations such as Villaseñor and Estrada (2014).

As maps are gaining an increasingly important role on the web (Kraak 2004), an engine is needed to provide all the functionalities that the user expects from modern web virtual globes. Thus, the other pillar of SmartPort is the G3M virtual globe platform, developed by the University of Las Palmas de Gran Canaria and other Spanish companies. This platform includes a SDK that provides all the drawing functionalities required by our web user interface (Trujillo et al. 2013).

The main goal of this paper is not only to give an overview of the whole SmartPort platform, but to provide a technical paper with novel useful contributions to readers interested in 3D-Monitoring Big Geo Data, like previously done in Suárez et al. (2015). We start with the functional requirements and data acquisition system established by the port authority of Las Palmas seaport. From there, we outline the more relevant features and main modules of what we consider is a seaport management system that can be extrapolated to most seaport cases.

Section 2 of the present document explains the architecture and individual components of the SmartPorts back-end. Section 3 is an introduction to the user Interface and the projects front-end, specially the 3D viewer of the port. Finally, Sect. 4 states the achievements and final remarks of the project as well as some approaches to future work.

## 2 Managing real-time Big Geo Data through FIWARE

Las Palmas seaport is highly related to the port infrastructures of nearby islands. Therefore, it was needed to gather data generated from physical sensors installed in the infrastructure of Gran Canaria, Fuerteventura and Lanzarote seaports. The port sensors are connected to the Internet, a concept known as the Internet of Things (IoT). There are two types of sensors: the sea gauges and the meteorological stations, which generate big volumes of data. The main features of SmartPort are collecting the generated data, analyzing and processing the measurements of the sensors, and visualizing the sensor data in a three-dimensional web-based application.

Apart from visualizing data geographically, a set of requirements was established. Some of those requirements were the visualization of the vessels arrivals and departures and the alert management, which allows to push warnings and notifications anytime a sensor measurement reach a determined value.

As a result of these requirements, SmartPort was developed. It is intended to be a web application, whose back-end is implemented with FIWARE back-end solutions.

The FIWARE platform is based on utility modules, also known as Generic Enablers, that provide solutions of Big Data analysis among others. The available data are recollected and processed using Big Data technologies, such as Hadoop (2016) or Flume (2016). The information is aggregated over time for its use in visualizations offered by the web front-end. The user interface is based on the G3M virtual globe viewer, which displays a 3D representation of the seaport and its surroundings. All the data sources and other elements of importance will be placed within this 3D scene. The use of 3D on a web application makes this front-end a Rich Internet Application (RIA) as some other examples on the bibliography (Sung et al. 2012).

## 2.1 Data context in Las Palmas seaport

Starting with the data captured from the sensors, Las Palmas seaport authority granted us access to its database, where the latest values of the sensors can be read. The sensors average update time is one minute.

Currently, Las Palmas seaport provides the following data from Aanderaa Data Instruments (Aanderaa 2016) Instruments series 3791–3798 water-level sensor and Geonica Datamar 2000C radar 26GHz sea level tide gauge:

- significant wave height spectral moment of order zero [meters];
- average height of the highest third of waves [meters];
- average height of 10% of maximum wave [meters];
- maximum wave height [seconds];
- average period of all waves [seconds];
- peak wave period [seconds];
- average pitch period by the upward zero [seconds];
- mean direction at the peak wave direction [clockwise arc degrees];
- scattering of wave direction at peak power [clockwise arc degrees];
- average direction from which the waves [clockwise arc degrees];
- directionality index [custom indicator];
- pressure of the water column above the sensor [decibars];
- orbital speed about sensor surface [meters per second];
- current direction on surface sensor [clockwise arc degrees];
- energy density spectrum for time series [spectral band].

The seaport also has meteorological sensors such as Geonica 41001 for temperature and air humidity, Geonica 05106 for wind speed and Geonica 52203 for rain gauge.

Once the sensors read the latest values, they are sent to the Orion Context Broker. Orion is an implementation of the Publish/Subscribe (Pub/Sub) Context Broker GE, providing the NGSI9 and NGSI10 interfaces. Pub/Sub, in software architecture, is a messaging pattern where senders of messages, called publishers,

do not program the messages to be sent directly to specific receivers, called subscribers. Using these interfaces, clients can perform several operations:

- registering context producer applications, e.g., a temperature sensor within a room;
- updating context information, e.g., send updates of temperature;
- being notified when changes on context information take place (e.g., the temperature has changed) or with a given frequency, e.g., get the temperature each minute;
- querying context information. The Orion Context Broker stores context information updated from applications, so queries are resolved based on that information.

Several Python (2016) scripts were set up to perform these operations. They take advantage of the Requests library, to simplify the HTTP (Hypertext Transfer Protocol) (HTTP 2016) transactions.

For every sensor that provides information to the SmartPort platform, there is a visual representation on the 3D map viewer.

