IoTI:Internet of Things Instruments Reconstruction Model Design

Zhijin Qiu[1*], Zhongwen Guo[1*], Shuai Guo[1], Like Qiu[1], Xi Wang[1], Shiyong Liu[2] and Chao Liu[1]
[1]Department of Information Science and Engineering Ocean University of China Qingdao, China
[2]Qingdao Star Information Technology Co.Ltd Qingdao, China
Email: guozhw@ouc.edu.cn, qzjouc@163.com

***Abstract*—As the 'Industry 4.0' has been proposed, the need of large-scale systems integration has become more urgent. Currently, Internet of things (IoT) systems used in intelligent manufacturing are commonly developed by different organizations using specific technologies and platforms, which brings a lot of difficulties in monitoring equipment access and seamless integration. To solve the aforementioned problem, a novel model of the Internet of things instrument (IoTI) is proposed. The IoTI designed based on the new model can achieve automatic access to different sensing devices,seamless integration and communication for heterogeneous environments. It can also improve the development efficiency. During the development, integration and application of IoTI, we validate the rationality and feasibility of aforesaid model. The results show that the model is efficient and feasible, which realizes the heterogeneous system integration and improves development efficiency.**

***Index Terms*—Internet of things; seamless integration; model calculation; development platform.**

I. Introduction

With the 'Industry 4.0' proposed, the industrial network oriented manufacture has transformed to intelligent manufacturing. During the process of the intelligent manufacturing, we can use the IoT system to monitor and track the manufacturing process, product and environment [1]-[3]. The IoT network is formed by the connection of variety kinds of sensors and Internet, which could manage and process the monitoring data intelligently. A typical case is the sensor collects the data from the monitoring devices, and then transfers the data to the central servers. The server process the data based on some rules and forms the curve, data list and report, which can be visually displayed.

Because of the variety kinds of application scenarios, the concept of virtual instrument (VI) has been put forward [4]-[5]. The VI utilizes the high-performance modular hardware and flexible software to finish the work of testing, measuring and automation application, which replaces the complicated code written work with the graphical development tool to realize the IoT system [6]-[11]. Although the VI technology is easy to use and simplifies the process of development, it still has a few problems: First, it provides the functions of data display and instrument control, but it is lack of the configuration and management functions of display panel, monitoring data and sensor information; Second, the VI can realize the data display function of a single physical instrument. However, it cannot display many related monitoring data on one panel. In many situations, we need to display many related data into one panel to contrastive analyze; Finally, the current VI is hard to be extended and updated. The developer needs to do lots of code work to redevelop when the system needs to be applied to a new environment.

To solve the mentioned problems, we need to provide a new model. In[12]-[14], according to the architecture acquiring information on different circumstances, they provided the dedicated design model for a the specific area. In[15]-[16], the authors gave the methods to solve the data acquisition and energy problem in physical layer. In[17], Song E. et al. designed a service-oriented sensor network data interaction architecture through defining Transducer Electronic Data Sheet (TEDS) and intelligent transmitter communication protocol based on IEEE 1451. In[18], the author proposed the concept of Complex Virtual Instrument System, however, the architecture and application area is restricted. The Open GIS Consortium (OGC) proposes to use Programmable Underwater Connector with Knowledge (PUCK) protocol to integrate the physical devices automatically [19]. The system uses computer to put the configuration information into PUCK model and connects the model to the physical devices, which can realize automatic integration. Although the above research results have solved lots of problems in system integration and development, none of them can solve all the mentioned problem one time.

To solve above problems, we provide a novel IoTI reference model based on the experience of IEEE 1851 [20]. The reference model is designed for IoT system development, which includes information acquiring, processing and output. The IoTI is combined by a group of sensor devices or physical instruments, and faces to a specific kind of application software, which can realize the functions of sensor automatic access, meta data management and large-scale data integration, and provides standard external service interfaces. It can realize the system integration through the connection between different kinds of IoTIs. At present, the IEEE standard association has passed the application of IEEE P2402. Based on the draft of IEEE P2402, we provide standard description on model architecture and interfaces. This paper has realized the model definition, calculation and validation.

