Lightweight Mashup Middleware for Coal Mine Safety Monitoring and Control Automation

Bo Cheng, Shuai Zhao, Shangguang Wang, and Junliang Chen

***Abstract*—Recently, the frequent coal mine safety accidents have caused serious casualties and huge economic losses. It is urgent for the global mining industry to increase operational efficiency and improve overall mining safety. This paper proposes a lightweight mashup middleware to achieve remote monitoring and control automation of underground physical sensor devices. First, the cluster tree based on ZigBee Wireless Sensor Network (WSN) is deployed in an underground coal mine, and propose an Open Service Gateway initiative (OSGi)-based uniform devices access framework. Then, propose a uniform message space and data distribution model, and also, a lightweight services mashup approach is implemented. With the help of visualization technology, the graphical user interface of different underground physical sensor devices could be created, which allows the sensors to combine with other resources easily. Besides, four types of coal mine safety monitoring and control automation scenarios are illustrated, and the performance has also been measured and analyzed. It has been proved that our lightweight mashup middleware can reduce the costs efficiently to create coal mine safety monitoring and control automation applications.**

***Note to Practitioners*—The main objective of this work is to provide a service mashup middleware to improve the coal mine monitoring and control automation, which allows the user create the ad-hoc safety monitoring and automation service intuitively. Our solution includes three phases: 1) to access the sensory data with OSGi-based uniform devices access framework; 2) adopt the publish subscribe mechanism to distribute the sensory data; 3) implement a lightweight services mashup approach that supports the on-the-fly integration of different services to build comprehensive and situational applications; and 4) apply the REST principles to define an extensible interface for end users. Our solution is easy to deploy and implement quickly, and may help to improve the coal mine safety monitoring and automation level.**

***Index Terms*—Coal mine safety, control automation, mashup, monitoring.**

I. Introduction

UNDERGROUND mines are usually extensive labyrinths, of which the tunnels are generally long and narrow with a few kilometers in length and a few meters in width. Thousands of mining personnel are needed to work under extreme conditions according to the construction requirements, and hundreds of miners die from mining accidents every year [1]–[3]. It is now widely approved that the underground mining operations are of high risk. In view of this, a monitoring and control system needs to be deployed as one important infrastructure in order to ensure the mining safety and coordinate various tasks. However, underground coal mines mainly consist of random passages and branch tunnels, and this disorganized structure makes it very difficult to deploy any networking skeleton. In such a case, the utilization of a wireless sensor network (WSN) and other sensing devices may have special advantages for realizing the automation of underground monitoring and control due to the rapid and flexible deployment. In addition, the multihop transmitting method can well adapt to the tunnel structure and thus provide enough scalability for the construction of a mining system [4]–[7], and it is very suitable to the comprehensive monitoring and control in coal mines, which can effectively compensate the deficiencies of the exiting underground cable monitoring system.

Traditionally, coal mine safety monitoring and automation systems were typically designed to meet the requirements of a single monitoring application. The coal mine application has already gone beyond the interconnection of a few large back-end systems, and more and more underground physical devices make the state of objects and their surroundings seamlessly accessible to software systems. As a matter of fact, most works are based on monolithic system architectures, which are brittle and difficult to adapt. A necessary step towards coal mine monitoring and control automation is to provide timely and fine-grained comprehensive alarming information and corresponding disposal process. It is necessary so that it allows the users to identify the levels for coal mine safety alarming, and possibly to adjust monitoring and control rules to ensure the coal mine safety. Furthermore, the user can also control the physical devices remotely via the Web. Currently available coal mine safety monitoring and control systems that focus on the real-time information collection are useful, but cannot meet the user needs fully with a very high usage obstacle and often requires a complex operation definition and configuration for monitoring and control automation applications, and cannot meet the demand for ad-hoc services by the end users.

Recently, in the area of comprehensive application integration, some works have introduced the use of “mashup” concepts [8]–[14], also known as user-generated comprehensive applications. However, they mainly focus on mashing up information services and do not address the requirements that come with a physical devices integration. The mashup middleware for coal mine monitoring and control automation needs to

Manuscript received April 30, 2015; revised September 15, 2015; accepted October 01, 2015. This paper was recommended for publication by Associate Editor W. Shen and Editor H. Ding upon evaluation of the reviewers’ comments. This work was supported by the National Natural Science Foundation of China (No. 61132001) and the National Key Technology Research and Development Program of China under Grant 2012BAH94F02.

The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876 China (e-mail: chengbo@bupt.edu.cn; zhaoshuaibyr@gmail.com; sgwang@bupt.edu.cn; chjl@bupt.edu.cn).

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/TASE.2016.2518677

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

rapidly coordinate interaction between the business processes and distributed, multisource sensory devices. Also, the mashup middleware for coal mine monitoring and control automation should change dynamically in a real-time way confronting with continuously and constantly changing for the underground coal mine physical world. With the help of visualization technology, the graphical user interface of different underground physical sensor devices could be created, which allows the sensors to combine with other resources easily.

II. Related Works and Challenges

The coal mine monitoring and control system can be classified into four categories: database oriented, message oriented, service oriented, and REST-based approaches.

A. Database Oriented Approach

Database oriented coal mine safety monitoring system, which is a Structured Query Language (SQL)-based approach [15], [16] to query underground coal mine sensors and other devices in a simple declarative style from the application layer. Thus, this is not the useful and essence of the all collected sensory data, and the device-specific data filtering and feature extraction is essential. Since this method is focused to collect the data from the network, and the data processing technology is needed in the network and the sensor nodes to reduce the amount of data and energy consumption. Hence, a large number of safety monitoring data are generated and processed in the process of coal production. It is also important for safety production in coal mines by analysis of massive of historical safety monitoring data with SQL-based approach to achieve forecast of the safety of coal mines.

B. Message Oriented Approach

Message oriented coal mine safety monitoring system, allows underground sensor devices to communicate with each other regardless of the underlying hardware. This approach masks the underlying network interfaces from the application layer, allowing the user to focus on application development, which provides an asynchronous communication mode. In most cases, the coal mine safety monitoring and control applications are event-driven, and have more advantages on the traditional request-response models [17]–[20]. This approach operates as an asynchronous message, and event-driven communication paradigm that supports many-to-many interactions. Furthermore, advanced message oriented approach adopts publish/subscribe patterns, in other words, the published messages could be defined regardless of the number of subscribers, and consumers subscribe their topic of interests in events that they would like to receive. Therefore, a message oriented approach allows for a loosely coupled relation between publishers and subscribers while greatly enhancing scalability and heterogeneity support.

C. Service Oriented Approach

Service-oriented architecture (SOA) [21], [22] makes the role of current industrial organizations more strategic as they establish high-level interoperability among the different components across the domain, which also provides the solutions for systems integration where the functionalities are encapsulated as interoperable services. In our early works, [23] and [24] presented a novel approach to integrate wireless sensor network into SOA environments using event-driven SOA technologies to develop a closed-loop coal mine safety alarming disposal process, and BPEL is used to define the coal mine safety alarming disposal process. Real-time coal miner localization and tracking system is also proposed in [25], which includes real-time coal miner dynamic display, 3D Geographic Information System (GIS) user interface, alarming, querying trajectories of all miners, and emergency rescue supporting.

D. Representational State Transfer (REST) Based Approach

REST [26]–[32] is a series of guidelines to meet the Web standards presented in a distributed architecture software style. RESTful APIs do not require XML-based Web service protocols (SOAP and WSDL) to support their interfaces. In our early study, a wireless sensor network was combined with the controller area network (CAN) bus technology for the comprehensive and timely monitoring and intelligent early warning in the underground environment, the production data, and the operating status of the equipment, and also design the RESTful API interface for monitoring and control for underground sensor network. All types of parameters were collected and transmitted to the remote monitor center for analysis to provide decision-making information for clients.