It is important to mention that, in order to confer the application a dynamic interaction with the data sources, the communication between client and server was developed making use of AJAX (Asynchronous JavaScript And XML) (AJAX 2016) web techniques. Thus, the front-end development was basically done in JavaScript.

Due to certain application components are available as iframes, they have to exchange information between them. For this reason, we have created a programming interface to allow easy messaging exchange between iframes.

In order to find an agile development, instead of making direct requests to the Publish/Subscribe Orion architecture (Orion 2016), an abstraction layer has been created to access the sensors information by “GET requests” to our servers. This intermediate layer enables us to preprocess the data so it can be sent in a friendlier and more compact format. In addition, they also help developers to withdraw from specific Orion aspects.

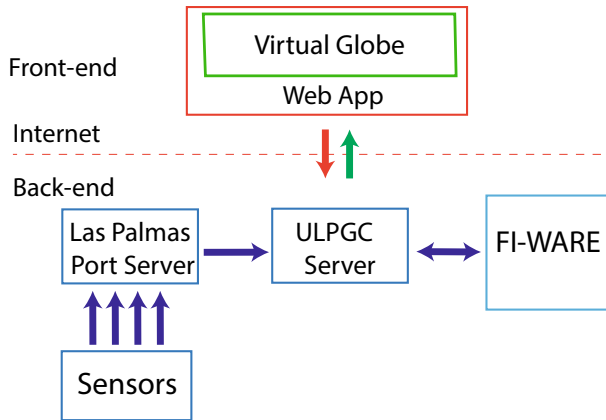
For instance, to get all the vessels data this query is generated:

```
http://<<IP>>:<<Port>>/orion/query.html?sensorType=ship&sensorID=.&pattern=true
```

Another example is to get the meteorological sensors temperature with id 2001:

```
http://<<IP>>:<<Port>>/orion/query.html?sensorType=meteo&sensorID=2001&attributes=[temperature]
```

Another advantage of this new abstraction layer is the potential to take more restricted user control, making use of sessions and avoiding the well-known statelessness of REST (Representational State Transfer) (Costello et al. 2002) APIs.



**Fig. 1** General architecture of SmartPort

The architecture proposed for managing the recollection of the data is shown in Fig. 1. This architecture shows the back-end based on the information given by the sensors, which is processed by the Port Authority, managed by a server located at the ULPGC (University of Las Palmas de G.C.), and then, data are stored to the FIWARE infrastructure where the web application makes directly the requests.

## 2.2 Analysis of Big Data in SmartPort

The next requested functionality to be undertaken was the possibility to display the time evolution of the sensors data and even make an analysis on a given dataset.

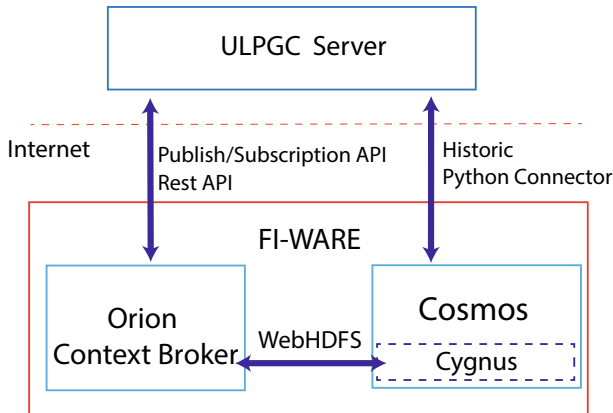
For this purpose, it was decided to make use of the Big Data Analysis Cosmos (Cosmos 2016). Cosmos is an implementation of the Big Data GE, allowing the deployment of private computing clusters based on Hadoop ecosystem. Current version of Cosmos allows users to:

- I/O operations regarding Infinity; a persistent storage cluster based on Hadoop Distributed File System (HDFS) (HDFS 2016);
- creation, usage and deletion of private computing clusters based on MapReduce and SQL-like querying systems such as Hive (2016) or Pig (2016);
- manage the platform, in many aspects such as services, users, clusters, etc, from the Cosmos API or the Cosmos CLI.

There is also a component called Cygnus (Cygnus 2016) in charge of receiving context data from Orion (Context Broker GE implementation) and storing it in a HDFS.

Firstly, Orion has to be connected with Cosmos via Cygnus, which creates the connection with Cosmos and manages the data queue to be stored.

At this point, the historical data have not been yet stored. Orion works with a MongoDB database (MongoDB 2016). For this reason, uploading all the data in



**Fig. 2** Interfaces between the data nodes using FIWARE

**Table 1** Costs of a single record retrieval in microseconds

Time	MySQL	PostgreSQL	Hive
All meteorological data	3.03	4.99	59.51
One attribute sorted by date	1.55	2.53	74.83

chronological order to Orion and relying on Cygnus for sending the data to Cosmos showed a poor performance.

The solution that was implemented consists in adding data to Cosmos using the WebHDFS API.

For instance, a use sample of WebHDFS to add data should be:

```
curl -i -X POST "http://<HOST>:<PORT>/webhdfs/v1/<PATH>
?op=APPEND[&bufferize=<INT>"]"
```

The full procedure of the back-end is shown in Fig. 2.

