Human-Computer Service Interface in Industrial Monitoring Services

Vyacheslav Dimitrov, Elena Filippova, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{efilippo, dimitrov, dkorzun}@cs.petrsu.ru

***Abstract*—Industrial Internet of Things (IIoT) supports multiparameter monitoring systems, where much data are continuously sensed and processed on the edge in real-time. In this paper, we consider the service development problem for such IIoT data. We study the opportunities of human-computer interface that make such services smarter: a) mobile assistance, b) interactive dashboard, c) post-analysis. The pilot prototype is implemented and tested for real production machinery.**

I. Introduction

Industrial Internet of Things (IIoT) supports multiparameter monitoring systems, where much data are continuously sensed and processed on the edge in real-time [1]. Modern enterprises and factories use information systems developed with Internet of Things (IoT) [2] technology for equipment monitoring to optimize the production operation and quickly respond to faults [3]. We consider the generic service development problem for such IIoT data.

Multiple sensors are installed on production machines to collect data and to transfer the data to server computers, where they are converted into a convenient form for the specialist to understand. The traditional method of checking equipment is long, and searching for faulty equipment can lead to long downtime of the entire production line, while the equipment monitoring system in the enterprise has the ability to check machine components in real-time and report the failure of specific equipment, e.g., see [4].

An equipment monitoring system collects big data to analyze the state of the entire production. Then a person (equipment personnel) makes decisions depending on the situation. In order to speed up understanding of the problem and finding a solution, the information from sensors and services needs representing in a human-friendly form [5], [6].

In this paper, we study the opportunities of human-computer interface that make such services smarter: a) mobile assistance, b) interactive dashboard, c) post-analysis. Our study is based on implemented pilot prototype, which is tested for the use with real production machinery.

The property of mobile assistance is essential when the user operates near the object of monitoring. That is, a service takes the context into account and personalize the assistance to the user [7]. This property frequently happens in healthcare systems where the challenging object for monitoring is a patient [8].

The property of interactive dashboard makes the use of data analytics services more effective, since the user is explicitly involved into the process. This property frequently happens in socio-oriented systems when digital solutions support cooperation between people [9]. Similarly, the user has the possibility to response immediately when a service provides new information for expert analysis.

The property of post-analysis is essential when regularly collected big data from many sensors are analyzed to discover events and their relations. In particular, data fusing and semantic analysis is needed to identify the reasons of the subject situation and predict its behavior [10]. This property frequently happens in location-based systems where each location is a source of regular data flows to collect and analyze.

The rest of the paper is organized as follows. Section II provides related work in the human-computer service interface area. Section III considers interface designing for service-based assistance in IIoT monitoring systems. Section IV discusses interface designing for interactive dashboard with access to multiple services. Section V describes interface designing for post-analysis services with collected data interplay and historical retrospective.

II. Related Work

A lot of research has been done in the field of intelligent monitoring and data visualization. Consider and compare some literature to identify properties that may subsequently appear in the interfaces of the industrial services monitoring system.

A. Traditional approach

Today, the dashboard is the most commonly used way to visualize IoT data. In [11] the authors use traditional methods of visualizing production data. Visualization approaches are also described for more convenient presentation of data and assistance in making effective decisions. For this, different dashboards were used depending on the user's role in production: operator, manager, manager. Work [12] describes an approach to data visualization using the example of an intelligent campus. The web interface was used on public displays and consisted of components: map-based interaction and search, and interaction with perceived data.

B. Contextual interface

The main task of contextual interfaces is to provide the user with information in accordance with the current situation. For example, in [13] a context-sensitive CAVisAP system is presented for visualizing indoor air pollution. The system not only reads the location and time, but also takes into account the level of user sensitivity to pollution and the deterioration of color vision impairments to visualize personalized pollution maps. Work [14] presents a context-sensitive information system Ubi-Board, which includes smart displays with the ability to recognize the closest users to personalize information content. The system can provide information to both groups and individual users depending on the distribution of people in space.

