Multilayer Big Data Architecture for Remote Sensing in Eolic Parks

Enrique Moguel, Juan C. Preciado, Fernando Sánchez-Figueroa, Miguel A. Preciado, and Juan Hernández, *Member, IEEE*

***Abstract*—Due to their nature, Eolic parks are situated in zones with difficult access. As a result, management of Eolic parks using remote sensing techniques is of great importance. In addition, the huge amount of data managed by Eolic parks, together with their nature (distributed, heterogeneous, produced, consumed at different times, etc.) makes them ideal to apply big data techniques. In this paper, we present a multilayer hardware/software architecture that applies cloud computing techniques for managing big data from Eolic parks. This architecture allows tackling the processing of large, distributed, and heterogeneous data sets in a remote sensing context. An innovative contribution of this work is the combination of different techniques at three different layers of the proposed hardware/software architecture for Eolic park big data management and processing.**

***Index Terms*—Big data, cloud computing, Eolic parks, remote sensing, wind turbine.**

I. Introduction

During the last years, research in renewable energies and smart grids has been gaining more and more popularity not only in companies but also in governments and agencies. Among these energies, the Eolic one has attracted a lot of interest, as it currently generates more than 5% of the world total power consumption (284 GW). It is expected that, no later than 2020, the Eolic energy will supply nearly 20% [1] of the world demand.

An Eolic park is formed by one or more electric substations. Each substation is formed by a set of wind turbines. Substations can now have up to 150 wind turbines, and there are companies managing hundreds of Eolic parks, thus having to manage the information coming from more than 10 000 wind turbines. As a result, an important challenge in Eolic parks is how to monitor and control the behavior and security of wind turbines. Attending to the number of wind turbines and the amount of data generated in each one, the monitoring system has to deal with a flow of data of more than 130 Gb/min. These data are generated at different levels: wind turbine, substation, and control center.

Due to their nature, Eolic parks are commonly situated in rural zones or in the open sea, which impose some problems in terms of accessibility. In this context, data acquisition and management (which is often subject to some kind of high performance computing due to the massive volumes of data involved) is generally performed by means of remote sensing techniques [2], [3]. This fact, together with the great volume of data to be processed, the distributed nature of such data, the heterogeneity of the data sources, and the different times of data production (synchronous and asynchronous), together with the different data structures involved (some structured but most of them not structured), and the fact that each node of the system can act as both producer and consumer, make Eolic parks an ideal scenario for the application of big data techniques for remote sensing [4]. According to the definition provided in [5], mainly three aspects characterize big data: 1) the data are numerous; 2) the data cannot be categorized into regular relational databases; and 3) data are generated, captured, and processed rapidly. The data collected from Eolic parks conform to these characteristics; hence the application of big remote sensing data processing techniques can provide an important asset for the management of Eolic parks as a relevant source of renewable energy.

Most available solutions for controlling and monitoring Eolic parks are mainly based on hardware procedures, i.e., by means of a combination between sensors and programmable logic controllers (PLCs), in which software techniques have played a secondary role. However, the combination of research results coming from different fields such as big data [5], web engineering [6], data visualization [7], and cloud computing [8], [9] can lead to obtaining better solutions from different points of view: performance, scalability, economy, maintenance, user experience, etc. In this paper, we develop a new multilayer software/hardware architecture for controlling and monitoring Eolic parks. This architecture which is applied to a case study allows tackling the processing of large, distributed and heterogeneous (regarding its source, formats, and production velocity) data sets in a remote sensing context. In the presented case study, we define three different levels for data management: wind turbine, substation, and control center. An important contribution (from a big data processing perspective) is given by different strategies that are specifically adopted in each layer. While the wind turbine level makes use of low cost devices for optimizing data management and transfer, the control center layer makes use of advanced features such as

Manuscript received December 01, 2014; revised February 12, 2015; accepted March 13, 2015. Date of publication April 05, 2015; date of current version January 08, 2016. This work was supported in part by the Ministerio de Ciencia e Innovacion (Spanish Contract MIGRARIA TIN2011-27340), in part by the European Regional Development Fund (ERDF), and in part by the Gobierno de Extremadura (GR10129).