In the application front-end, for security reasons, the Cosmos instance was only accessible from machines within the same network. For this cause, making requests directly from the client was discarded. Thus, another abstraction layer was necessary.

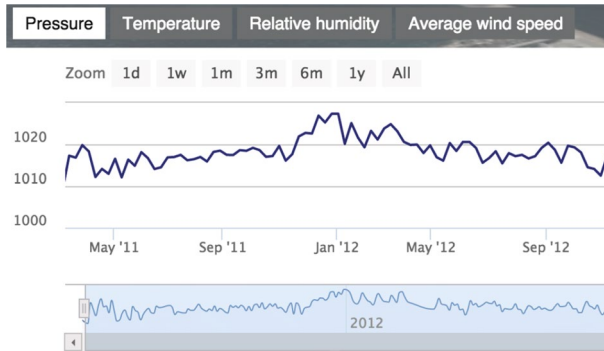
The technologies and techniques used to process and store the data allow to achieve high data availability and ensure that the whole system is scalable. However, some technologies like Hive (2016) or Hadoop imply a performance trade-off, as seen in Table 1. The proposed architecture minimizes the impact of these modules by integrating intermediate layers. This way the data transactions achieve the speed that the interactive user interface demands.

To solve these speed problems, we proceeded to generate a specific Big Data management architecture where we differentiate on three different layers (Marz and Warren 2015):



**Table 2** Costs of a single record retrieval in microseconds using a specific Big Data architecture

Time	MySQL	PostgreSQL	Hive
All meteorological data	3.03	4.99	4.76
One attribute sorted by date	1.55	2.53	5.98

**Fig. 3** Historic chart showing pressure evolution over time

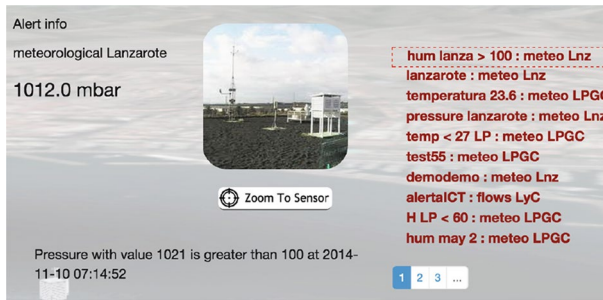
- **Batch Layer:** Where all the data are stored using Hadoop. This layer precalculates views as data can be processed. The post-processing of all data will go to the service layer.
- **Service Layer:** The main role of this layer is to answer queries efficiently; using an RDBMS (Relational Database Manager System) (RDBMS 2016) the query results are stored in this layer.
- **Speed Layer:** One of the major problems to be solved is the processing of data streaming. Thanks to this layer, the latest data may be included in Service Layer while being propagated in the Hadoop distributed system.

Implementing this architecture we got the results shown in Table 2.

For the time-series visualization, there is a wide range of web technologies and components available. It was decided to make use of the Highcharts library, developed by HighSoft (2016), which covered our necessities. Specifically, the component *StockCharts* (StockCharts 2016), shown in Fig. 3, was used since these charts are perfectly adapted to our representation necessities.

A set of intermediate layers has been developed to improve the performance of data analysis and retrieval. This enhancement allows creating on-the-fly sets of charts to show our historical data. These charts are split into tabs showing all the data gathered by a specific sensor. A final business layer implements the logic that decides the sensor to be shown according to messages of other modules.

The requests are made by a Python connector that sends “HIVE requests,” using the `hive_utils` library, in a server with access to the Cosmos global instance. Once this connector is enabled, the requests are analogous to the ones that would be made to a MySQL database. From the front-end point of view, the request is reduced to



**Fig. 4** Notifications and alert center showed when data reach a prescribed margin

ask for a specific sensor data. An example is the following request that asks for all meteorological sensor data with id 2001.

```
http://<<IP>>:<<Port>>/historic.html?sensorType=meteo&sensorID=2001
```

The data are provided in a JSON format (Javascript Object Notation) (JSON 2016). This format is almost ready to be used by the charts present in the front-end.

### 2.3 Enabling data alert notifications

In addition, due to the huge volume of data to be considered, an alert system is a desirable functionality. This system must inform the user that certain values have reached given thresholds. For example, the emergency management system could use these alerts to take quick decisions on reliable and up-to-date data.