The paper is organized as follows: section II provides IoTI construction model; section III makes definitions of every layer; section IV provides the math description of the model and defines two kinds model operations; section V validates the feasibility and performance of this model; section VI shows the conclusions and future works.

II. Construction Model

A. Model Desgin

The IoTI is combined by a group of sensor devices or physical instruments and faces to a specific kind of application software. The input of IoTI is the data collected by sensors. The monitoring data can be displayed after processed by some modules. The processed data can also be accessed by standard interface, such as Web Service, Message Queue (MQ) and File Transfer Protocol (FTP) interface. As shown in Fig.1, the IoT construction model has been abstracted into sensing layer, acquisition layer, management layer and application layer.

Sensing layer: it is the data source of IoTI, It utilizes many kinds of sensors to collect the simulated signal from the monitoring objects and transforms the simulated signal to digital one.

Acquisition Layer: the acquisition layer is responsible for acquiring data. It uses the computer communication interfaces, such as RS232, RS485, Ethernet, USB and so on, to gather the sampled data.

Management Layer: it can realize the functions of data processing, storage and publish through loading the IoTI description file.

Application Layer: it provides friendly GUI panel. Based on loading IoTI description file and designing unified GUI, it can dynamically display the IoT information.

B. Model Description

According to the construction model, we can generate different kinds of IoTI based on different environments through loading different description files. Through abstracting the IoTI description information, the description file format is shown in Fig.2. It includes attributes, TestUnit and Function. The attribute stores the IoTI's number, name, function description and so on. The number is the unique identifier of one IoTI.

III. Layer model

Through the definition above, we can separate the model into four layers. In this section, we propose the functions and realization of each layer.

A. Sensing Layer

Sensing layer is composed of sensor devices, whose main task is acquiring the data of monitoring objects and transforming the simulated signal into digital one which can be understood by computer. The output of sensor devices processing the digital signal is the real numeric value which can be read by users.

B. Acquisition Layer

The acquisition layer uses communication interfaces to collect the data from the devices based on standard protocol. Different devices' interfaces have different communication protocols. During IoTI system developing, the developer needs to rewrite the code based on different devices. To realize the automatic devices access and reduce the code work load, we design an acquisition software and module in acquisition layer. Each acquisition module is mapped to a device. Through the standard definition of the interfaces between acquisition module and software, we realize automatic device access. Application layer and sensing layer can be logically separated by acquisition layer. To be convenient for access, we abstract the sensor device information and standardize the format of the device description file. When the device has been changed, we only need to change the description file in this layer. The system will automatically load the new module to realize device access.

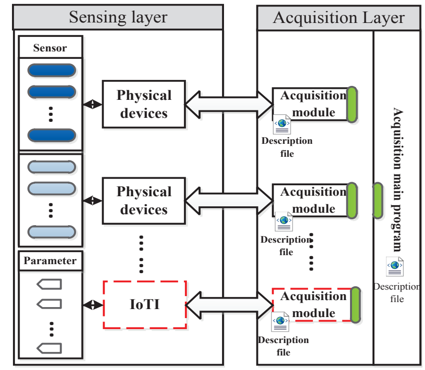


Fig. 4. Data acquisition system architecture

Fig.3 shows the definition of the device description file. It includes the device attributes, interface, sensor parameters and so on. The model node stores the information that matches the loaded acquisition module. The information in this node can be read to discover the matched module. The interface information contains the information of different kinds of interfaces. The sensor parameter describes all kinds of parameters containing the device, such as name, units, precision and relative ID. Through the standard definition of the interface between acquisition module and software, we can use sensor ID and relative ID to identify sensor information. The formation is shown as follow: GlbID&RelativeID,data@RelativeID,data@,...