E. Moguel, J. C. Preciado, F. Sánchez-Figueroa, and J. Hernández are with the Department of Telematic and Informatic Systems Engineering, Universidad de Extremadura, Cáceres 10003, Spain (e-mail: enrique@unex.es; jcpreciado@unex.es; fernando@unex.es; juanher@unex.es).

M. A. Preciado is with Homeria Open Solutions, S.L., Edificio Tajo de Gestión del Conocimiento, Universidad de Extremadura, Cáceres 10003, Spain (e-mail: mapreciado@homeria.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2015.2415583

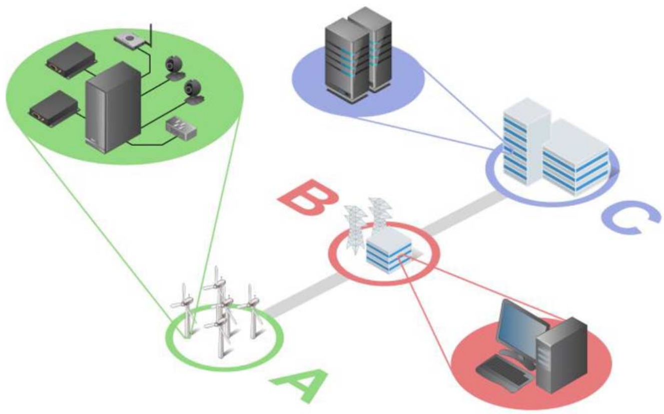


Fig. 1. Multilayer architecture adopted by solutions currently being used for Eolic park management.

cloud computing provided by the infrastructure of Amazon Web services (AWSs).

This paper is organized as follows. Section II establishes the requirements of the system. Section III describes some related work in the literature and identifies the main problems and the solutions currently being used in Eolic parks. Section IV presents the proposed architecture, which is split into three different layers. Section V describes the implementation of the system and provides some information about its experimental validation. Finally, Section VI concludes the paper with some remarks and hints at plausible future research lines.

II. Requirements of the System

In this section, we describe the main requirements for Eolic park management. Fig. 1 shows the multilayer architecture currently used in most developments. As shown in Fig. 1, the architecture can be decomposed into three different layers or levels: A) wind turbine layer; B) substation layer; and C) control center layer. The first layer (A) is mainly in charge of gathering data from sensors (such as cameras, volumetric sensors, etc.) and also from managing actuators (such as alarms and lights) and sending the data to the next layers. The second (B) and third (C) layers are in charge of storing logs, processing tasks for data analytics, applying automatic rules for triggering actuators, applying visualization techniques for the obtained operators, among several others. The volume of data generated each day in Eolic parks is high (near to 15 Mb/min in each wind turbine), and the data are coming from multiple sources (sensors) with diverse formats, different storage needs, and with different production and consumption rhythms. In addition to this dynamically generated data, there are other static data sets that do not change over time (e.g., wind turbine geographical location/position, name, etc.)

In the following, we summarize the main requirements of systems with the architecture described in Fig. 1.

1) *Data production and consumption:* The three layers act both as data producers and consumers. The data produced can be either stored or transmitted to be consumed by a different layer. A consumed data can be stored, processed for triggering an action, or visualized for operators.
2) *Data processing:* Each layer must incorporate data processing capabilities in order to filter data or transform it into useful information.
3) *Data storage:* An important requirement is that data must be stored in the three levels as presented in Fig. 1. Levels B and C generally store logs and analytical/operational tasks, while level A stores the data necessary for working in offline mode when connection problems arise.
4) *Data transmission:* Another important requirement of the system architecture described in Fig. 1 is that the data must flow bidirectionally among the three existing levels.
5) *Scalability.* Finally, in terms of scalability, the incorporation of new wind turbines in substations or new substations in the system is important requirement in order to guarantee the full incorporation of the system in new Eolic parks.

Once the main requirements have been presented, next we show some related works, from both, an academy point of view and industrial point of view.