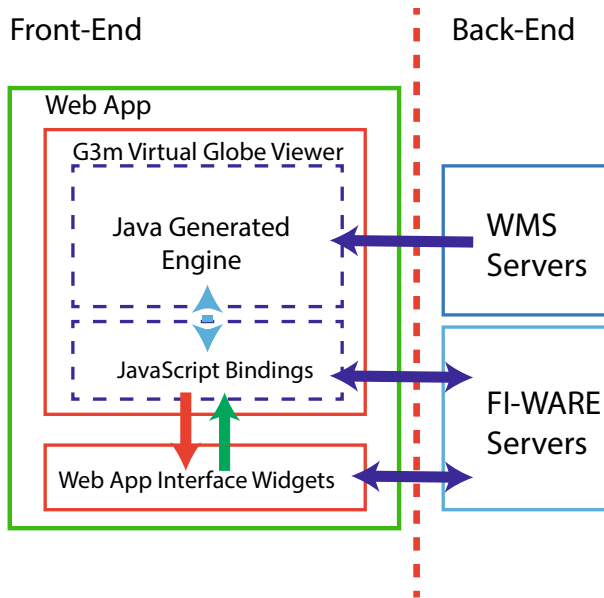
The Orion Application Programming Interface (API) Pub/Sub covers all the functionalities that this system requires. A subscription for each sensor to be watched is created. Thus, whenever an alert is created, it is checked whether a subscription to that sensor attribute already exists and it is made if needed. Then every alert with its activation conditions is stored in a database.

Finally, there is a service that receives the Cosmos subscriptions notifications. The front-end shows a relation of the current alerts and gives the possibility to activate or deactivate them, as shown in Fig. 4.

The next section describes how all these data extracted from the seaport are displayed in 3D. With this purpose, the G3M (2016) map application platform allows us to create an interactive seaport scene, providing a rich and immersive experience to the final user.

## 3 3D visualization of georeferenced data in a Virtual Globe (G3M)

Modern web technologies allow us to develop rich 3D environments to display our georeferenced data. In a resource management system like SmartPort, it is important



**Fig. 5** Main interactions between G3M, the rest of the UI and the back-end of SmartPort

for the final user to spatially locate the existing items and data sources. This makes the application richer in content and much easier for users that already know the port.

Thus, the main element of the user interface is a virtual globe viewer. This element displays a 3D scene that represents the natural environment and the human-made resources of the port area. Within this scene, the application will locate three-dimensional and two-dimensional representations of the elements of interest.

The G3M framework (Pedriza et al. 2012), also known as G3M, was chosen to develop our virtual globe viewer, as it provides all the features demanded by SmartPort.

The web-based version of G3M can be embedded as a web widget based on Google Web Toolkit (GWT) (GWT 2016) in our front-end. This element can be generated from a WebApp project, based on GWT and written in Java. This WebApp allows two main ways of interaction with the developer and the rest of the interface elements.

The G3M API and source code are available in Java. Therefore, the virtual globe is fully configurable from the initialization parameters of the WebApp code, setting different options as the desired imagery, the navigation controls or the initial position of the camera.

It is possible as well to develop all the functionalities required for our viewer in Java and to make it accessible to the browser through JavaScript bindings. This allows dynamic behaviors of our virtual globe controlled by other elements present in the user interface. The interactions between the viewer based on G3M and the rest of the SmartPort modules are shown in Fig. 5.

The main difference of G3M inside the open-source 3D maps community is its multiplatform approach. At this moment, the deployment methodology allows us to add new capabilities to the main engine programming in C++. These changes are directly applied to all the platforms: iOS, Android and Web.

Despite the fact that the current SmartPort user interface is built on Web, developing a native application for Android and iOS is a key step on future developments. Besides, G3M focus on mobile platforms implies a native support for multitouch interaction. Therefore, multitouch interaction is integrated on our web front-end and can be used from devices with capacitive touch screens.

### 3.1 Online terrain imagery for the monitoring of Big Geo Data

From a functional point of view, the G3M engine should provide certain features to support the operational requirements of the SmartPort user interface. One key feature is displaying different layers of imagery over the terrain depending on the particular interest of the user. There are many ways that the georeferenced images could be fetched. In our case, it was decided to utilize the Web Map Service (WMS) protocol developed by the Open Geospatial Consortium community (WMS 2016). WMS has proven to be a reliable source of spatial images on many kinds of applications, including environmental control (Ming-Hsiang 2004). Using this protocol based on HTTP, it is possible to obtain imagery to cover our representation of the globe surface. These images can be synthetic such as street maps, political maps, LiDAR (Laser Imaging Detection and Ranging) (Lidar 2016) visualization or aerial photographs as satellite imagery.

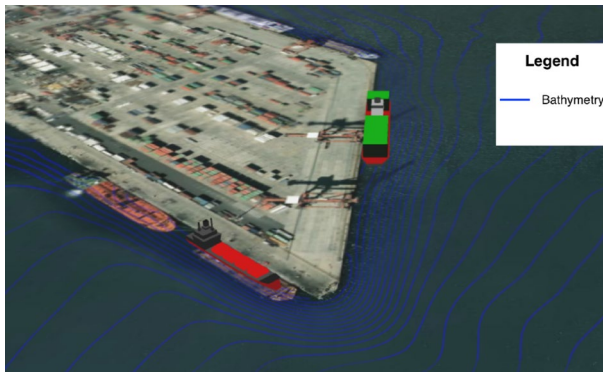
SmartPort viewer uses the Virtual Earth Web Service as base imagery source. This image set is provided by Microsoft and covers the whole planet with high-resolution aerial photographs. However, an application aimed to represent the elements in a port environment requires an even higher image resolution.