Fig.4 shows the definition of acquisition system architecture. Through acquiring the sensor device description file, the system can dynamically load the acquisition module to realize device automatic access. It doesn't need to rewrite the software. Especially, the input can be not only physical devices but also the output of Web Service, MQ and FTP interfaces.

C. Management Layer

This layer is designed to process, analyze and count the data from acquisition layer. And they store the processed data into the database. We use the interface management module to output the data to application layer and provide standard Web Service, MQ and FTP interfaces in order to build the foundation of data sharing and system integration. The management layer uses Socket to communicate with acquisition layer. As shown in Fig. 1, it contains data module, calculation module and interface management module.

Data module: the data module is directly connected with acquisition layer to get the monitoring data. Then data module

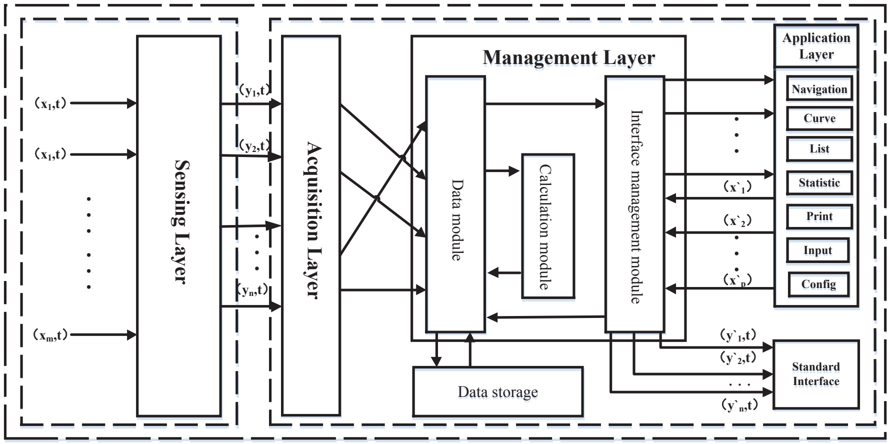


Fig. 1. Internet of things instrument construction model

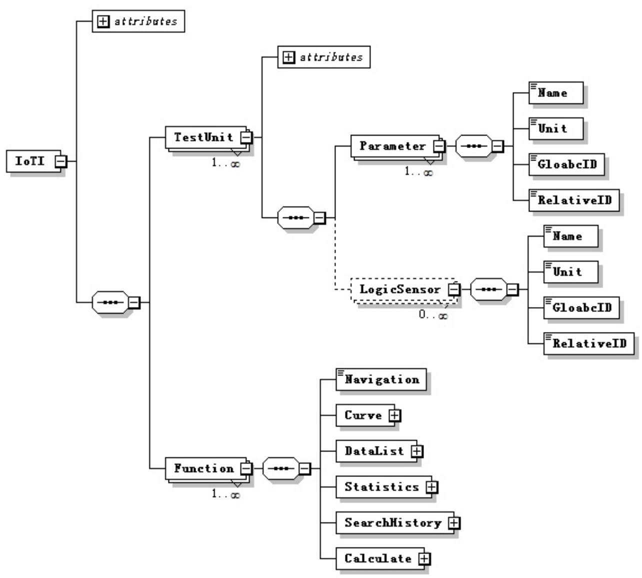


Fig. 2. IoTI description file format

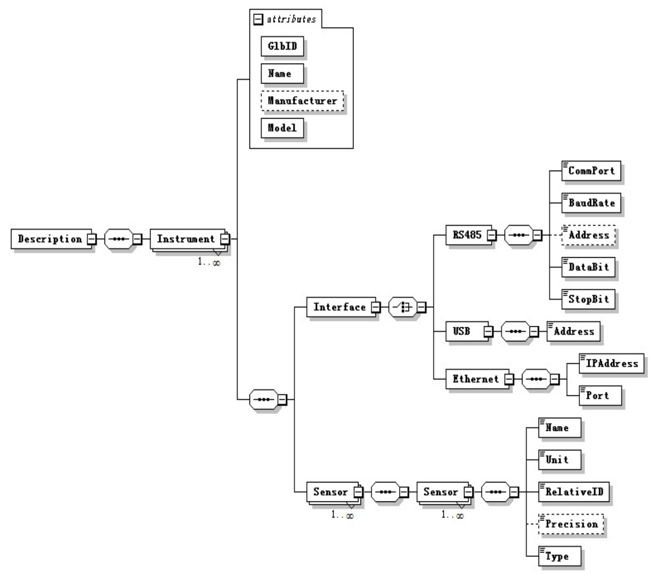


Fig. 3. Device description file format

loads the calculation module through the IoTI description file, and combines with the application layer input parameters to finish the job of data calculation and processing. Based on the flexible database design structure, the module stores the data into database and then transfers the data to interface management module. The device can be identified by sensor ID and relative ID. In IoTI description file, global ID and relative ID in a test unit can identify the monitoring parameter and logical parameter. So the management layer should map the devices and the parameters to the corresponding data and object.

Calculation module: In IoTI system, we need to analyze the collected data. This module is designed to calculate logical parameters. One module maps to one logical parameter. Through IoTI description file, we could find related sensor parameters, constant and mapping module. By loading the module, we can calculate the logical parameters from sensor parameters based on the calculation rule.

Interface management module: The management layer communicates with application layer by interface management module. This module needs to provide and manage data interfaces based on application layer's functions. It also needs to transfer the data to calculation module based on the demands of application layer. Finally, it should provide the standard data interface by Web Service, MQ and FTP interfaces.

D. Application Layer

Application layer is the GUI of IoTI. This layer interacts with interface management module. That is to say, it acquires data from interface management module and also transfers the data to that module. Based on the IoTI development experiences, the IoTI system mainly contains navigation function, input data function, curve display function, data listing function, statistic analyzing function, printing function and so on.