E. Challenges and Main Contributions

The acquisition, distribution and integration of large-scale, multisource information, along with the need to appropriately respond to dynamic changes in the physical world in real time, which pose new technological challenges to the provision of coal mine monitoring and control automation services. First, the heterogeneous nature of underground sensory devices require an abstraction from the lower device level layer to a common access layer for other applications, which need an abstraction instance as it has the capability to gather and connect the data from different sensor platforms. Second, with the huge amounts of data to be made available from the edge of the sensor networks or electronic devices to the applications to be available anywhere at any time, how to dispatch, assemble/integrate those sensory data among the distributed, loosely coupled information systems across different business fields or even organizations. Third, the deployment and provision of the coal mine monitoring and automation application is expected to consider with a new range of user-centric data services, how to meet the demand for ad-hoc services by the end users. Hence, we propose an easy to use, deploy and develop upon device-level coal mine safety monitoring and control automation middleware. Also, the middleware can empower the actual end users to create and adapt individual information centric applications. Hence, the situational monitoring and control applications can be created, which makes underground sensors accessible for the Internet applications and users over the Web. To summarize, the primary contributions of this paper are as follows.

1) Propose a lightweight mashup architecture for coal mine monitoring and control automation application. From the perspective of the structure, the lightweight mashup architecture is divided into four layers, i.e., the device resource

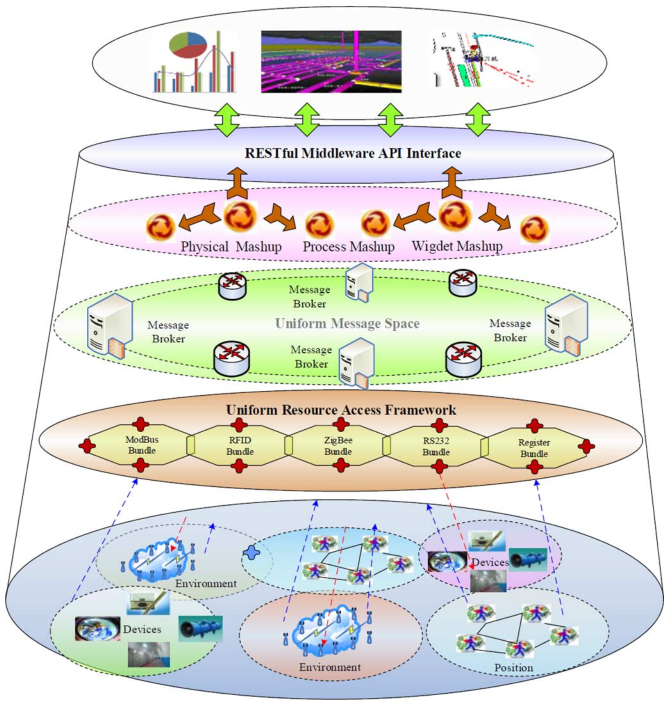

Fig. 1. The layers of the lightweight mashup middleware architecture for coal mine safety monitoring and control automation, the WSN layer, the uniform resource access framework, uniform message space, service mashup layer and application interface.

abstraction and access layer, uniform message space and data distribution layer, service mashup layer, and open service interface layer.

2) Propose a uniform devices access framework, which adopts the OSGi technology to create a uniform protocols management to access the different underground physical sensory devices, and also can dynamically manage the physical sensory devices and the corresponding protocols.
3) Propose a uniform message space and real-time data communication model, which adopts the distributed publish subscribe mechanism, and implemented as one or more message brokers, which matches messages sent from publishers with subscriptions and delivers them to the interested parties.
4) Propose a lightweight services mashup approach that supports the on-the-fly integration of different data services mashup level, and apply the REST principles to define an extensible interface to build comprehensive and situational mashup applications.

The rest of the paper is organized as follows. Section III is the proposed system architecture. Section IV is the illustrated scenarios. Section V is the performance measurement and analysis. Section VI is the conclusions and future works.

III. System Architecture

Fig. 1 illustrates the architecture of lightweight mashup middleware for coal mine safety monitoring and control visualization, which consists of the following component layers: sources of data from WSNs, the RESTful resource API, service mashup, and application interface. These four components are connected into a pipeline, which can provide the coal mine safety monitoring and control automation with real-time intelligence through the RESTful API. Thus, the underground WSN and existing services or other external resources can be integrated into a whole expediently.

Wireless sensor layer, each sensor is plugged into the sensor network, and the nodes are deployed in appropriate areas to acquire the environmental data and detect possible anomalies.

Resource access layer, provides a uniform framework to obtain sensory data and operate various physical devices, which adopts the OSGi technology to create a uniform protocols management framework.

Uniform message space layer, provides a publish/subscribe-based messages distribution service, which contains one or more distributed message brokers, which matches messages sent from sensors with subscriptions and delivers them to control visualization parties.

Service mashup layer, use the sensory data from underground with other resources to form a novel safety monitoring and control application. The mashup approaches can be categorized into physical mashup and widget mashup.

Open service interface layer, provide the REST API interface to access the mashup services, and to get the sensory data and send the control commands on the Internet.

Application layer, a JavaScript-based dashboard is provided for displaying various sensor data visualizations quickly and easily, and thus achieves the control automation of underground physical devices.

A. *Underground ZigBee WSN Deployment*

In underground mines, wireless transmission attenuation is always influenced by various factors such as the corner, the damper, and the slope. Considering these factors, the locations of sensor node should be properly selected for the convenience of communication. Meanwhile, the layout density of nodes should be increased appropriately. However, too dense layout density of nodes may lead to communication latency. ZigBee is a short-range and low-rate wireless network technology based on the IEEE 802.15.4 standard, which defines the network layer and the application layer in the protocol stack. The network comprises of a ZigBee coordinator and multiple ZigBee routers/end-devices, of which the former possesses the functions of initialization, maintenance and control [33], [34], and the latter (only the router) has a forwarding capability of sending the sensed data to a sink node. ZigBee network can support star, cluster-tree, and mesh topologies, but the transmission mechanism of different topologies is different. To be specific, multiple ZigBee end devices can be connected directly to the ZigBee coordinator in a star network, whereas communications must be conducted in a multihop fashion through ZigBee routers in cluster-tree and mesh networks.

The desirable topology of a WSN in coal mines should not only gain high system reliability and robustness, but also realize efficient localization of miners. Considering the complex environment in coal mines, a cluster-tree-based ZigBee WSN is selected and deployed in this study. As shown in Fig. 2, each ZigBee router along with its surrounding devices is viewed as a respective cluster, and the cluster can operate individually as a star network. Here, the node of monitoring center acts as a coordinator that is the cluster head with an identifier of zero. The cluster head cannot only manage inner cluster nodes but also

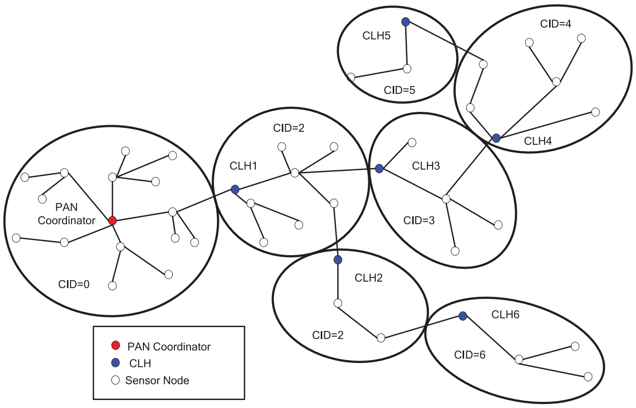

Fig. 2. The topology for cluster-tree-based underground ZigBee WSN, which IS comprised of a coordinator, a cluster header, and multiple sensor nodes.