C. AR/VR

The AR/VR technologies allow you to display information in real time and creates a connection between the virtual and physical worlds. Work [15] provides an example of visualizing sensor data through AR, in which, when hovering over a sensor, the application recognizes it, connects to the system and displays information from the sensor. Using AR allowed to increase the speed of detection of faulty sensors, and present information from sensors in a more convenient form. Document [16] uses a method of automated visualization of sensor data to create indoor conditions using VR technology. The system provides users with information to help create comfort conditions in real time, showing step-by-step guides in a virtual environment.

D. Historical visualization

In monitoring systems, information from devices form a large amount of historical data. Typically, data is updated in real time and presented in history in a static form for a certain period of time, which makes it difficult to comprehensively analyze information about the status of the device during the selected period. To visualize historical data in [17] a system is developed for display multidimensional information and dynamically present information. Superset [18] provides an interactive interface for visualizing data. It also supports many methods for visualizing historical data, such as charts, histograms, and trees.

III. Mobile Assistance

An edge-centric IIoT monitoring system needs a server and user interface that provides the necessary information to the user on his devices. The purpose of monitoring is to help make sure that all services or modules of the system are working correctly, and to identify failures in the system. Therefore, it is necessary to display to the user important events and failures in the system so that he focuses on them, minimizing the reaction time to the problem. This section shows prototypes of user interfaces for quick response to solve problems.

A. Widgets

Widgets should highlight the main information in the dashboards in order to focus on the screen the necessary parameters. An interface was developed for the monitoring system to display the values of the sensor and the status of indicators in real time. Since the number of sensors and their indicators can be huge, the interface must be customizable and display only the information that a particular user needs. The user is also provided with advanced widget settings with display conditions. Fig. 1 shows an example of visualizing widgets in the dashboard.

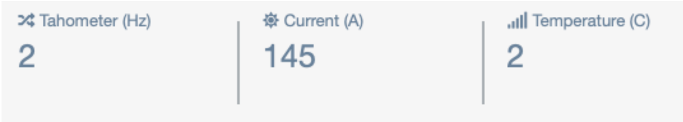


Fig. 1. Widgets example

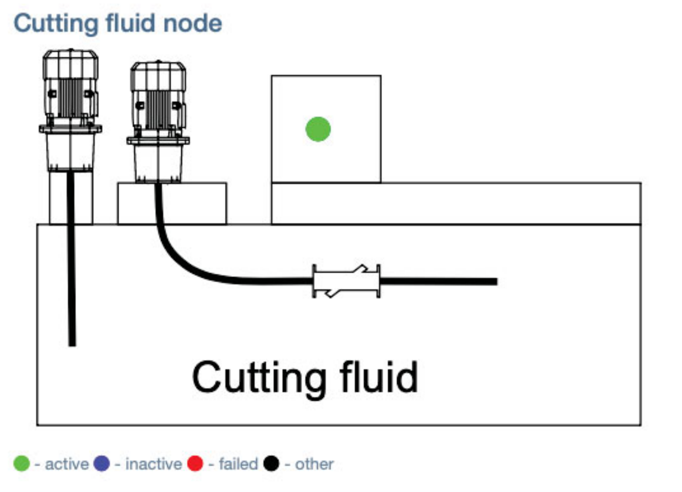


Fig. 2. Mnemoschema example

B. Mnemonic schema

The main task of the mnemonic schema is to simplify the search and identification of information for operational decision making. The mnemonic schema displays the structure of systems and equipment, which helps operators memorize schemes of objects, and also displays information about the current state of equipment and facilitates the systematization and processing of information.

In the interface, the mnemonic schema is an SVG file on which objects with information are displayed. The location of the sensors on the schema is determined in accordance with their location in the real world. Indications of the sensor status are presented in the form of a circle of a certain color. Also on the mnemonic schema you can configure a c hild scheme, which you can go to by clicking on the corresponding object. Fig. 2 shows an example of a mnemonic schema with a sensor and a child element.