The surrounding of the seaport is displayed using images fetched from a local WMS service. In this case, SmartPort uses images from GrafCan, a public company responsible for the maintenance of the geographical data of the Canary Islands. These two layers form the default base image set that defines the appearance of the terrain on the viewer.

In addition to these base layers, the G3M system allows to merge these base images with other semitransparent layers to enrich the information content. A WMS server can provide these added images as well. In this case, the SmartPort allows the user to merge information from bathymetry maps, as is shown in Fig. 6. These maps make easy for the final user to make well-informed decisions concerning routes or maximum draft of ships.

The system also provides by default a layer of natural spaces. These images can be overlaid on the base imagery, showing important natural spaces in the surroundings of the port.

SmartPort allows the final user to fetch different WMS services. The interface provides an easy-to-use dialog in which the user indicates the “URL” of the server and, automatically, the system gives him the whole list of maps.



**Fig. 6** Bathymetry WMS layer loaded



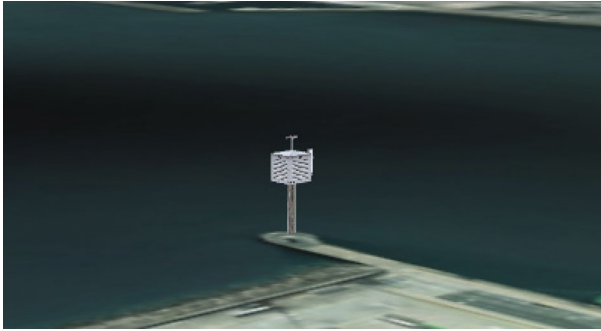
**Fig. 7** 3D view from the port infrastructure using an elevation model

All the selected layers are available to the user as thematic layers. These thematic layers can be merged on the base layers with a selectable degree of transparency.

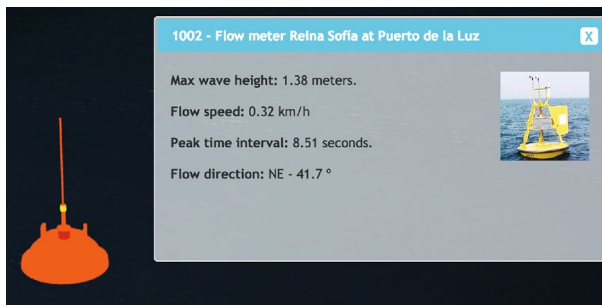
The G3M engine uses a Hierarchical Level Of Detail (HLOD) (Luebke et al. 2003) strategy to show all the terrain tiles with different zoom levels according to the 3D camera position. For every tile, one image request is sent to every imagery source to combine them in a synthetic texture that is mapped on the tile. All this process is automatically carried out by the engine and explained in more detail in Trujillo et al. (2013).

### 3.2 Digital elevation model for 3D earth representation

An elevation model could seem not very relevant on a port control system. However, ports are often surrounded by mountains and other terrain features that make easy for their users to locate themselves and the nearby elements. This effect is remarkable in a mountainous landscape as Gran Canaria. Besides, the inclusion of an



**Fig. 8** Image of a model representing a meteorological station in Puerto de Las Palmas



**Fig. 9** Image of a model representing a buoy and associated information

elevation model makes the viewer easier to use and much more visually attractive, as shown in Fig. 7.

Using data extracted from LiDAR scanning, the terrain features surrounding the seaport are represented in an accurate way. That makes easier for the user to orientate himself within the 3D view and makes it more immersive.

Gran Canaria is a circular-shaped island with a diameter of 45 kilometers approximately. To represent elevations, a regular grid of 1000x1000 values is used. This resolution represents approximately an elevation value every 45 meters, that it is enough for a good visualization.

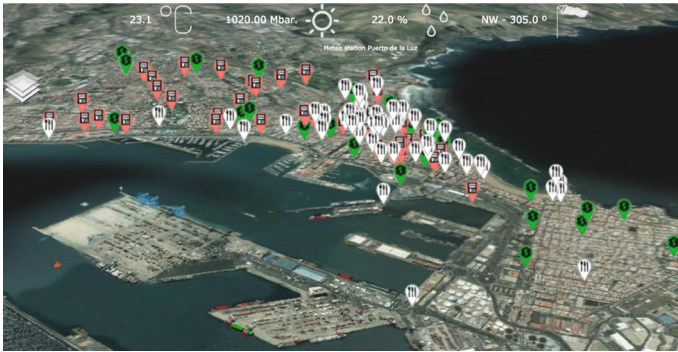
The G3M system allows the use of multiple sources of elevation data at different resolutions. Therefore, the developer can use highly detailed elevations files only on zones of interest and use coarser approximations on other areas. Furthermore, plain areas, such as the sea, can be displayed without the need of loading any elevation data.