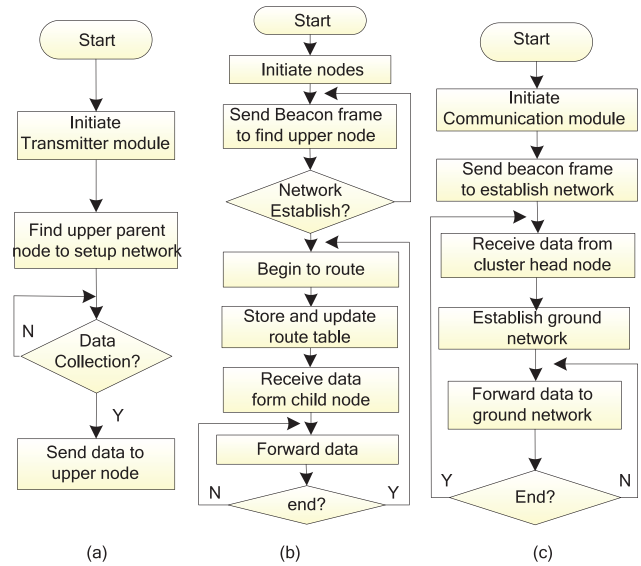

Fig. 3. The operating flowcharts for the underground ZigBee wireless sensor network: (a) the flowchart for the sensor node; (b) the flowchart for the routing node; and (c) the flowchart for the aggregation node.

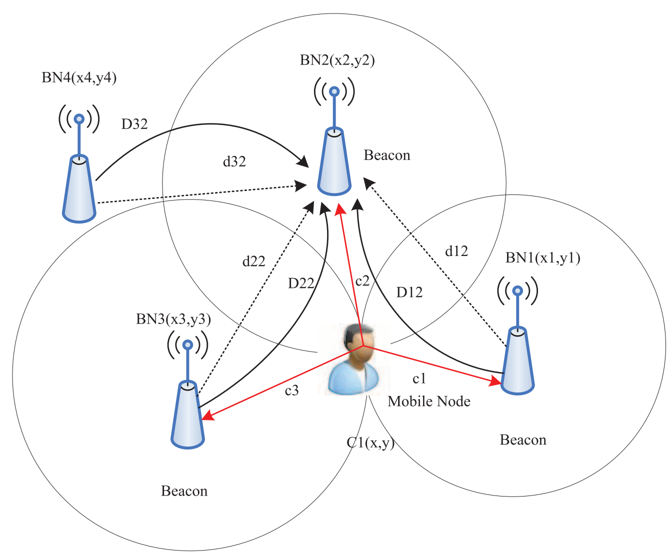

Fig. 4. The positioning algorithm using RSSI values of reference nodes in the targeted position area, which is composed of four beacon nodes and one mobile node.

transmit information to the monitoring center through fiber cables. Therefore, its communication and power capacity is significantly higher than that of battery powered clusters. The cluster node is equipped with methane sensors, humidity sensors, and other sensors. However, its computing capability is low and there are significant limitations in terms of radio bandwidth and battery capacity. Hence, in order to transfer their sensed data, the cluster nodes far away from the cluster head have to select appropriate routes. For example, the neighboring nodes can transmit the data one by one to the cluster head. To facilitate the combination of deployment, each fixed node in the initialized network should have a fixed Personal Area Network Identifier (PAN ID), and sends the beacon frame through broadcasting to the other adjacent equipment. Then, the candidate devices that receive the beacon frame can be applied to the cluster head. If the cluster head is allowed to join in the network, it puts the child node into its neighboring table, and meanwhile the device appends the cluster head as a parent node in its neighboring table, becoming a slave of the network. Once the network capacity reaches a certain limit, the cluster head assigns the slave device as another cluster head in a new cluster network, and a wider range of nodes can join. Thus, all nodes within the network coverage can finally join in the network.

As the first device in the network, the coordinator is responsible for starting the wireless network. Then, the network begins to be constructed, and the network layer selects an idle channel from all compliant channels, which is assigned with a new ID. Afterwards, the wireless monitoring state begins. The above operations can be illustrated as the flowchart shown in Fig. 3(a). For the router nodes, they mainly have the following functions such as discovering network, selecting appropriate router, allocating address to subdevices of the network, transferring data, leaving network, etc. The operating flow can be seen in Fig. 3(b). Finally, for the terminal nodes, they can obtain the physiological information of underground personnel, and further convert the information into data. Immediately after that, the data are sent to the father node, and the instructions from the parent node can be fed back. Its operating flow is shown in Fig. 3(c).

Position node is a mobile node that can move freely within the area enclosed by the reference node. The coordinates of the position nodes can be calculated by positioning algorithm according to all RSSI values [35] of reference nodes in the targeted position area. As Fig. 4 shows, the region is composed of four beacon nodes and one mobile node, which contains two networks as Beacon Node $\mathrm{BN_i}$ ($\mathrm{BN_1}$, BN2, $\mathrm{BN_3}$, $\mathrm{BN_4}$) and mobile node network $\mathrm{C_i(C_1)}$. Especially, the distance between any two beacon nodes is known, and the mobile node can receive the broadcast packets from at least three beacon nodes. In such a case, the mobile node can select the strongest energy beacon node according to the received beacon broadcast packets and attached information. Here, $\mathrm{BN_2}$ is selected as the regional reference node in the beacon network, and $\mathrm{D_{bi}(d_{32}, d_{22}, d_{12})}$ and $\mathrm{D_{bi}(D_{12}, D_{22}, D_{32})}$ are the actual distance and the measured distance from the reference node to the beacon node, respectively. $\mathrm{C_1}$ is attached to the beacon node $\mathrm{BN_2}$, and it only

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

receives broadcast information for the slave node BN_2 and its neighboring beacon nodes BN_1 and BN_3. $C_i(C_1, C_2, C_3)$ is the distance from C_1 to BN_2, and BN_3 uses the RSSI attenuation. In order to get the position information of the unknown node C_1, the scaling factor from mobile slave node BN_2 to neighboring beacon nodes (BN_3, BN_1) can be calculated as follows:

$$\alpha = \sqrt{\sum_{i=1}^{2} \left(\frac{d_{bi} - D_{bi}}{d_{bi}}\right)^2}.$$

The distance of difference correction factor from mobile node to the beacon node can be formulated as $\lambda_i = \exp(-c_i/c_i + \alpha\delta_d)$, in which $\delta_d = 1/2 \sum_{i=1}^{2} d_{bi} - D_{bi}$ is the average value of the differential distance from the mobile slave node to the neighboring beacon nodes. The distance from the mobile node to the three beacon nodes can be corrected as

$$CN_i = c_i + \lambda_i \delta_d.$$

Based on the above formula, the corrected distance CN_1, CN_2, CN_3 can be calculated. Combined with triangular positioning algorithm, the two-dimensional coordinate information of the mobile node can be obtained.