C. Alerts

Alerting system is the basis for effective monitoring of equipment. Setting up an alert system supports responding quickly when deviating device parameters, since it is difficult to manually monitor the status of each indicator. Alerts help draw attention to important events, which leads to checking event-related information and device metrics. To monitor the values set Email, SMS and push notifications.

At the moment, the notification system being developed uses push notifications that appear on the device screen on top of all other windows, and e-mail. The user is notified when an event occurs for which he has subscribed, for example, a change in the status of a service or an out-of-bound parameter value.

D. Reports

The system under development allows the user to create reports on specified parameters. There are several types of reports:

1) report on the program module in real time,
2) report on the program module in retrospect (for a given period),
3) report on equipment load.

The report is built from the blocks specified in the settings of the program module. The interface supports the following types of blocks: various graphs with real-time update mode, text messages with support for HTML code, broadcast video from cameras, action blocks with the ability to send the program module a specified event with parameters, equipment load, widgets.

IV. Smart Dashboard

The requirements for the industrial equipment monitoring system Opti-Repair determine the client part, which should implement user interactions with system services. At the moment, the user interface is implemented as a web application for users.

We provide the current general concept for building the user interface of the Opti-Repair system. The architecture is presented in Fig. 3. It applies the following properties.

1) Sensors are mounted on industrial equipment and connected to a special microelectronic board. This board reads data from sensors and transfers it to the system.
2) The Opti-Repair system has set of program modules (PM) that perform specific functions.
3) The host OS of the Opti-Repair system can have several instances of program modules (PMI can also take this as a process).
4) For example, PMI can read data from board, process this data and write to a database storage or send it to a message broker (for another PMI or server).
5) Pay attention that PMI may be greater than PM. For example, let the system have a PM for reading data from the temperature sensor board. Two temperature sensors can be installed on industrial equipment, so two PMI must be created to read data from these two sensors. Also, PM may not connect to any sensors, but this PM may do some other work, for example, read events from a message broker, analyze events and notify the user about some alerts.
6) Each PM must have configuration file in a special system directory. This configuration file may contain a description of the interface blocks (an overview of these blocks is given below).

There are several types of user interfaces:

1) PMI page — a page for displaying information about the current PMI operation (status, events, log and interfaces blocks). This page is designed for the PM developer, but not for the system operator.
2) Service page — a page with information about system service. This page is designed for operator (system service may depend on several PMI).
3) Dashboard — a main page for displaying summary and important information.

These types of user interfaces are based on interface blocks. There are different types of blocks:

1) chart — a chart with line type. Datasets and display style are described in the PM configuration file.
2) text — some text with supporting HTML tags.
3) chart-array — set of charts (data must be array)
4) timeline — the distribution of events over time.
5) widget — small block with the current data value (usually from the sensor or status from any service).
6) *et al.*

In sum, 15 types of blocks are supported. The data for blocks are extracted from databases (or events from the message broker) due to the description in the configuration file. For example, the description of blocks may be as follows:

```
block1.type = chart
block1.name = Rotation frequency, Hz
block1.event.types = value_changed
block1.data.field1 = frequency
block1.data.name1 = Rotation frequency
block1.data.color1 = red
block1.y = frequency, Hz
```

Thus, the PMI page consists of blocks which are defined in the configuration file. Different PM can use the same blocks. The service page also consists of its PMI blocks (several PMI are connected to the service). Dashboard is configured by users and also consists of different PMI blocks or some special blocks (for example, a list of recent important events or list of recent tasks from services). The building UI architecture is presented in Fig. 4.

This architecture allows the user to build dashboard according his requirements. Examples of different dashboards are presented in Fig. 5. Dashboards are built without changing program code.