To realize the mentioned functions, we develop the IoT unified GUI, which could display different kinds of monitoring data on the unified panel. The GUI analyzes the IoT description file, and dynamically loads the calculation module to acquire the monitoring data and system information. It can form the IoTI based on the file and display the data.

E. Combination Operation

To make the management work easy, we need to integrate many heterogenous IoTIs that used in different situations. The combination operation will meet this demand.

The combination operation is designed to integrate multiple IoTIs. It takes test unit as the unit to display their monitoring data in one unified GUI.

The input is the combination of the orginal ones. Every IoTI includes the time-serialization data input and non-time-serialization input items. Because all the input data is independent, the input data is the sum of the IoTIs' input. The function and output are the combination of the orginal IoTIs.

F. Connection Operation

During the monitoring procedure, some IoTI's function relies on other IoTI's output. The connection operation describes the cascading system relationship based on the standard data interfaces. For example, when doing the numerical calculation during ocean atmosphere data and ocean power environment data, the later ones result relies on the former ones. The connection operation is based on the standard data interfaces.

IV. Model Validation

A. System Deployment

After defining the model and operation, we design the visual development platform for the IoTI to facilitate the procedure of making model description file. The platform lets the developer to develop the system by drag-drop operations and setting parameters. After developing, we can deploy the system. The IoTI description file is stored in cloud server. The unified client sends the request to get the file and calculation module. And then the generated IoTI acquires, processes and stores the monitoring data. The IoTI can be combined and connected. The more complex the operation is, the lower the efficiency will be.

B. Development efficiency

There are many development methods, such as coding, modification and component reusing. Learning a new object-oriented programming language costs lots of time. According to our model, our platform can lead the developer to finish the development procedures without writing the code. All the procedures are displayed on panels. The result shows that our platform improves the efficiency and reduces the cost.

We take the ocean environment monitoring system as an example. During the procedure of developing, testing and maintaining, our platform shows lots of advantages. We compare our development platform with three other methods, including coding, modification and component reusing[21]-[22]. Fig.5 shows the development efficiency of the four methods. At the beginning, it takes lots of time to develop our platform. However, as the number of the developed system grows, our platform's efficiency is much higher than others.