Considering the data exchange between the monitoring center and the monitoring node, most of the data flow from the monitoring node to the monitoring center, and then the monitoring center sends command data to the device. During this process, the data exchange between monitoring nodes is very little. Thus, the routing algorithm can be optimized based on the combination of these characteristics. The upward vertical routing is needed for sending the data to the monitoring center directly along the cluster tree, while the downward vertical routing is also necessary for sending the data from the monitoring center to a node along the cluster tree. Meanwhile, the data exchange between a small amounts of monitoring nodes needs to start the routing discovery and selection. When the network layer receives the data from the upper, it is necessary to judge whether it is a broadcast frame. If it is, the processing in accordance with the broadcast frame starts to handle the data specifically with reference to the ZigBee protocol. Then, the network layer determines whether the data should be sent to the monitoring center. If so, the data will be sent directly to the parent node as the next-hop routing; if not, the data will be sent based on the next known hop routing. If the node does not have the route discovery ability or cannot perform routing selection, it is still routing along the tree. If the data are sent from the monitoring center to the monitor node, it is necessary to send the data to the CAN bus for the fixed node to view its own PAN ID. If it is just the right one, the downward vertical routing can jump to the destination node along the cluster tree.

B. Uniform Devices Access Framework

Beside the wireless sensors network, also the other physical devices are also deployed underground coal mine. An effectively heterogeneous adaptation mechanism is needed to access devices and normalize the way they are exposed via a unified API to applications. As Fig. 5 shows, it can integrate heterogeneous underground physical sensory devices to a uniform service interface. The responsibility of uniform device access framework is not only registering the physical devices and adapting the protocols but also the adaptation of device's capabilities, the publication of resources provided by devices, and follow-up resource lifecycle management.

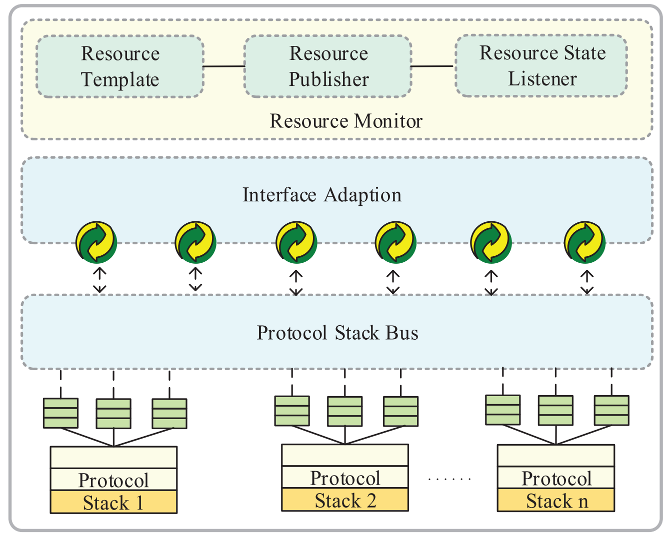

Fig. 5. Uniform devices access framework, which contains the components of protocol stack bus, protocol stack management, interface adaption, and resource monitor.

The uniform devices access framework is implemented in the way of OSGi dynamic component and dependency-inversion pattern. Based on the design pattern, the framework is flexible, which has a high cohesion and low coupling characteristics. Hence, the protocols are hot deployment by definition the protocols bundles, and the protocol stacks bundles can be assembled or reassembled dynamically. Once a new protocol stack bundle is assembled, it registers itself to the protocol stack bus. When a new protocol data packets arrive, it could also receive those data packets, and then checks whether it can analyze the data packets correctly. The whole process of protocol stack bundles assembling is Plug and Play without influencing of the adaptation continuity.

When a new physical sensory device accesses, the physical device register packet is published to the protocol stack bus, then the protocol stacks can analysis whether they can parse it correctly. If a protocol stack can parse the corresponding register packet, and the protocol stack will bind the corresponding device. After that, the other data packets provided by this device are handed over to this instance. The protocol stack manager monitors the state of this instance and checks the validity of device protocol stack binding. If no protocol stacks can analyze this packet, it tries to assemble a protocol stack based on the deployed protocols. If no suitable protocol stack can be assembled, the uniform devices access framework will return the failed analysis result unless the suitable protocol stack is deployed. The device access process is automatically handled without any manual configuration. Each protocol stack has a few protocol stack instances, and each instance is bound to some resources. When one instance becomes unavailable, the protocol stack management will renew a protocol stack instance or reallocate the resources to existing protocol stack instances. The scalability and continuity can be ensured in this way.

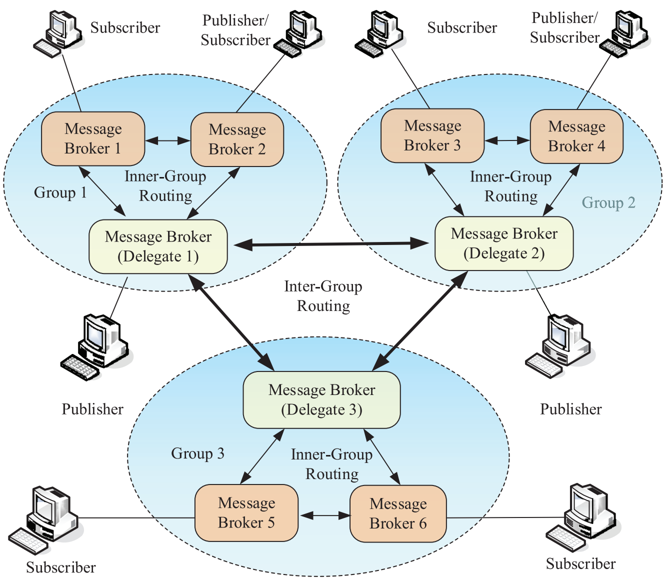

Fig. 6. Cluster-based unified message space, which contains distributed message broker group, there is a broker acts as delegate broker in each group.

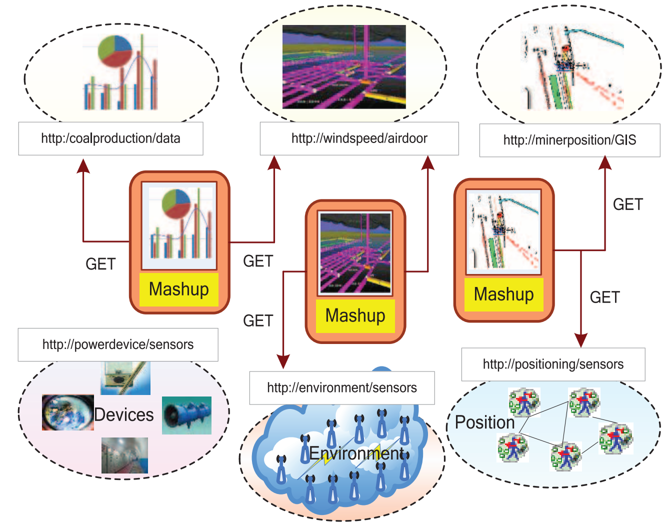

Fig. 7. Mashup-based architecture for coal mine safety remote monitoring and control visualization, which contains three types of mashup engines as data mashup engine, process engine, and widget mashup engine.

C. Uniform Message Space and Data Distribution Service

Real-time data distribution service uses the publish/subscribe mechanism to distribute messages, the publish/subscribe service provide loose coupling between participants, which is typically implemented as one or more message brokers, which matches messages sent from publishers with subscriptions and delivers them to interested parties. The publish/subscribe service is an asynchronous, powerful, and event-driven communication paradigm that supports many-to-many interaction between event clients, where an event is a piece of information that represents an instantaneous occurrence or happening of interest. Each subscriber sends one or more subscription requests to the message broker, specifying the message types that it is interested in. Once a message producer publishes the messages and events, the message broker leverages the routing mechanism to deliver the messages or events to all subscribers that have subscribed to such messages or events. The message broker normally acts a store and forward function, which routes those messages from the publisher to subscriber.