III. Related Work

During the last years, big data techniques have been rapidly adopted in different areas. However, in the industrial environment, big data techniques have not been adopted so quickly [10]. In [11], big data techniques are used for analyzing data in industrial processes, but with a limited scope for remote sensing. In [12], a big data architecture is proposed for federated sensor services, but without an implementation. More related to energy production, in [13], big data and stream data mining techniques are used for accurately predicting the energy production from renewable sources. This work is more focused on prediction than monitoring and is not based on a multilayer architecture.

With no doubt, those related works which are closer to our contribution come from the field of smart cities, where a similar data ecosystem to Eolic parks can be found (in terms of data heterogeneity, data distribution, different production, and consumption rhythms, etc). For example, in [14], an architecture for cloud-based big data analytics for smart future cities is proposed. However, an implementation is not provided in this contribution. The work in [15] tries to determine human dynamics by analyzing the ecosystem defined through the triangle formed by big data, smart cities, and wearable computing. Here the focus is on gathering data from different and heterogeneous distributed sensors, but without providing a multilayer architecture.

From an industrial point of view and focusing on level A in Fig. 1, a common solution to address the aforementioned requirements has been the use of PLCs [16]. This represents a hardware solution in which a computer is used in industrial environments for automation tasks in electromechanical processes. However, the solution based on PLCs does not completely address the requirements as Table I shows. Specifically, the first three rows of Table I summarize the main problems with the existing solution (based on PLCs) regarding the requirements established above. In the PLC solution, each wind turbine has sensors connected to the PLC and data are sent from

TABLE I
MAIN PROBLEMS OF THE EXISTING (PLC-BASED) SOLUTION AND DESIRED PERFORMANCE AFTER SOLVING THE EXISTING PROBLEMS

		Data production	*Consumption*	*Data processing*	*Data storage (off line)*	*Data transmission (bidirectionally)*	*Scalability*
Current	Layer A	✓	⊘	⊘	⊘	⊘	Low
	Layer B	✓	✓	✓	✓	⊘	Low
	Layer C	✓	✓	(!)	(!)	⊘	Low
Desired	Layer A	✓	✓	✓	✓	✓	High
	Layer B	✓	✓	✓	✓	✓	High
	Layer C	✓	✓	✓	✓	✓	High

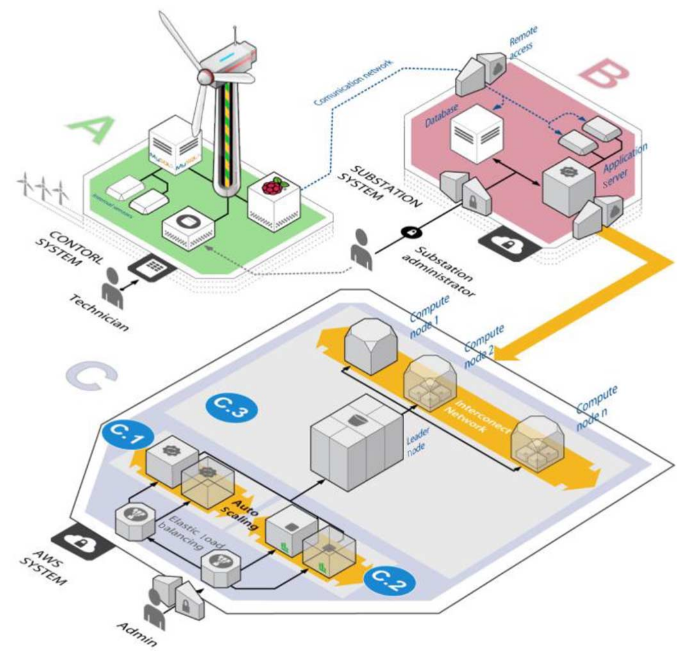


Fig. 2. Proposed multilayer big data architecture for Eolic park management.

level A to level B (not vice versa) without being filtered or processed, resulting in a huge amount of data to be transmitted. This means that, if there is no connection at that time, the data are lost (in addition, no storing capabilities are provided).