After finishing the development of IoTI platform, Fig.6 shows the developing, testing, maintaining efficiency of our platform is also ten times higher than that of others.

C. Performance

Through model operation, we can integrate different IoTIs. The more IoTI integrated, the lower the system efficiency will be. The latency of the data display will affect the user experience directly, because the long time from requesting to receiving and displaying data. We test the response time under different integeration levels and different amount data.

We use LAN and WAN to transfer the data. The IoTI integration level changes from one to five. The data size is 100KB, 10M and 100M. All the processors are the same. The result is shown in Fig.7. x axis is IoTI integration level, y axis is the response time. It demonstrates that, no matter what transmission method used, the response time always grows as the integration level become deeper. And the rate of increase becomes bigger. Meanwhile, in the low level of integration situation, the response time is nearly the same. However, as the requested data grows, the response time grows faster. As the integration level and the requested data grow, it needs high quality network. So the LAN is more efficient than the WAN.When IoTI integration to four level, to process 100M data, the response time is longer than one minute. With the IoTI integration level growing, due to the multiple data and information transfers, the IoTI system performance will be reduced. So we suggest not integrate to many IoTIs.

D. Case study

To valid our model, we use the platform to develop a household appliance testing system and an ocean monitoring system in short time. The household appliance testing system includes air conditioner test lab and refrigerator test lab. The lab condition is shown in Table 1. The SR93, UT35A and 8775A are connected with acquisition system through RS485 interface. The WT200 is connected through GPIB. The MX100 is connected with Ethernet.

Every lab has its own description files on monitoring system. Lab A has two test units and lab B has three.

The input is the devices deployed in the lab,which is mapped to the corresponding monitoring parameter, function module and calculation module. There are six kinds of output data interfaces.

Through model combination operation, we integrate the systems into two labs, and check their data in unified client. The combination operation has generated the new description file. The input data is the union set of lab A and B, that means all the parameters of MX100*5, WT200*2, SR93*2, 8775A*3, UT35A*2.

The combination function is the union set of lab A and B. That means the power consumption, refrigeration ability, energy efficiency ratio, on-off ratio, average temperature and air volume. There are six kinds of output interfaces. The