The messages are published with topics, the message subscribers will receive all messages that published on the subscribed topics. To support the messages forwarding and routing, the message broker need to maintain the message sets and the topic subscription table. The message sets stores the latest topic-related messages. The message routing dispatches the incoming messages or the events based on the topic subscription table. To support the large-scale distributed message distribution, the message brokers can organized different groups to provide scalability, and the groups of message brokers may then be linked together for geographic scaling. Therefore, the unified space layer is implemented as a distributed overlay network formed by a set of message brokers in Fig. 6. The message distribution network should support intracluster routing and intercluster routing. In each cluster, a particular event broker, named the delegate broker, is responsible for intercluster routing. A publisher or subscriber connects to an event broker in the unified message space and publishes or subscribes to an event of interest via this broker. When a broker receives a subscription from a client, this broker forwards the subscription to an adjacent node inside a cluster. The delegate broker for this cluster is responsible for forwarding subscription information to other clusters. Likewise, when a broker receives a published event from a client, it forwards the event, via the message distribution network, to brokers that match the subscription. Then, these brokers deliver the event to interested subscribers.

D. Lightweight Service Mashup Approach

The mashup middleware supports the on-the-fly integration of data of different modalities, e.g., the sensory data stored in databases, as well as the real-time data with historical data. Thus, the decision-making process can be executed based on such real-time sensed data. Also, the underground wireless sensor data and other information system can be integrated easily to create new composite monitoring and control automation applications, as shown in Fig. 7. According to mashup concept, the seamless integration of physical sensor objects into the Web. Hence, based on this global view of physical sensors and other existing information resources, a new range of coal mine monitoring applications can be developed to build situational applications via a visual mashup. The mashup runtime engine framework is shown in Fig. 8. When the broker intercepts events, the engine can refer to not only user and system actions during mashup execution, but also the dynamic definition of the composition. Then, the event broker dispatches the events to the widgets in charge of handling the events. The events during mashup execution can be managed by an execution handler based on a publish/subscribe model which addresses the component integration at a presentation level. To be specific, the events generated by the underground wireless sensor network with one mashup component can be mapped to operations corresponding to one or more widget components that subscribe to such events.

Event-driven mashup execution engine consists of four submodules, including the message receiving module, the message pushing module, the data source connection module (the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

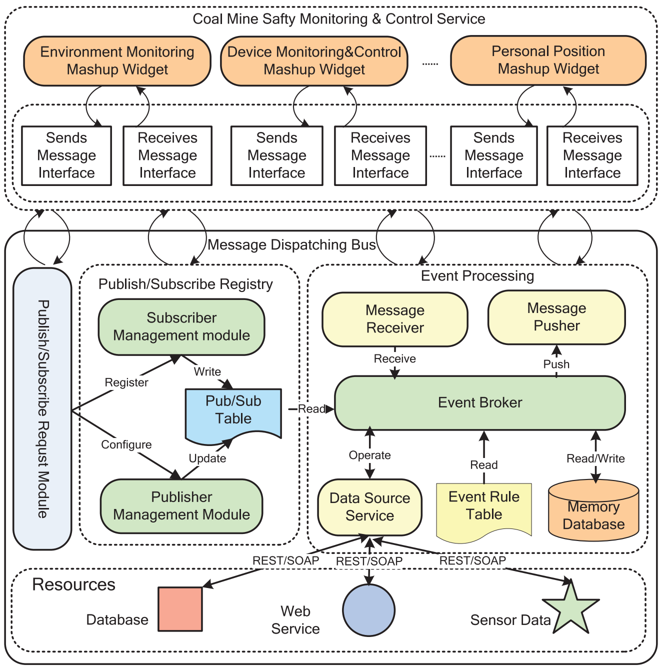

Fig. 8. The implementation for the RESTful Web service mashup runtime engine consists of a client-side application that supports an event-driven execution paradigm, and the mashup ecosystem consists of two parts: mashup maker and mashup runtime engine.

event-driven module), and the event rule tables and memory databases. The function of the message receiving module is to receive the messages from the widget application, and then pass the unpacked messages to the event-driven module for corresponding operations. The message pushing module takes the initiative to push the messages to the widget application side. The event-driven module mainly contains two parts of functions. One is to receive the messages from the message receiving module and send them to the corresponding data source, and the other is to query the event rule tables to determine whether the messages from the data source satisfy the event definition rules. If they comply with the rules, the event-driven module will send the messages to the message pushing module. As to the messages that need not to be dealt with, they will be delivered directly to the message pushing module, and meanwhile be sent to the in-memory database according to the defined time temporary. The event rule table is used to define processing rules of the messages or events, which can facilitate the completion of corresponding logic process.

It is necessary to provide a uniform and easy approach to access the underground wireless sensor devices for the sake of interoperability and integrating these sensor devices into the Internet of the ground coal mine safety monitoring and control system. As a resource-oriented architecture style, REST can provide great conveniences for building Internet-scale distributed applications. The main challenge is the resource bottleneck. The main abstraction of RESTful Web service is the resources, and various resources are linked using their own Uniform Resource Identifier (URIs). The RESTful API on sensor nodes can provide access to sensors and actuators by virtue of the Web. Since each sensor node connects with several sensors and actuators, too many sensor nodes may lead to great difficulty in manual configuration of the central monitoring system. Not only the sensor is a resource, but all sensors can provide accessible resources over the network. Consequently, the sensors will be integrated with the internet and the Web simultaneously, and become abstract resources identified by URIs. Here, some monitoring and control automation scenarios of the proposed RESTful APIs are presented. The majority of service requests for controlling and monitoring the devices can be handled in a single invocation.

Controlling the list of available devices, an operator can control the list of available physical devices sending a GET request to retrieve the list of possible configurations and use this information in order to access each context (URI: /Resource/ < id >).

Accessing to the current setting of a device, the parameters of a device represents the current setting of a device. The list can be retrieved using a GET request to /Resource/ < id > /Device/ < id > /Parameters, where /Device/ < id > identify a device with configuration /Resource/ < id > /.

Retrieve the list of devices controlled by a given device, assuming that the URI /Recource/ < id > /device/ < id > represents the device supervisor, the list of controlled instrument can be retrieved using a GET request to /Resource/ < id > /Device/ < id > / that return the list of URIs identifying the devices controlled by the supervisor.

Retrieve the list of available commands of a given device, the URI /Resource/ < id > /Device/ < id > / addresses the device that we want to use. We can retrieve the list of available commands sending a GET request at the URI /Resource/ < id > /Device/ < id > /Command/.

Execute a measure, we can execute a measurement command sending a POST request at the URI /Resource/ < id > /Device/ < id > /Command/measure the results of the measure can be fetched sending a GET request to the URI /Resource/ < id > /Device/ < id > /Attributes/ < id > /.

IV. Monitoring and Control Automation Illustrations

The flexibility of control automation application framework can be achieved according to RESTful principles, which facilitate the development of remote control automation applications in the form of physical mashups. Thus, the underground physical devices become Web resources that can be addressed and used to build mashup. In this study, a JavaScript based dashboard is provided for conveniently displaying various sensor data visualizations and realizing remote control automation of underground physical devices. Therefore, a range of new applications can be realized based on this unified view of a Web of information resources and physical sensor objects. Using operations such as greater than, if/then, and/or, etc., the staff can create monitoring and control rules to automate the underground coal mine sensors in the form of physical mashup in just a few clicks. Furthermore, the staff can combine environmental sensory measurements with services offered from underground physical devices, such as sensing the environment parameters in real time and switching them on/off.