As shown in Table I, the main problems of the PLC-based solution are located in levels A and C. Concerning level A, the main drawbacks are related with the fact that there are neither consumption, nor processing or storing capabilities, and also to the fact that the transmission mode is only unidirectional. This means that most of the actions have to be taken by operators and forced to deploy a powerful transmission channel ($+10\,\mathrm{MB/s}$). Regarding level C, the main drawbacks are the data storage and processing capabilities due to the continuous re-dimensioning of servers and database infrastructure. Finally, the scalability of the overall system is low, while the cost is quite high.

IV. PROPOSED MULTILAYER BIG DATA ARCHITECTURE

This section describes the additions performed to the classic architecture in Fig. 1 in order to cope with the requirements of modern processing scenarios in Eolic parks, i.e., obtaining the desired performance highlighted in rows 4–6 of Table I. Specifically, we describe for each layer the hardware and data management-oriented modifications conducted. Fig. 2 illustrates the main parts of the proposed system, which will be explained separately.

A. Wind Turbine Layer

This layer is in charge of orchestrating all sensors and actuators by means of a rule-based system. The enhancements conducted to this layer can be summarized as follows. From the hardware perspective, the main modification is that the PLC has been substituted by a *Raspberry Pi* [17], [18] which is connected to both sensors (such as cameras, volumetric sensors, smoke detectors, etc.) and actuators (such as alarms, lights, etc.) This brings several important benefits:

1) The *Raspberry Pi* has processing capabilities and can be easily programmed.
2) The *Raspberry Pi* can provide persistent storage using a built-in MySQL database that allows saving the data when the connection links are not available, thus allowing an offline working mode.
3) By means of the *Raspberry Pi*, the communication is now bidirectional. Although the *Raspberry Pi* can use WiFi, Wimax, or 3G networking capabilities, we use a simple Ethernet connection due to the long distances among wind turbines and also to avoid disturbances from the spinning turbines. This means that the *Raspberry Pi* can act both as a producer as well as a consumer. This allows us to fully exploit the potential of the *Raspberry Pi* as a PLC, by connecting it to appropriate interfaces for input/output (I/O).

It should be noted that the *Raspberry Pi* used in our configuration has an ARM 700 processor at 700 MHz and RAM memory of 512 MB. Regarding its connectivity, the *Raspberry Pi* has two USB ports and HDMI. Its computing capability is roughly that of a Pentium III processor but with one-tenth of its electrical consumption (5 W vs. 50 W). The *Raspberry Pi* chosen uses a 64-GB SD card for local storage. This reduces the number of write operations to the same area of the card, extending its life. The operating system (Raspbian OS Linux) is located on this SD card.

The *Raspberry Pi* also has the capacity to transmit data to the server located at the substation layer. This is done through the MAC address of the *Raspberry Pi* (in the case of video transmission, this is continuously sent to a fixed directory at the substation layer). At this layer, the activation of actuators is performed using the HTTP POST protocol. All the sensors and actuators are connected to the Raspberry Pi thanks to the Arduino shield [19]. The *Raspberry Pi* is connected to a router that provides the Internet connection. When the *Raspberry Pi* loses the connection with the substation layer, it stores the local data produced coming from its sensors and the timestamp for each one to be resent later. Our proposed combination of a software and hardware approach for the wind turbine layer favors the scalability of the system and reduces its cost, being 60% less than the same layer in the previous design reported in Fig. 1. This reduction mainly comes from two facts: 1) the price of the *Raspberry Pi* is much lower than the price of the PLC; and 2) the installation of the *Raspberry Pi* is easier (*plug and play*) and quicker than that of the PLC.

From a data management perspective, it should be noted that the wind turbine layer is the level that produces the most significant amount of data, mainly due to the high number of data producers located in this layer. Specifically, each wind turbine produces 14 MB of data per minute (which amounts to near to 20 GB of data per day). This massive volume results from a video frame configuration of 320×240 pixels with 15 frames/s. The rest of the sensors produce about 1 Mb of data per day. In this regard, the preprocessing logic that we have implemented in the *Raspberry Pi* saves nearly 80% of data to be communicated to the subsequent level (substation layer). This information can be simply stored on the SD card in case of communication problems. In the following section, we describe the modifications conducted to the substation layer.