This user interface has problem with scalability. More than 70 sensors can be installed on equipment, so output values of these sensors in widgets and charts on the main dashboard can be very inconvenient for the user. This huge amount of data does not allow the user to quickly understand the cause of the problem and decide what to do.

As a result, we suggest an idea for creating a smart dashboard. This dashboard will fit to the current situation on the equipment and display only important information for the user. Smart Dashboard has the following requirements:

1) the system operator should do minimal actions to obtain information about the current problem or do nothing for this.

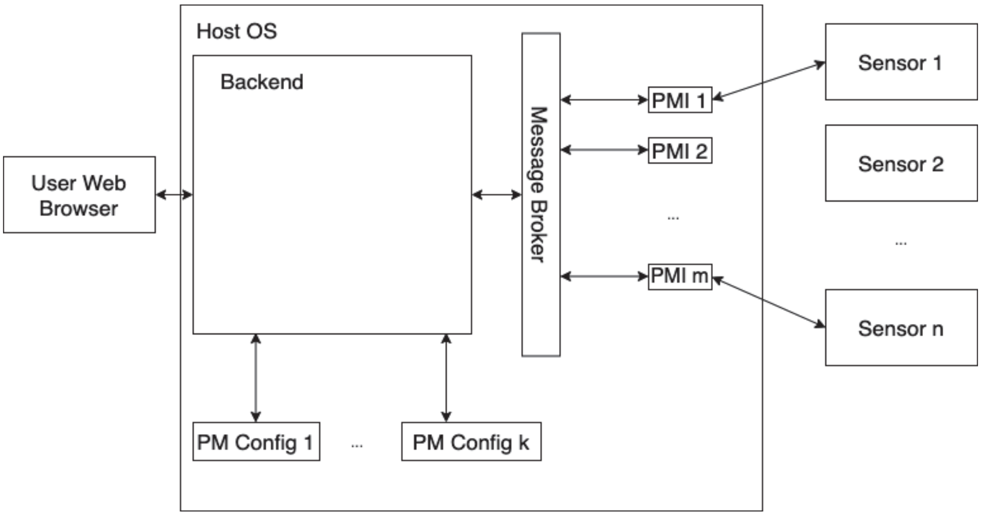


Fig. 3. Basic architecture of the system

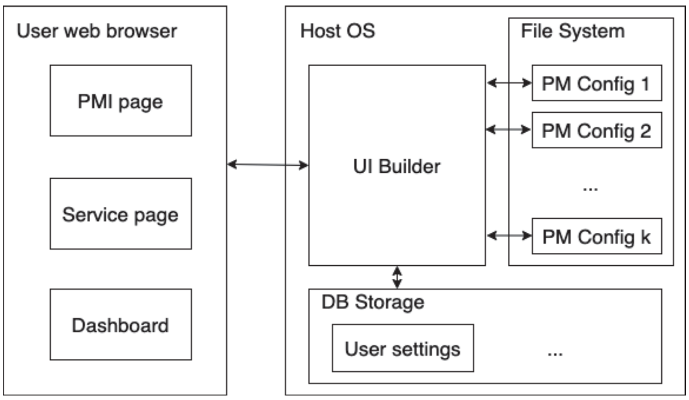
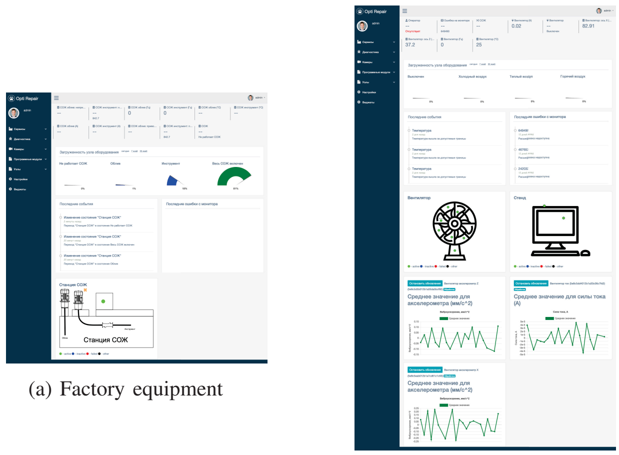


Fig. 4. System User Interface architecture



(a) Factory equipment

(b) Experimental stand

Fig. 5. Dashboard examples

2) in the case the problem is related to the equipment, the dashboard should display all the information related to this problem.
3) in the case problem is absent, the dashboard should receive information from the user settings.