The first scenario is the real-time data monitoring on the underground coal mine. Fig. 9 shows the global real-time sensor data mashups along with the coal mine graphs. The user cannot

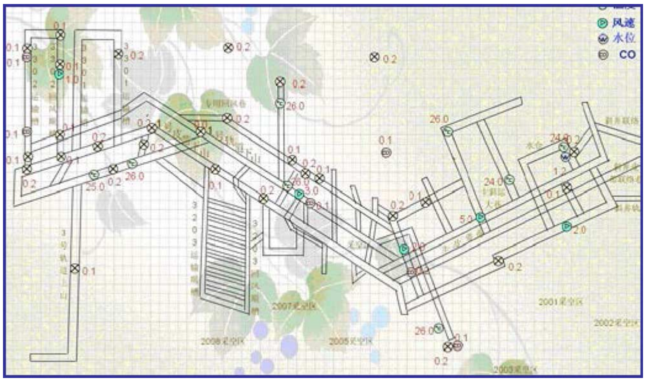

Fig. 9. The integrated and global mashup view for monitoring and control automation in underground coal mine.

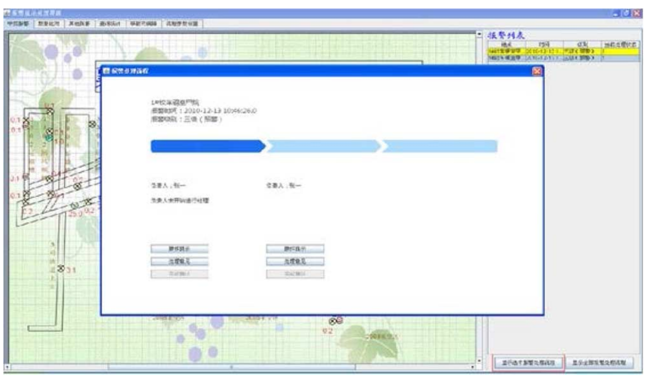

Fig. 10. Coal mine safety alarming disposal process scenarios. It is necessary to detect the complex accidents and to take some certain measures immediately.

Fig. 11. The remote video surveillance for the coal mine, which can provide the first-hand image data for the analysis of accidents.

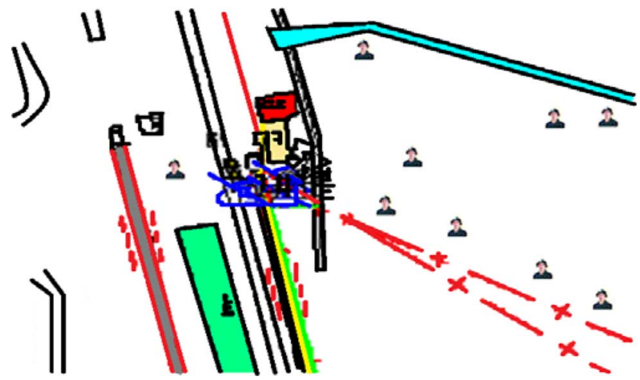

Fig. 12. Miner positioning combined with GIS. It is important to achieve the precise positioning of coal miners once the disaster happens.

only query a single mote but also access the average data. The client can only query a single mote in the former case, whereas he can query all the motes in the WSN to calculate the average by the web server in the latter case. In order to dynamically display the current underground environment parameters, the application can directly send HTTP GET request to http://…/ https://www.sensorrest.com/sensors/all.json or subscribe to those resources.

The second scenario is safety alarming disposal process scenarios. Especially, when a safety alarming trigger condition is satisfied with the corresponding complex situation pattern, the higher level disposal process event is created, and some human or automated process that is invoked when the trigger event is reached. Therefore, it is necessary to detect the complex accidents pattern and generate the corresponding alarming disposal process for underground coal mine safety. As Fig. 10 shows, when the coal mine safety alarming occurs, it is necessary to take some certain measures, and to deal with the corresponding alarming disposal process, which is triggered by the alarming events. All the information for the safety alarming, and the charged staffs for each disposal steps, and the status for the current disposal process should be monitored in a real-time way during the course of the alarming disposal processing.

The third scenario is the remote video monitoring scenario. Through real-time video monitoring, the control personnel cannot only obtain an image record of the underground work site and production safety, but also detect the incident seedling and take preventive measures. Besides, the real-time monitoring can provide the first-hand image data for the analysis of accidents. As Fig. 11 shows, the end-user can watch live images using a computer, personal digital assistant (PDA) and monitoring device, and browse images via the Internet. Thus, the video monitoring and control automation application can provide a general picture of the current activity of all video sensor nodes.

The fourth scenario is the coal miner personnel positioning. Once the disaster happens, available escaping paths should be found immediately. It is also important to achieve the precise positioning of coal miners, such as the accurate miner number, distribution and location of each coal miner. Besides, in order to improve the efficiency of the rescue work, it is also necessary to provide an intuitive personnel distribution map at any time. Combined with coal mine GIS, the accurate miner number, distribution and location of each coal miner can be obtained, as shown in Fig. 12. The administrator staff in coal mine safety monitoring center cannot only inquire the current and historical distribution records for underground coal miners, but also request the historical coal miner's tracking routes on the underground coal mine GIS system.

V. Performance and Discussion

The lightweight mashup middleware consists of several elements and their performance can be tested and evaluated through different scenarios.

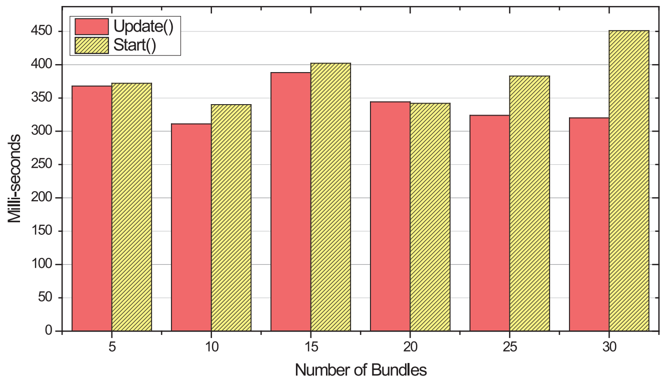

Fig. 13. Execution times with bundles variation. Measure the execution time to start the device bundles and update the device bundles, respectively.

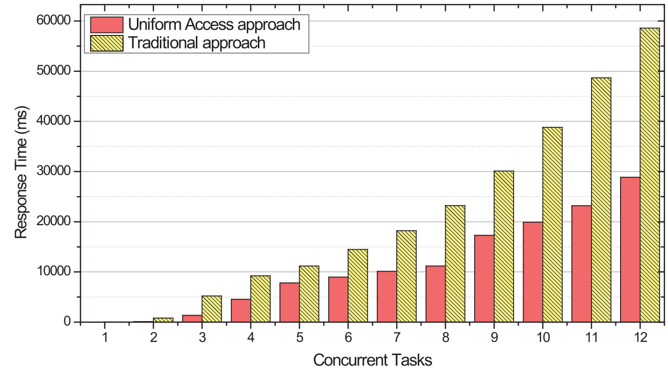

Fig. 14. Performance evaluations of different models. The response time for OSGi framework against native traditional Java code approach.