B. Substation Layer

This level is in charge of receiving data from the associated wind turbines and processing it for storage and visualization purposes. The level is also in charge of performing actuations automatically based on rules, or transferring data to the control center layer. In the following, we describe the improvements obtained in this layer, which come mainly as a result of the modifications conducted to the wind turbine layer.

From a data management perspective, thanks to the preprocessing performed in the wind turbine layer, here we can receive nearly 600 GB/day. The processing performed at this level allows both controlling the different wind turbines belonging to a specific substation in an autonomous way, and reducing the amount of data to be transferred to the control center layer (105 GB/day approximately).

From a hardware dimension perspective, the servers installed in this layer are based on the Intel Xeon E5-2600 processor. These servers dramatically boost application performance and offer highly efficient memory management. Specifically, the large storage capacity (30 TB) allows recording a large amount of video data and reports. The hardware has been selected in order to facilitate the scalability of the system and its portability to new stations.

C. Control Center Layer

This layer is in charge of receiving data from the substation layer and processing/storing it or performing tasks related to analytics, visualization, and automatic actuation based on rules. In the following, we describe the modifications performed to this layer from a data management and also from a hardware perspective.

From the data management point of view, the control center manages more than 65 substations simultaneously, which means that it receives information from all of them almost constantly. Specifically, the control center receives a data flow of 75 Mb/min from each substation, which can reach a data flow of 4.76 GB/min (6.69 TB/day) from all the available wind turbines that manages. This large volume of data requires a cutting-edge infrastructure that facilitates massive data storage and processing.

From a hardware point of view, a main contribution of our design is that we completely replace the hardware infrastructure of the system in Fig. 1 by cloud services provided by the AWS infrastructure[1] to store and manage the resources. To perform big data analytics at this level with an increasing volume of data, production velocity and variety of information, cloud computing services are required, allowing us for the automatic management of resources to meet demand. This represents a significant contribution as we virtualize the required hardware resources in the cloud while taking advantage of the advanced computing infrastructure provided by cloud resources. Specifically, the modifications conducted to layer C in Fig. 1 (attending to the different services provided) can be summarized by the following contributions.

1) *Data analytics (sublayer C.1)*: This sublayer has been designed to perform all the data processing operations required by the system. It is based on a high performance computing environment [9] that allows for the management of a large amount of data. It provides autoscaling capacities seamlessly to the spike of demands, maintaining the overall performance of the system. This complex computational workload is managed using parallel processes that also increase significantly the computational throughput and the availability of resources. Specifically, all these virtual instances are run within a placement group that provides low latency between instances and databases.
2) *Control services (sublayer C.2)*: This sublayer is provided to manage and perform the visualization and interaction services, which allow us to review all the data processed by the data analytics section. It is based on the so-called elastic web scalable computing nodes. Autoscaling features are also provided in this section. The load balancing nodes automatically distribute the incoming traffic across the virtual instances. The proposed design enables us to improve fault tolerance and performance in general. All the information, reports and data collected in the subjacent layers and already processed in the data analytics section can be managed here.
3) *Massive data storage (sublayer C.3)*: This sublayer is mainly in charge of the data warehousing procedures needed for analytics. It is implemented by a commodity cluster-based solution that delivers fast query performance technology to improve I/O efficiency and parallelize queries across multiple nodes. We have also

[1] [Online]. Available: http://aws.amazon.com

implemented a variety of innovations to obtain very high query performance on datasets ranging from gigabytes to petabytes in size. Specifically, we use columnar storage, data compression, and zone maps to reduce the amount of I/O needed to perform queries. The cluster used for parallelizing the queries is composed of several nodes. A master node coordinates the compute (worker) nodes and handles external communications. The data analytics (C.1) and control services (C.2) sublayers interact directly only with the leader node. The compute nodes are transparent to external applications. The master node parses and develops execution plans to carry out database operations such as the series of steps needed to obtain results for complex queries. Based on the execution plan, the master node compiles code, distributes the compiled code to the workers, and assigns a portion of the data to each compute node. Each worker then executes the compiled code and sends intermediate results back to the master node for final aggregation. Each compute node has its own dedicated CPU, memory, and attached disk storage.