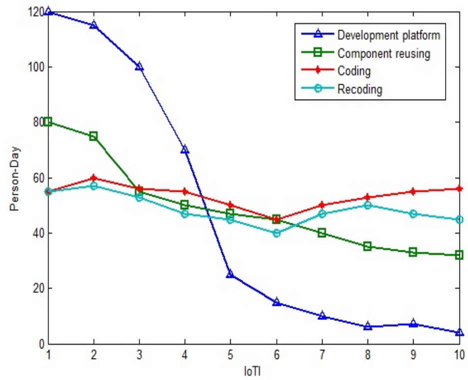


Fig. 5. Development efficiency of the four methods

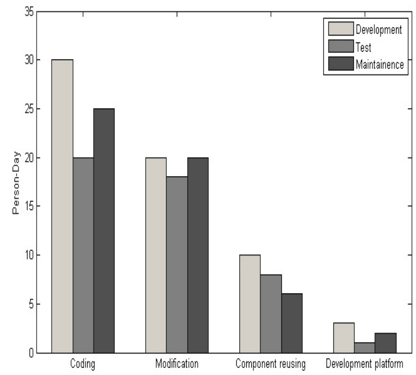


Fig. 6. Comparison of development efficiency

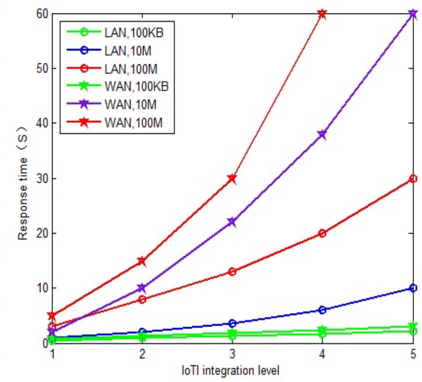
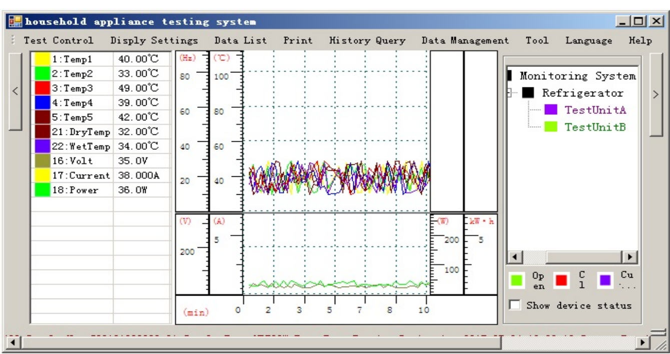


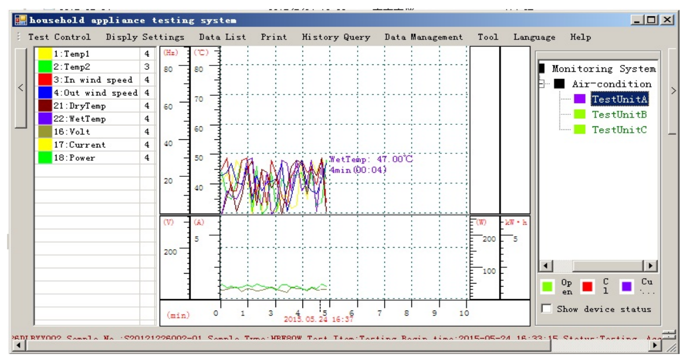
Fig. 7. Integrated performance

TABLE I
LABORATORY CONDITIONS

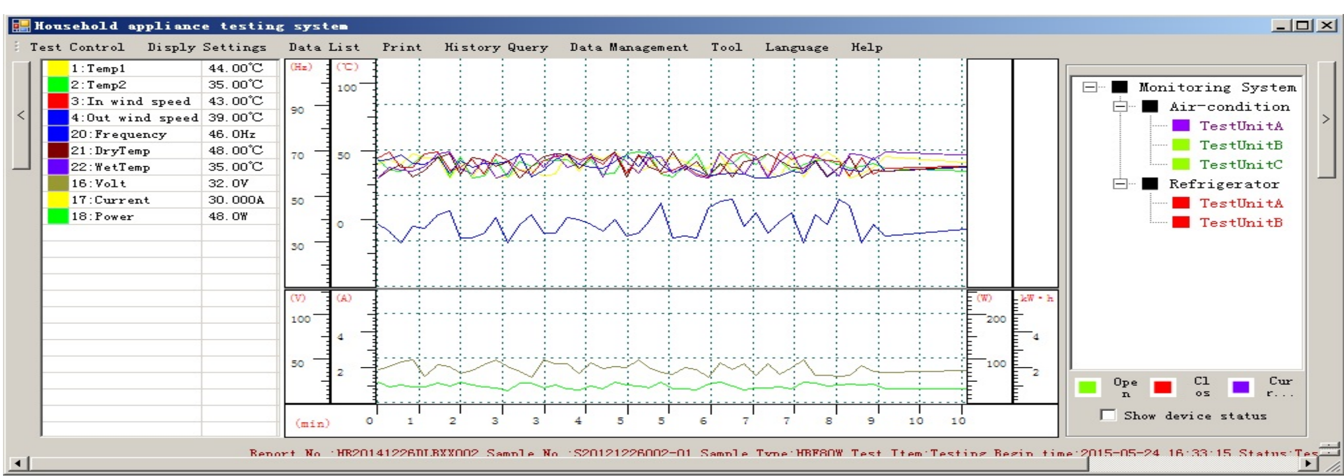
Description	*A*	*B*
Function	Refrigerator test laboratory	Air-conditioner test laboratory
Amount of test unit	2	3
Device	MX100*2,WT200*2,SR93*2	MX100*3,8775A*3,UT35A*2
Calculation module	Power consumption, Ration of energy efficiency, Refrigerating capacity	Power consumption, Ratio of turn-on/turn-off,Air quantity



(a) Refrigerator test system



(b) Air-conditioner test system



(c) Integrated test system

Fig. 8. The GUI before and after combination

contrast GUI of before and after combination is shown in Fig.8.