### 3.3 Plotting thematic information using map marks and 3D models

In Las Palmas seaport example, five buoys have been located in the vicinity of the port. Every one is represented by its own 3D model. The G3M engine allows the



**Fig. 10** Application uses different icons for different types of point of interest



**Fig. 11** Aerial view showing most of the marks that the application can show. Mark cluttering can be avoided with map generalization techniques such as feature filtering or clustering

developer to use color models, as the model shown in Fig. 8, or textured models and render them using light and shading effects, as shown in Fig. 9.

It is of great importance that all 3D objects of the scene be selectable so it is possible to perform actions and display information as the user interacts with them. In the case of SmartPort for web, this interaction could be through using the mouse device or multitouch screens.

The implementation of this selection feature is based on bounding box hierarchies that allow computing efficiently the intersection problem. Once the object has been selected, a message to the rest of the system is sent. This way a certain action could be performed, as displaying data collected by the selected sensor.

Despite the 3D nature of the G3M environment, sometimes the clearest way to display the location of a particular item within the scene is simply to use a 2D icon, or mark, located on a fixed 3D position. Vectorial information can be easily imported into a GIS system through many standard protocols and languages (Zhong-Ren and Chuanrong 2004). In addition, a 2D mark can represent an element whose model is too complex or just point to a location. Some examples of marks are found in Fig. 10.

One of the functionalities of SmartPort is to show several points of interest that are located near the port. These points of interest are locations that the user may want to find on the map, so there is no physical representation for them. Besides, there are a significant number of marks on the scene, as shown in Fig. 11, which need to be rapidly drawn. G3M implements different techniques to visualize mark sets that exceed the screen space, such as mark clustering, mark filtering, distance culling and dynamic positioning (Santana et al. 2017).



The G3M rendering pipeline allows drawing thousands of marks at a suitable frame rate. It uses a billboarding technique (Wolff 2011) based on the programmable pipeline of mobile GPUs (Cheng and Wang 2011) that avoids the need of maintaining a 3D model for each mark. In addition, the G3M software development kit adds an abstraction layer that prevents the user from dealing with GPU programming (Trujillo et al. 2014). This way, rendering numerous marks does not increase considerably the frame drawing time.

Regarding the generation of icons for the marks, the G3M project provides a multiplatform canvas API (Glob3Mobile 2016) for the creation of these images. This API enables us to generate icons on real time by combining different images or even adding text to them. Finally, G3M supports user interaction with the points of interest marks. This way the system is capable of determining whether the user is touching a given mark. Whenever a mark is touched an alert is created by the system so it performs the corresponding action. In this case, the system shows a web informative dialog, describing the nature of the entity represented by the mark.

SmartPort also uses 3D models to represent important assets within the port area. These entities are the buoys and maritime sensors. The users can easily recognize these elements by their position and shape.

## 4 Discussion and conclusion

The enormous amount of data generated by a regular seaport infrastructure is generally poorly used and often displayed through many channels. Due to this situation, most port authorities obtain a lesser benefit from their quite complex and expensive sensor networks. However, for the first time, it is now possible to process, store and show these Big Data environments so they can offer a truly added value.

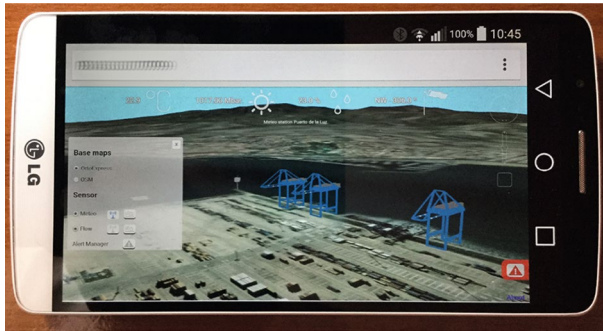
SmartPort reveals itself as an integral tool to assist the analysis and control of all the data sources available for the Las Palmas Port Authority. The analysis offers a geospatial and temporal view of the recollected data, while the control is based on the alert management system. Although there are some other ad hoc solutions to the marine traffic and data analysis in port environments, our work adds a novel 3D experience in which the displayed data respond to the camera movements. Besides, SmartPort seeks to be an open project, offering a scalable and reliable back-end, a mutable data model and a highly customizable web user interface.

At the time this paper is being written, SmartPort has been a successful development and it is about to be delivered to a real environment to play an important role in the decision-making process of the port authority of Las Palmas.

Some of those decisions are:

- to define the location of vessels depending on the environment conditions;
- to use the alert manager to make any needed action when a sensor reaches a defined value;
- to load multiple layers for several purposes such as facilities planning;
- to download the historic sensor data to find specific patterns;
- to know the conditions of the sea currents in the case of oil spill.





**Fig. 12** SmartPort running on an Android mobile phone

At the domain study stage, we have fully examined the seaports infrastructure to obtain a data model useful for the management of the port resources. The biggest challenge, however, was to offer a reliable, scalable and fast data treatment to support the user interface. In this sense, the Generic Enablers provided by FIWARE offered us all the Big Data needed resources.