V. System Implementation and Experimental Validation

The software system developed for the *Raspberry Pi* at the wind turbine layer (A) has been developed using Python [20]. The database used at this level is MySQL,[2] which is enough for the volume of data to be processed. The operational logic and visualization systems used in the substation layer (B) and control center layer (C) have been developed using model-driven engineering [21] principles. To be more precise, the WebRatio [22] tool has been used. The code of the deployed applications is based on J2EE (a platform-independent, Java-centric environment for developing, building, and deploying web-based enterprise applications online) running on Apache servers. The interface of the applications is based on HTML5.

Fig. 3 shows a screenshot of the state of a particular wind turbine. The software interface for wind turbines (layer A) allows their management from the substation layer or the control center layer. Using this interface, operators are able to check the state of each wind turbine and interact with them in a straightforward manner. In addition, operators can view the video streams from the cameras located on each wind turbine in real time. Fig. 4 shows a screenshot of the state of a particular substation. A color code has been used to indicate the current state of the controller (red: disarmed; green: armed; orange: isolated; gray: offline; blue: lights on; purple: alarm on).

A prototype of the system was tested in an Eolic park with 756 wind turbines belonging to six different substations. From a data management point of view, we obtained a significant improvement. First, we obtained a significant reduction of data traffic (more than 65%) and an increase on data processing speed (about 40%) compared to the current solution being used. Secondly, the new architecture allowed us to store, process, and consume data at the wind turbine level, granting bidirectionality through the different levels. These features were the different requirements shown in Table I. Additionally, a great degree of

[2][Online]. Available: http://www.mysql.com/

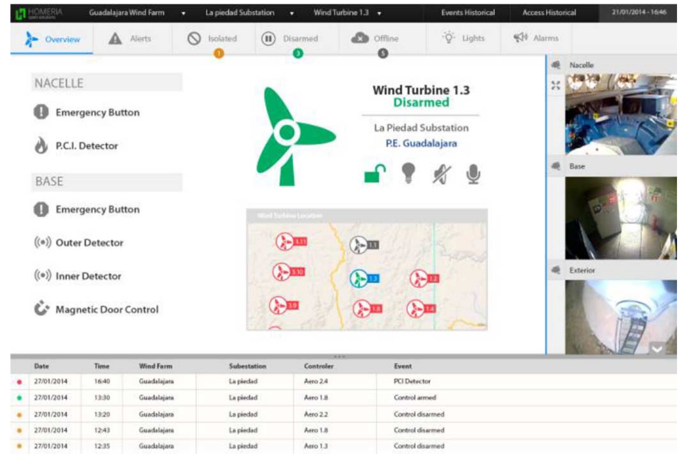


Fig. 3. Software interface: state of a particular wind turbine.

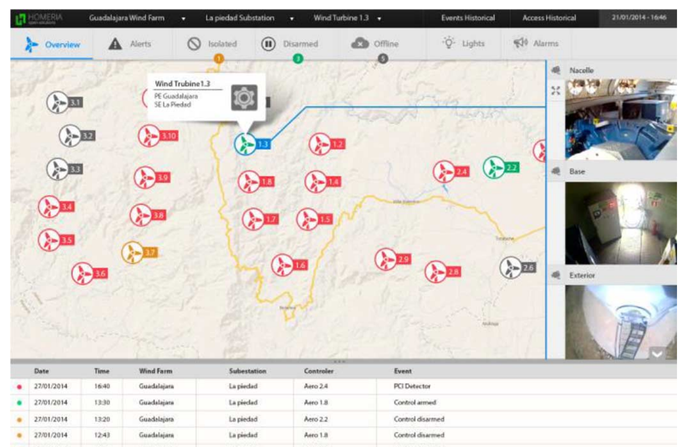


Fig. 4. Software interface: state of a particular substation.

scalability was obtained since the elastic infrastructure used at level B and C dynamically adapted to the growth of processing and storing necessities. Last but not least, from an economic point of view, we obtained a significant reduction at the cost of level A (from 900$ per wind turbine to just 130$).