The ocean monitoring system has three buoys with three ADCPs, six CTDs, three hydro-meteorological instruments,

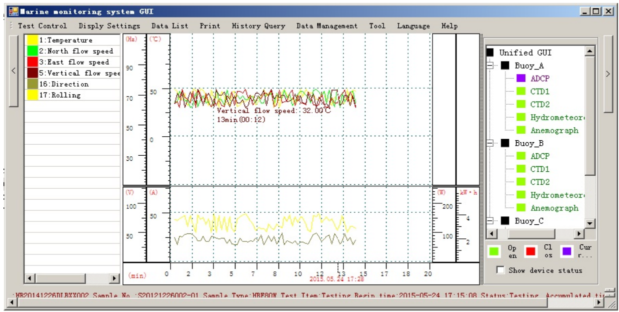


Fig. 9. The GUI of Ocean Observing System

and three anemometers. The buoys are connected with the system in high-speed network. The acquisition system transfers the data every minute.

According to our model, we treat the sensor as IoTI. The buoy is a new IoTI combined by sensors. The GUI is shown in Fig.9.

Meanwhile, the ocean observing system needs to forecast the ocean atmosphere and power environment. We use the connection operation to connect the forecasting system and the monitoring system. The forecasting system transfers the data in NetCDF format by FTP interface. The monitoring system receives the calculated data and the monitoring data, and then displays them on the unified GUI.

Through validation, we draw the conclusion that our model is feasible and reasonable.

V. Conclusion

In this paper, we propose a novel IoTI model, which realizes the functions of device automatic access. It simplifies the procedure of development and increases code reusability. Through defining the model operations, we realize the integration of different IoTIs. We use the household appliance testing system and ocean monitoring system as prototypes to valid our model. The result shows that our model realizes the functions of automatic access and seamless integration, and improves the developing and monitoring efficiency. We test the performance of our model and give the conclusion of response time to different IoTI integration levels. Our contributions as follows:

1) We define the standard description file format and the data interface to realize the function of device automatic access and meta data management.

2) Through the combination and connection operations, we integrate heterogeneous monitoring systems in one unified GUI.

3) Our platform can quickly develop a new IoTI system without writing code and code reuse.

Our future work will focus on data security transmission. Meanwhile, we will extend our system to more common areas.

Acknowledgment

*These authors contributed equally to this work and should be regarded as co-first authors. This work was supported by the National Natural Science Foundation of China grant 61170258, 61103196, 61379127 and 61379128.

References

[1] Z. W. Guo, P. P. Chen, H. Zhang, M. X. Jiang, C. R. Li, "IMA: An integrated monitoring architecture with sensor networks, "IEEE Trans. Instrum. Meas. , vol. 61, no. 5, pp. 1287-1295, May. 2012.

[2] S. D. T. Kelly, N. K. Suryadevara, S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes, "IEEE. Sensors Journal. , vol. 13, no. 10, pp. 3846-3853, Oct. 2013.

[3] A. M. Abdelgawad, M. A. Bayoumi, "Remote measuring for sand in pipelines using wireless sensor network, " IEEE Trans. Instrum. Meas. , vol. 60, no. 4, pp. 1443-1452, April. 2011.

[4] Sheik Mohammed Ali. S, George. B, Vanajakshi. L, Jayashankar. V, Jagadeesh Kumar. V. A multiple loop vehicle detection system for heterogeneous and lane-less traffic. In Proceedings of IEEE I2MTC Hangzhou, China, May 10-12, pp.1-5.

[5] A.Cataliotti, V.Cosentino and S.Nuccio,"A virtual instrument for the measurement of IEEE Std. 1459-2000 Power Quantities," IEEE Trans. Instrum. Meas., vol. 57, no. 1, pp. 85C94, Jan. 2008.