On the other hand, G3M was a key element in the development of the front-end. Using the latest version of WebGL allows us to produce a 3D user experience that was at the same time, interactive and customizable. This way, the user is able to display the data he is really interested into and navigate through a representation of Las Palmas seaport. In a similar manner, the rest of the interface elements have been thought to offer the best user experience in terms of intuitive use and fast data visualization.

The main line of future work is the migration of the SmartPort to truly be supported on mobile handheld devices as smartphones and tablets, where the G3M virtual globe has been proved and tested so far. Some other upcoming tasks are to keep improving the data transfer rates and usability of the user interface. Finally, some other features could be added based on the current available data, such as forecasting and automatic control systems.

Currently, we have made some tests of the Web Application with mobile devices, as we show in Fig. 12. This allows to the port staff to access the sensor data without the need of a computer. In these tests, we have noticed the following:

- Performance using SmartPort in mobile browsers is significantly slower compared to using it on a desktop browser.
- Gestures are highly dependent on the browser support of the Touch Event API, which can lead to interaction problems.

As a conclusion for the development focused on mobile devices support, we would develop using G3M natively (for iOS and Android). In this way, the native implementations are much more efficient and compatible with all the touch gestures. Once the app can be executed natively in these systems, its functionality

will be improved; for instance, staff could take their smart devices to the port and access on the go all the information and make on-site decisions.

**Acknowledgements** This work has been supported by the Spanish Ministry of Economy and Competitiveness (MINECO) project RTC-2014-2258-8 and by the European Commission FP7 project “FIWARE: Future Internet Core Platform” FP7-2011-ICT-FI 285248. We would like to thank the port authority general director, Salvador Capella, for supporting the access to the data of Las Palmas seaport. The fifth author wants to thank Agencia Canaria de Investigación, Innovación y Sociedad de la Información, for the grant “Formación del Personal Investigador-2012 of Gobierno de Canarias”.

## References

- Aanderaa (2016) Aanderaa web page. <http://www.aanderaa.com>. Accessed 5 Aug 2016
- AJAX (2016) Ajax web page. <https://api.jquery.com/category/ajax/>. Accessed 5 Aug 2016
- Borouhshaki S, Malczewski J (2010) ParticipatoryGIS: a web-based collaborative GIS and multicriteria decision analysis. *Urisa J* 22(1):23
- Cheng KT, Wang YC (2011) Using mobile gpu for general-purpose computing—a case study of face recognition on smartphones. In: 2011 international symposium on VLSI design, automation and test (VLSI-DAT). IEEE, pp 1–4
- Claramunt C, Devogele T, Fournier S, Noyon V, Petit M, Ray C (2007) Maritime GIS: from monitoring to simulation systems. In: Information fusion and geographic information systems. Springer, pp 34–44
- Cosmos (2016) Cosmos web page. <http://catalogue.fiware.org/enablers/bigdata-analysis-cosmos>. Accessed 5 Aug 2016
- Costello RL et al (2002) Rest (representational state transfer). Last updated Jun 26
- Cygnus (2016) Cygnus web page. <https://github.com/telefonicaid/ifiware-cygnus>. Accessed 5 Aug 2016
- Dragičević S, Balram S (2004) A web GIS collaborative framework to structure and manage distributed planning processes. *J Geogr Syst* 6(2):133–153
- FIWARE (2016) Puerto de la luz web page. <http://www.fiware.org>. Accessed 5 Aug 2016
- Flume (2016) Flume web page. <http://www.http://hadoop.apache.org>. Accessed 5 Aug 2016
- G3M (2016) G3M web page. <http://glob3mobile.com>. Accessed 5 Aug 2016
- Georgas N, Blumberg A (2010) Establishing confidence in marine forecast systems: the design and skill assessment of the New York Harbor observation and prediction system, version 3 (NYHOPS v3). In: 11th international conference in estuarine and coastal modeling (ECM11). Spalding, ML, Ph.D., PE, American Society of Civil Engineers, Washington, pp 660–685
- Glob3Mobile (2016) G3M documentation web page. <https://github.com/glob3mobile/g3m>. Accessed 5 Aug 2016
- GWT (2016) GWT web page. <http://www.gwtproject.org>. Accessed 5 Aug 2016
- Hadoop (2016) Hadoop web page. <http://www.http://hadoop.apache.org>. Accessed 5 Aug 2016
- HDFS (2016) HDFS web page. [https://hadoop.apache.org/docs/r1.2.1/hdfs\\_design.html](https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html). Accessed 5 Aug 2016
- HighSoft (2016) HighSoft web page. <http://www.highcharts.com/about>. Accessed 5 Aug 2016
- Hive (2016) Hive web page. <https://hive.apache.org>. Accessed 5 Aug 2016
- HTTP (2016) HTTP web page. <http://www.w3.org/Protocols/>. Accessed 5 Aug 2016
- JSON (2016) JSON web page. <http://www.json.org>. Accessed 5 Aug 2016
- Kaminski L, Kulawiak M, Cizmowski W, Chybicki A, Stepnowski A, Orlowski A (2009) Web-based GIS dedicated for marine environment surveillance and monitoring. In: OCEANS 2009-EUROPE. IEEE, pp 1–7
- Kraak MJ (2004) The role of the map in a web-GIS environment. *J Geogr Syst* 6(2):83–93
- Lidar (2016) Lidar wikipedia web page. <https://en.wikipedia.org/wiki/Lidar>. Accessed 5 Aug 2016
- Luebke DP, Reddy M, Cohen JD, Varshney A, Watson B, Huebner R (2003) Level of detail for 3D graphics. Morgan Kaufmann, Boston, MA
- Marz N, Warren J (2015) Big Data: principles and best practices of scalable realtime data systems. Manning Publication, New York