VI. Conclusion and Future Lines

This paper has described a new system based on big data strategies to manage and process the huge amount of data generated in Eolic parks. The system approaches the problem from a remote sensing perspective, and has been designed in order to cope with the specific characteristics of the data generated in this context (apart from its huge volume, the data are distributed, heterogeneous, asynchronous, and bidirectional, which complicates the design of the system). The proposed solution is based on the use of three different levels where different techniques apply. The first level (wind turbine layer) incorporates preprocessing capabilities by using low cost devices such as the *Raspberry Pi*. The solution adopted allows saving more than 80% of the cost of the previous solution, based on a specialized hardware (PLC). The second level (substation layer) is implemented using traditional database management with a local server. Finally, the third level (control center layer) is based on the use of cloud computing techniques which allows us to manage data storage and processing, together with data analytics and visualization tasks. This multilayer architecture has been tested with success in one of the most important electric companies in the world with a great presence in the renewable energies sector.

Our future work will be focused on two main directions. On one hand, we will improve the system by including the use of drones for gathering external information. On the other hand, we are also planning on applying the results obtained from our tests to other Eolic parks of the same company, with the aim of demonstrating the scalability of our system.

ACKNOWLEDGMENT

The authors would like to thank the guest editors and the anonymous reviewers for their valuable comments and suggestions to improve the paper.

REFERENCES

[1] E. F. Camacho, T. Samad, M. García-Sanz, and I. Hiskens, "Control for renewable energy and smart grids," in *The Impact of Control Technology*, T. Samad and A. M. Annaswamy, Eds. IEEE Control Systems Society, vol. 31, no. 5, pp. 26–27, 2011.

[2] A. Plaza, Q. Du, Y.-L. Chang, and R. L. King, "Foreword to the special issue on high performance computing in earth observation and remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 503–507, Sep. 2011.

[3] C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B. Huang, "Recent developments in high performance computing for remote sensing: A review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 508–527, Sep. 2011.

[4] A. Ferran, S. Bernabe, P. G. Rodriguez, and A. Plaza, "A web-based system for classification of remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 4, pp. 1934–1948, Aug. 2013.

[5] I. Hashem *et al.*, "The rise of big data on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, no. 1, pp. 98–115, 2015.

[6] A. Ginie and S. Murugesan, "Web engineering: An introduction," *IEEE Multimedia*, vol. 8, no. 1, pp. 14–18, Jan./Mar. 2001.

[7] M. Lambers and A. Kolb, "Visual assistance tools for interactive visualization of remote sensing data," in *Proc. IEEE Geosci. Remote Sens. Symp. (IGARSS)*, 2010, vol. 1, pp. 4745–4748.

[8] J. Dong, Y. Xue, Z. Chen, H. Xu, and Y. Li, "Analysis of remote sensing quantitative inversion in cloud computing," in *Proc. IEEE Geosci. Remote Sens. Symp. (IGARSS)*, 2011, vol. 1, pp. 4348–4351.

[9] P. Cappelaere *et al.*, "Cloud implementation of a full hyperspectral unmixing chain within the NASA web coverage processing service for EO-1," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 408–418, Apr. 2013.

[10] M. Obitko, V. Jirkovský, and J. Bezdíček, "Big data challenges in industrial automation," *Ind. Appl. Holonic Multi-Agent Syst.*, vol. 8062, pp. 305–316, 2013.

[11] L. Zheng *et al.*, "Applying data mining techniques to address critical process optimization needs in advanced manufacturing," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min. (KDD'14)*, 2014, pp. 1739–1748.

[12] L. Ramaswamy, V. Lawson, and S. V. Gogieni, "Towards a quality-centric big data architecture for federated sensor services," in *Proc. IEEE Int. Congr. Big Data (BigData Congr'13)*, 2013, pp. 86–93.

[13] M. Ceci *et al.*, "Innovative power operating center management exploiting big data techniques," in *Proc. 18th Int. Database Eng. Appl. Symp. (IDEAS'14)*, 2014, pp. 326–329.