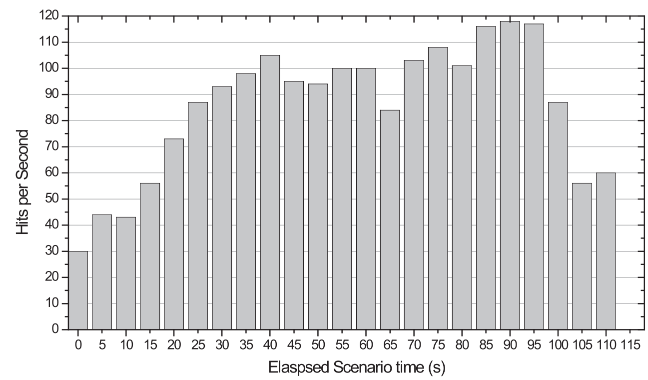

Fig. 15. Simulated 30, 40, 50, 60 to 110 sensory devices to send concurrent sensory data, respectively.

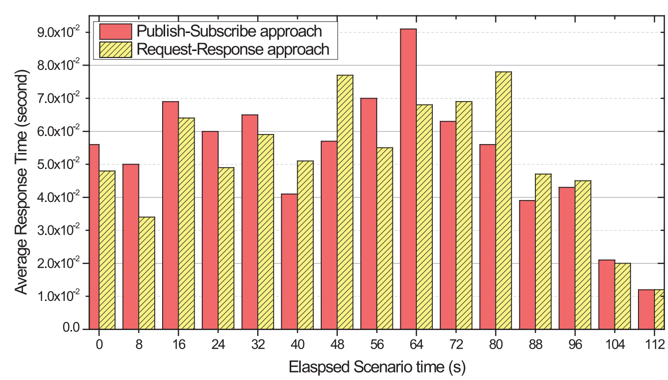

Fig. 16. The comparison diagram for transaction response time for publish-subscribe approach and traditional request-response approach.

A. Experimental Setup

We need a flexible and reusable experiment environment. The lightweight mashup middleware software was installed on a PC server (Intel Xeon E5420 2.5GHz with 10G RAM). We choose a real coal mine as the basic experiment environment, where 30 Mica2 sensor motes were deployed on a tunnel wall of about 10 meters wide and 5 meters high, and also 100 other same coal mines have been simulated as a semi-physical emulations experiment environment to evaluate the performance of the lightweight mashup middleware.

B. Experiments

Here, we prepared four different experiments to test the lightweight mashup middleware with actual scenarios, and each experiment reflected the performance of the whole system from a different aspect.

In the first experiment, we test the performance of the uniform devices access framework. Here, the OSGi-based devices access framework to be deployed on top of physical sensors and sensor networks with capabilities for plugging any device according to a Plug&Play paradigm, and offering the prompt and universal use for the monitoring and control applications. We measure the execution time when increasing the number of device bundles, and especially, we measure the execution time to start the device bundles and update the device bundles, respectively, as depicted in Fig. 13. When increasing the number of bundles, the execution time of the two methods Start() and Update() seems to vary independently of the number of bundles. However, when increasing the level of dependencies, the execution time of the methods increases linearly with the number of dependency levels. In fact, when a bundle is started, its dependencies are resolved first. Also, we compare the response time for OSGi framework against native traditional Java code approach, especially, each sensory device bundles is defined as sequentially gathering data, and the data retrieval latency are both set as 100 ms. Fig. 14 shows the experiment results. The response time is decreased drastically with OSGi based uniform devices access framework. The delay is caused by the network connection establishing and dependency resolution at the beginning for the uniform devices access framework, with the elapsed time, which stores all of its persistent state in the later, and the invocation time is significantly reduced.

In the second experiment, we compare the data distribution performance of publish-subscribe approach and traditional request-response approach, and the response time was evaluated when concurrent sensory data were received from the different underground coal mine devices. Especially, for the convenience of receiving the sensory data and immediately dispatching them to applications, we simulated 30, 40, 50, 60 to 110 sensory devices to send concurrent sensory data, respectively, as Fig. 15 shows, and the corresponding average response times were tested as Fig. 16 shows, to the response

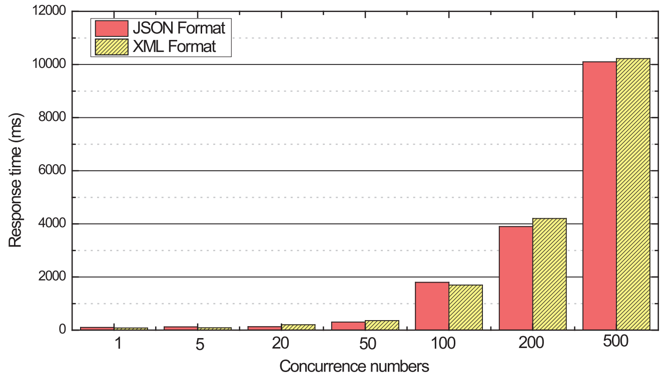

Fig. 17. The comparison diagram for response time between the JSON data format and XML data format.

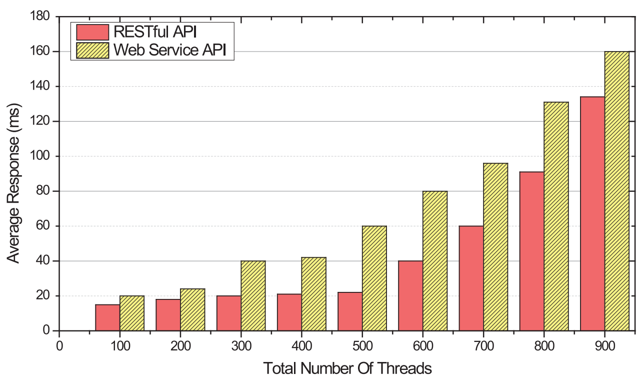

Fig. 18. The comparison diagram for the average response time between the Web services API and RESTful API interface.

time of the traditional request-response pattern, the maximum response time is 180 microsecond, the average response time is 150 microsecond. However, to publish-subscribe pattern, the maximum response time is 90 microsecond, the average response time is 50 microsecond. Because of publish-subscribe pattern adopts an asynchronous multicast-like communication pattern, and can dispatch the published data from waiting an acknowledgment for the subscriber. Thus, publisher can quickly move on to the next receiver within deterministic time without any synchronous operations. Therefore, the data-centric paradigm is most efficiently realized by publish-subscribe based data distribution pattern rather than traditional request-response pattern.

In the third experiment, we compare the response time for different data format. When implementing the REST API interface, either JavaScript Object Notation (JSON) or Extensible Markup Language (XML) can be selected as the data format. We took the remote methane concentration monitoring services as an example and conducted performance tests on the REST Web services established using the JSON and XML data transmission formats, respectively. The result is shown in Fig. 17. When the concurrent numbers were the same, the minimum time of the JSON data format was slightly shorter than the minimum time of the XML data format. Because of XML decoding means more or less handling the trivial events, and with JSON processing outperforming XML processing for the same encoded content.

In the fourth experiment, the concurrent threads were applied to invoke the remote control service with RESTful API and Web service API respectively. Here, the number of concurrent threads was automatically generated and configured. The goal of this experiment is to understand the scalability, the overhead, and the flexibility of our proposed RESTful APIs relative to the existing Web service implementation. The response time comparison for both RESTful API and Web services API interface is shown in Fig. 18. The traditional Web service API consumes much more time than RESTful API. Considering the average response time, RESTful API performs better than Web service as expected, for SOAP messages, the actual payload is included inside the envelope element, whereas, for REST entire message is the payload. Thus, SOAP service would have to perform additional processing to extract the payload information. Similarly, when sending a response message, SOAP service would have to perform additional processing to construct a SOAP formatted message.