- McCann M (2004) Using geovml for 3D oceanographic data visualizations. In: Proceedings of the ninth international conference on 3D Web technology. ACM, pp 15–21
- Ming-Hsiang T (2004) Integrating web-based GIS and image processing tools for environmental monitoring and natural resource management. *J Geogr Syst* 6(2):155–174
- MongoDB (2016) MongoDB web page. <https://www.mongodb.com>. Accessed 5 Aug 2016
- Orion (2016) Orion web page. <http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker>. Accessed 5 Aug 2016
- Pedriz A, Citores M, de la Calle M, Gómez-Deck D, Trujillo A, Santana JM, Perdomo K, Suárez JP (2012) Glob3 mobile: Sistemas de información geográficos 3d en entornos de movilidad. In: III Jornadas Ibéricas de Infraestructura de Datos Espaciales, Madrid, Spain
- Pig (2016) Pig web page. <https://pig.apache.org>. Accessed 5 Aug 2016
- Python (2016) Python web page. <https://www.python.org>. Accessed 5 Aug 2016
- RDBMS (2016) Relational database manager system wikipedia web page. [https://en.wikipedia.org/wiki/Relational\\_database\\_management\\_system](https://en.wikipedia.org/wiki/Relational_database_management_system). Accessed 5 Aug 2016
- Santana JM, Trujillo A, Suárez JP (2018) A physical model for screen space distribution of 3D marks on geographical information systems. *Math Comput Simul* 147:264–278. <https://doi.org/10.1016/j.matcom.2017.06.003>
- StockCharts (2016) StockCharts web page. <http://stockcharts.com>. Accessed 5 Aug 2016
- Suárez JP, Trujillo A, Dominguez C, Santana JM, Fernández P (2015) Managing and 3D visualization of real-time big geo-referenced data from Las Palmas port through a flexible open source computer architecture. In: 1st international conference on geographical information systems theory, applications and management. Scitepress, Barcelona
- Sung AhK, Dongyoun S, Yoon C, Thomas S, Steffen PW (2012) Integrated energy monitoring and visualization system for smart green city development: designing a spatial information integrated energy monitoring model in the context of massive data management on a web based platform. *Autom Constr* 22:51–59
- Talukder A, Panangadan A (2009) Online visualization of adaptive distributed sensor webs. In: Aerospace conference, 2009 IEEE. IEEE, pp 1–8
- Trujillo A, Gómez-Deck D, de la Calle M, Suárez JP, Pedriz A, Santana JM (2013) Glob3 mobile: an open source framework for designing virtual globes on ios and android mobile devices. In: Progress and new trends in 3D geoinformation sciences. Springer, pp 211–229
- Trujillo A, Suárez JP, Santana JM, de la Calle M, Gómez-Deck D (2014) An efficient architecture for automatic shaders management on virtual globes. In: 2014 fifth international conference on computing for geospatial research and application (COM. Geo). IEEE, pp 38–42
- Villaseñor E, Estrada H (2014) Informetric mapping of big data in fi-ware. In: Proceedings of the 15th annual international conference on digital government research. ACM, pp 348–349
- WMS (2016) WMS specification web page. <http://www.opengeospatial.org/standards/wms>. Accessed 5 Aug 2016
- Wolff D (2011) OpenGL 4.0 shading language cookbook. Packt Publishing Ltd., Birmingham
- Wright NR, Yoon J (2007) Application of GIS technologies in port facilities and operations management. ASCE Publications, Reston
- Zhang L, Stoffel A, Behrisch M, Mittelstadt S, Schreck T, Pompl R, Weber S, Last H, Keim D (2012) Visual analytics for the big data era—a comparative review of state-of-the-art commercial systems. In: 2012 IEEE conference on visual analytics science and technology (VAST). IEEE, pp 173–182
- Zhong-Ren P, Chuanrong Z (2004) The roles of geography markup language (GML), scalable vector graphics (SVG), and web feature service (WFS) specifications in the development of internet geographic information systems (GIS). *J Geogr Syst* 6(2):95–116

Reproduced with permission of copyright owner.  
Further reproduction prohibited without permission.