To implement this concept, the following points are suggested:

1) a list of important events is defined. Events are generated by PMI or services.
2) each event is associated with a list of blocks of user interfaces, which visualize information for understanding the causes of the problem.
3) in the case of an event, the dashboard visualizes all associated blocks with this event.

Let us consider an example of this event. The voltage sensor is exceeding limit of the maximum value. The PMI for analyzing data from the voltage sensor emits an event about this problem. The dashboard receives an event, analyses it and builds a new user interface (loading blocks and data associated with this event).

The advantages of this user interface building model are as follows:

1) notifications from the system may not be delivered or ignored by the user, so this increases the reaction time to the problem.
2) the user will always pay attention to the current problem.
3) the user should not go to other pages of the web application to analyze the causes of the problem, so this reduces the reaction time to the problem.
4) in the case system supports several equipment, dashboard shouldn't display all the information from all the equipment, but only with its problem.

The new architecture for building this interfaces is presented in Fig. 6. Now building the user interface depends on important events of the system.

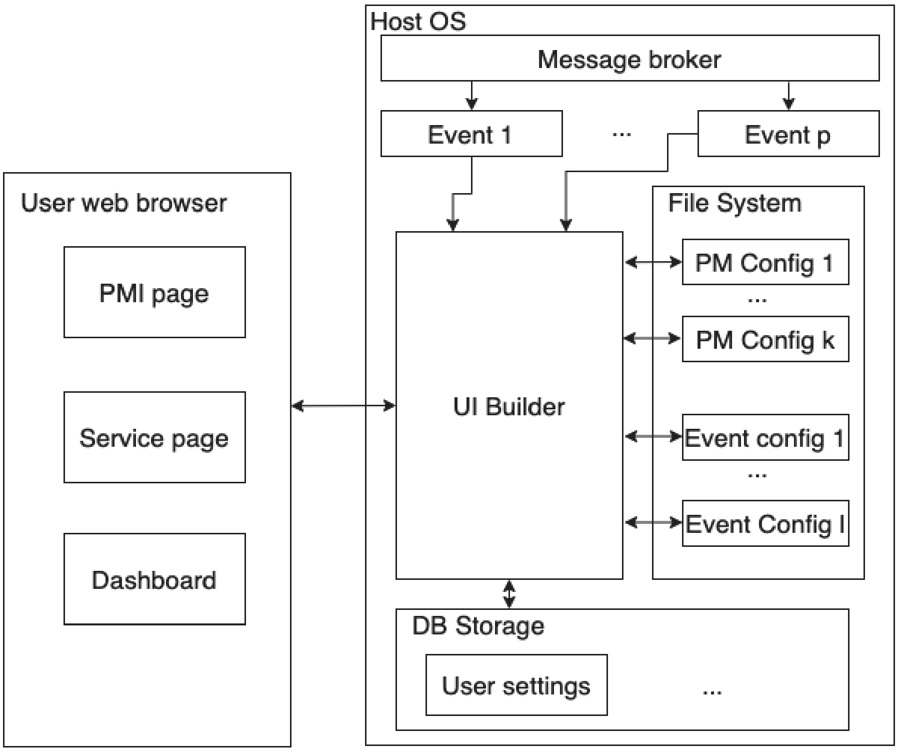


Fig. 6. New User Interface architecture

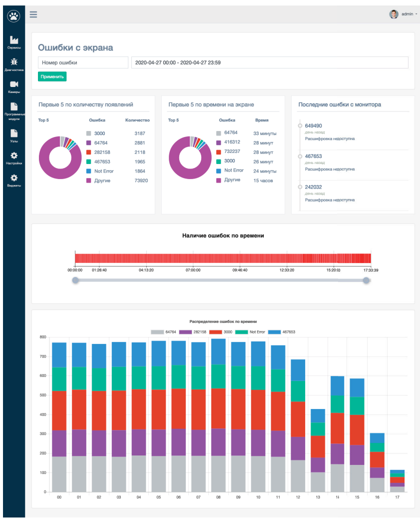


Fig. 7. Report example

V. Post-Analysis

To analyze the causes of problem with the industrial equipment, a report form is used. The date range is selected by the user and the system builds a report from this range. The Opti-Repair system also has ability to generate reports. An example of this report is presented in Fig. 7

This static report does not allow to understand the cause of the problem, if the problem may be related to the operator's actions from the video camera. For example, sensors captured an event that is caused by the penetration of the operator into some equipment node or by some other external impact.

The sensor event and the cause of the problem also can be separated in time. This anomaly can be recognized by watching historical data in dynamics.

To solve this problem, there is the idea of creating the user interface with playing monitoring system events in time, like a video player. Video streams from saved files, data from sensors and events from PMI and services will be generated, and the system will display data in real time (for all blocks of the user interface).