[6] C.Young, M.Devaney and S.Wang, "Universal serial bus enhances virtual instrument-based distributed power monitoring," IEEE Trans. Instrum. Meas., vol. 50, no. 6, pp. 1692-1697, Mar. 2001.

[7] O. A. Postolache, J. M. D. Pereira, and P. M. B. S. Girao. Smart sensors network for air quality monitoring applications. IEEE Trans. Instrum. Meas., vol. 58, no. 9, pp. 3253-3262, Sep. 2009.

[8] Juan Jos Gonzlez de la Rosa, et. al. A novel virtual instrument for power quality surveillance based in higher-order statistics and case-based reasoning. In Measurement Volume 45, Issue 7, Pages 1824-1835, Aug. 2012.

[9] Jin Wang, Xiangming Gao, Qiang Gao. The design of wireless testing system for concrete rigidity based on virtual instrument. In Proceedings of CECNet 2012, pp.2968-2971.

[10] I .Orovic, M.Orlandic, S.Stankovic and Z.Uskokovic, "A virtual instrument for time-frequency analysis of signals with highly nonstationary instantaneous frequency," IEEE Trans. Instrum. Meas., vol. 60, no. 3, pp. 791-803, Mar. 2011.

[11] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks, IEEE Trans. Instrum. Meas. , vol. 58, no. 6, pp. 1886-1896, Jun. 2009.

[12] M. L. Massie, B. N. Chun, D. E. Culler, "The ganglia distributed monitoring system: design, implementation, and experience," Parallel Computing. , vol. 30, no. 7, pp. 817-840, July . 2004.

[13] K. Lee, D. Murray, D. Hughes, and W. Joosen. Extending sensor networks into the cloud using amazon web services. in Networked Embedded ystems for Enterprise Applications (NESEA), in 2010 IEEE International Conference on, pages 1-7, 2010

[14] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. Commun. ACM, vol. 47, no. 6, pp. 34-40, Jun. 2004.

[15] R. X. Gao and Z. Fan, Architectural design of a sensory node controller for optimized energy utilization in sensor networks, IEEE Trans. Instrum. Meas., vol. 55, no. 2, pp. 415-428, Apr. 2006.

[16] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, Power management in energy harvesting sensor networks, ACM Trans. Embedded Comput. Syst. , vol. 6, no. 4, pp. 1886-1896, Sep. 2007.

[17] E. Song and K. Lee. Service-oriented Sensor Data Interoperability for IEEE 1451 Smart Transducers. In Proc. I2MTC, Singapore, May 5-7, 2009, pp. 1043-1048.

[18] Zhongwen Guo, Chao Liu, Xi Wang, Hongyang Ma and Yongguo Jiang. CVIS: Complex Virtual Instruments System Architecture. In Proc. I2MTC, Minneapolis, May 6-9, 2013, pp. 534-539.

[19] I.Puillat, R.Person, C.Leveque, J.Drogou, M.Diepenbroek, P.Garreau C.Waldmann and Y.Auffret, "Standardization prospective in ESONET NoE and a possible implementation on the ANTARES Site," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment., vol. 602, no. 1, pp. 240-245, April. 2009.

[20] Zhongwen Guo, Hao Zhang, et. al. IEEE P1851. Standard for Design Criteria of Integrated Sensor-based Test Applications for Household Appliances. IEEE Standards Association 2012.

[21] R.Jones, A.Rastogi. "Secure coding: building security into the software development life cycle," Information Systems Security, vol.13, no.5, pp. 29-39, Nov. 2004.

[22] A.Ramaswami,T.Hillman,B.Janson,M.Reiner and G.Thomas,"A Demand-Centered, Hybrid Life-Cycle Methodology for City-Scale Greenhouse Gas Inventories," Environmental science & technology, vol. 42, no. 17, pp. 6455-6461, Jul. 2008.