[14] Z. Khan, A. Anjum, and S. L. Kiani, "Cloud based big data analytics for smart future cities," in *Proc. IEEE/ACM 6th Int. Conf. Utility Cloud Comput. (UCC)*, 2013, pp. 381–386.

[15] A. J. Jara, Y. Bocchi, and D. Genoud, "Determining human dynamics through the internet of things," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)*, 2013, vol. 3, pp. 109–113.

[16] X. Qin, "A model-integrated computing based tool prototype for designing programmable logic controllers," in *Proc. IEEE Conf. Anthol.*, 2013, vol. 1, pp. 1–6.

[17] C. Edwards, "Not-so-humble Raspberry Pi gets big ideas," *Eng. Technol.*, vol. 8, no. 3, pp. 30–33, 2013.

[18] V. M. Ionescu, F. Smaranda, and A.-V. Diaconu, "Control system for video advertising based on Raspberry Pi," in *Proc. Int. Conf. Netw. Edu. Res.*, 2013, vol. 1, pp. 1–4.

[19] M. A. Islam, T. Islam, M. A. Syrus, and N. Ahmed, "Implementation of flash flood monitoring system based on wireless sensor network in Bangladesh," in *Proc. Int. Conf. Informat. Electron. Vis. (ICIEV)*, 2014, vol. 1, pp. 1–6.

[20] Z. Chen, L. Chen, Y. Zhou, and Z. Xu, "Dynamic slicing of python programs," in *Proc. Comput. Softw. Appl. Conf. (COMPSAC)*, 2014, vol. 1, pp. 219–228.

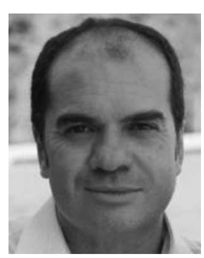
[21] M. Linaje, J. C. Preciado, and F. Sánchez-Figueroa, "Multi-device context-aware RIAs using a model-driven approach," *J. Univers. Comput. Sci.*, vol. 16, no. 15, pp. 2038–2059, 2010.

[22] M. Brambilla, S. Butti, and P. Fraternali, "WebRatio BPM: A tool for designing and deploying business processes on the web," in *Proc. Int. Conf. Web Eng. (ICWE)*, 2010, vol. 1, pp. 415–429.

Enrique Moguel received the Bachelor degree in computer engineering from the University Carlos III of Madrid, Madrid, Spain, in 2010, and the Master's degree in computer engineering from the University of Extremadura, Cáceres, Spain, in 2014. Currently, he is pursuing the Ph.D. degree in smart cities and big data from the University of Extremadura, Cáceres, Spain.

Juan C. Preciado received the Ph.D. degree from University of Extremadura, Cáceres, Spain, in 2009.

He works as a Researcher and Professor with the University of Extremadura, Cáceres, Spain, in which he is currently Vicerrector in charge of quality issues. He has authored more than 50 relevant publications in journals and conferences related to these areas. His research interests include model-driven development, Web engineering, smart cities, and ambient intelligence.



Fernando Sánchez-Figueroa received the Ph.D. degree from University of Extremadura, Cáceres, Spain, in 1999.

He is currently a Professor with the Department of Computer Science, University of Extremadura, Cáceres, Spain. He is also the CEO of Homeria Open Solutions, Cáceres, Spain, a spin-off arisen from Universidad de Extremadura. His research interests include Web engineering, big data visualization, and model-driven engineering.

Miguel A. Preciado received the Bachelor degree in computer engineering from the University of Extremadura, Cáceres, Spain, in 2013.

He is a COO with Homeria Open Solutions, Cáceres, Spain. Currently, he participates actively in two European projects. His research interests include data analytics, visualization techniques, and development of new kinds of user interfaces and experiences.

Juan Hernández (M'89) received the Ph.D. degree from Polytechnic University of Madrid, Madrid, Spain, in 1995.

He is a Full Professor and the Head of the Quercus Software Engineering Group, University of Extremadura, Cáceres, Spain. He has authored more than 100 papers in international journals, conference proceedings, and book chapters. His research interests include service-oriented computing, ambient intelligence, and model-driven development.

Prof. Hernández is a member of ACM.