There are three options for implementing this function:

1) uploading all data to the client. But this case has a big disadvantage: the amount of data can be very huge, so the system must extract all the data from the storage and upload this data to the client. So in this case may be a very long delay.
2) emulation of sending data from the server to the client is the same as sending data from sensors to the user interface in real time. But in this case the data will be extracted from databases storage. To interact with the server a web socket can be used. The advantage of this method is that the implementation of the user interface in real time and playing historical data don't change. There may also be no delay to uploading data to the client.
3) combine option: data are uploaded for the client in part.

The user interface should have the following opportunities:

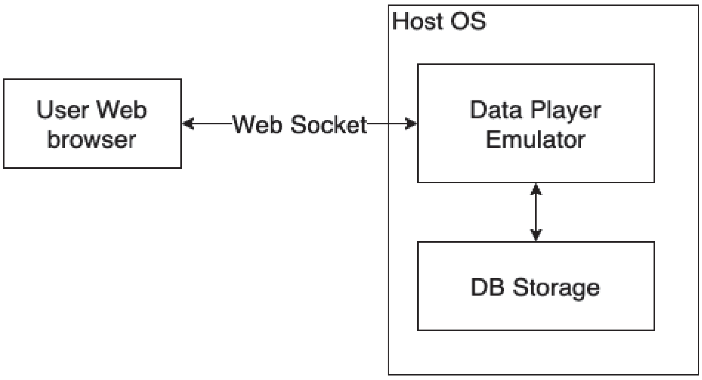


Fig. 8. Historical Data Player Architecture

1) date range selections for historical data.
2) playback speed control. By default, one to one time is used (one second of real time corresponds to receiving data in one second). Other modes are also supported: one to two (one second of real time corresponds to receiving data in two seconds), one to five (historical data in five seconds) and one to ten (historical data in ten seconds)
3) move to any second of historical data range
4) pause playback.
5) save settings for the player and historical data.

The architecture of playing historical data is presented in Fig. 8. Interaction with the server and client parts occurs using web socket protocols. The data player emulator module manages the sending of data to the client and extracts data from the storages.

VI. Conclusion

This paper considered the generic service development problem in IIoT multiparameter monitoring system, where much data are continuously sensed, processed on the edge, and feed constriction of real-time analytics services. Our particular study subject is the opportunities of human-computer interface that make such services smarter: a) mobile assistance, b) interactive dashboard, c) post-analysis. The above elements can be a part of any IoT monitoring system, making the work of personnel more effective and "digitally closer" to the object of monitoring and analytics. To evaluate the opportunities, we implemented a pilot prototype, which is experimented for the use with real production machinery.

Acknowledgment

This research is implemented in Petrozavodsk State University (PetrSU) with financial support by the Ministry of Science and Higher Education of Russia within Agreement no. 075-11-2019-088 of 20.12.2019 on the topic "Creating the high-tech production of mobile microprocessor computing modules based on SiP and PoP technology for smart data collection, mining, and interaction with surrounding sources".

References

[1] A. Gurtov, M. Liyanage, and D. Korzun, "Secure communication and data processing challenges in the Industrial Internet," *Baltic Journal of Modern Computing*, vol. 4, no. 4, pp. 1058–1073, 2016.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[3] D. Hasselquist, A. Rawat, and A. Gurtov, "Trends and detection avoidance of internet-connected industrial control systems," *IEEE Access*, vol. 7, pp. 155 504–155 512, 2019.

[4] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018.

[5] M. D. Lee, M. A. Butavicius, and R. E. Reilly, "Visualizations of binary data: A comparative evaluation," *International Journal of Human-Computer Studies*, vol. 59, no. 5, pp. 569–602, 2003.

[6] F. Ansari, P. Hold, and M. Khobreh, "A knowledge-based approach for representing jobholder profile toward optimal human–machine collaboration in cyber physical production systems," *CIRP Journal of Manufacturing Science and Technology*, vol. 28, pp. 87–106, 2020.

[7] D. Korzun and S. Balandin, "Personalizing the Internet of Things using mobile information services," in *UBICOMM 2016 : The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2016, pp. 184–189.

[8] A. Y. Meigal, D. G. Korzun, L. I. Gerasimova-Meigal, A. V. Borodin, and Y. V. Zavyalova, "Ambient intelligence at-home laboratory for human everyday life support," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 10, no. 2, 2019, in Press.

[9] E. Gilman, O. Davidyuk, X. Su, and J. Riekki, "Towards interactive smart spaces," *J. Ambient Intell. Smart Environ.*, vol. 5, no. 1, pp. 5–22, Jan. 2013.

[10] D. Korzun, E. Balandina, A. Kashevnik, S. Balandin, and F. Viola, *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities*. IGI Global, 2019.

[11] M. Mahmoodpour, A. Lobov, M. Lanz, P. Makela, and N. Rundas, "Role-based visualization of industrial iot-based systems," 07 2018, pp. 1–8.

[12] L. Monti, C. Prandi, and S. Mirri, "Iot and data visualization to enhance hyperlocal data in a smart campus context," in *Goodtechs '18*, 2018.

[13] M. Nurgazy, "Cavisap : Context-aware visualization of air pollution with iot platforms," 2019.

[14] D. M. Sergey Balandin and Y. Koucheryavy, "Next generation teletraffic and wired/wireless advanced networking," in *8th Int'l Conf. NEW2AN and 1th Russian Conf. on Smart Spaces, ruSMART 2008)*, ser. LNCS 5174. Springer-Verlag, 2008, pp. 318–329.

[15] R. Yasmin, M. Salminen, E. Gilman, J. Petajajarvi, K. Mikhaylov, M. Pakanen, A. Niemel?, J. Riekki, S. Pirttikangas, and A. Pouttu, "Combining iot deployment and data visualization: experiences within campus maintenance use-case," in *2018 9th International Conference on the Network of the Future (NOF)*, 2018, pp. 101–105.

[16] W. Natephra and A. Motamedi, "Bim-based live sensor data visualization using virtual reality for monitoring indoor conditions," 2019.

[17] Z. Zhengxing, Z. Shuai, and C. Bo, "Design and implementation of multi-dimensional history playback system based on iot data," 02 2020, pp. 161–166.

[18] H. Cao and M. Wachowicz, "Analytics everywhere for streaming iot data," 10 2019.