VI. Conclusions and Future Work

This paper builds a lightweight mashup middleware for coal mine safety remote monitoring and control visualization. Focus on the design and implementation for underground ZigBee wireless sensor network deployment, uniform devices access framework, distributed data distribution service, event-driven mashup service execution engine, and RESTful-based open API interface. The main novelty of this study is to develop a lightweight mashup middleware for coal mine monitoring and control middleware which is easy to use and install for engineers. Since most of the application is Web-based, any personal computer and a web browser can connect the Internet and enter the Web page to use the application, and which can reduce the costs of coal mine safety monitoring and control automation. Therefore, it is expected to be a main contribution to coal mines for better and safer working environments. Several issues remain to be addressed further. First, as the expansion of existing coal mine safety monitoring and control system, visualization technology can further improve the visibility of underground sensor objects, such as 3D technology, which provides significant support for decision making and real-time control in underground mines. Second, it is essential to optimize the real-time data distribution service and data congest scheduling strategy with different QoS constraints for a large-scale coal mine deployment. These works are currently in progress in our lab.

References

[1] K. Page, "Blood on the coal: The effect of organizational size and differentiation on coal mine accidents," *J. Safety Res.*, vol. 40, no. 2, pp. 85–95, 2009.

[2] L. Mallet, C. Vaught, and M. J. Brnich Jr., "Sociotechnical communication in an underground mine fire: A study of warning messages during an emergency evacuation," *Safety Sci.*, vol. 16, no. 5, pp. 709–728, 1993.

[3] M. Ndoh and G. Y. Delisle, "Underground mines wireless propagation modeling," in *Proc. 60th IEEE Veh. Technol. Conf.*, 2004, vol. 5, pp. 3584–3588.

[4] J. Wood, J. Dykes, A. Slingsby, and K. Clarke, "Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1176–1183, Nov.–Dec. 2007.

[5] X.-G. Niu, X.-H. Huang, Z. Zhao, Y.-H. Zhang, C.-C. Huang, and L. Cui, "The design and evaluation of a wireless sensor network for mine safety monitoring," in *Proc. IEEE GLOBECOM*, 2007, pp. 1230–1236.

[6] M. Li and Y.-H. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 5, no. 2, pp. 1–29, 2009.

[7] G.-Z. Chen, Z.-C. Zhu, G.-B. Zhou, C.-F. Shen, and Y.-J. Sun, "Strategy of deploying sensor nodes in the chain wireless sensor network for underground mine," *J. China Univ. Mining Technol.*, vol. 18, no. 4, pp. 561–566, 2008.

[8] A. Bouguettaya, S. Nepal, W. Sherchan, X. Zhou, J. Wu, S.-P. Chen, D.-X. Liu, L. Li, H. B. Wang, and X.-M. Liu, "End-to-end service support for mashups," *IEEE Trans. Serv. Comput.*, vol. 3, no. 3, pp. 250–263, Jul.–Sep. 2010.

[9] R. Tuchinda, C.-A. Knoblock, and P. Szekely, "Building mashups by example tuchinda," *ACM Trans. Web*, vol. 5, no. 3, pp. 1–45, 2011.

[10] Z. Yang, F. Yushun, H. Keman, T. Wei, and Z. Jia, "Time-aware service recommendation for mashup creation in an evolving service ecosystem," in *Proc. IEEE Int. Conf. Web Serv. (ICWS)*, 2014, pp. 25–32.

[11] D. Benslimane, S. Dustdar, and A. Sheth, "Services mashups: The new generation of web applications," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 13–15, Sep.–Oct. 2008.

[12] E. Maximilien, A. Ranabahu, and K. Gomadam, "An online platform for web APIs and service mashups," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 32–43, Sep.–Oct. 2008.

[13] J. Cao, Z. Wen, and T. Wei, "Dynamic control of data streaming and processing in a virtualized environment," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 365–376, Apr. 2012.

[14] A. Bozzon, M. Brambilla, F. M. Facca, and T. Carughu, "A conceptual modeling approach to business service mashup development," in *Proc. IEEE Int. Conf. Web Serv.*, 2009, pp. 751–758.

[15] Z. Wentao and L. Suqing, "Analysis of coal mine safety monitoring data based on column-oriented database," in *Proc. 2nd Int. Conf. Artif. Intell.*, 2011, pp. 1920–1922.

[16] W. An and M. Li, "The research of coal-mining control configuration software's real-time database," in *Proc. 3rd Int. Symp. Comput. Sci. Comput. Technol.*, 2010, pp. 074–076.

[17] Y. Tang, "Pervasive high reliable monitor and alert system based on EDA," in *Proc. Joint Conf. Pervasive Comput.*, 2009, pp. 257–262.

[18] B. Yang, "The research on service-oriented virtual environment visualization platform for coal mine," *Adv. Mater. Res.*, vol. 271–273, pp. 303–307, 2011.

[19] J. C. Ralston, D. W. Hainsworth, R. J. McPhee, D. C. Reid, and C. O. Hargrave, "Application of signal processing technology for automatic underground coal mining machinery," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. 249–52.

[20] M. Li and Y.-H. Liu, "Nonthreshold-based event detection for 3D environment monitoring in sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 12, pp. 1699–1711, Dec. 2008.

[21] W. Tan, Y. Fan, M. C. Zhou, and Z. Tian, "Data-driven service composition in enterprise SOA solutions: A Petri net approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 686–694, Jul. 2010.

[22] F. Jammes *et al.*, "Technologies for SOA-based distributed large scale process monitoring and control systems," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Conf.*, 2012, pp. 5799–5804.

[23] C. Bo, X. Qiao, B. Wu, X. Wu, R. Shi, and J. Chen, "RESTful web service mashup based coal mine safety monitoring and control automation with wireless sensor network," in *Proc. IEEE 19th Int. Conf. Web Serv.*, Honolulu, HI, USA, 2012, pp. 24–29.

[24] C. Bo, P. Zhou, D. Zhu, and J. Chen, "The complex alarming event detecting and disposal processing approach for coal mine safety using wireless sensor network," *Int. J. Distrib. Sens. Netw.*, vol. 2012 (2012), 2012, Article ID 280576.

[25] W. Yang, H. Liusheng, and Y. Wei, "A novel real-time coal miner localization and tracking system based on self-organized sensor networks," *EURASIP J. Wireless Commun. Networking*, 2010, doi: 2010:142092.

[26] A. P. Sheth, K. Gomadam, and J. Lathem, "SA-REST: Semantically interoperable and easier-to-use services and mashups," *IEEE Internet Comput.*, vol. 11, no. 6, pp. 91–94, Nov.–Dec. 2007.

[27] W.-J. Qin, Q. Li, L.-M. Sun, H.-S. Zhu, and Y. Liu, "RestThing: A restfulweb service infrastructure for mash-up physical and web resource," in *Proc. IFIP 9th Int. Conf. Embedded Ubiquitous Comput.*, 2011, pp. 197–204.

[28] S. Duquennoy, G. Grimaud, and J. Vandewalle, "The web of things: Interconnecting devices with high usability and performance," in *Proc. Int. Conf. Embedded Softw. Syst.*, 2009, pp. 323–330.

[29] P. Schramm, E. Naroska, P. Resch, J. Platte, H. Linde, G. Stromberg, and T. Sturm, "A service gateway for networked sensor systems," *IEEE Pervasive Comput.*, vol. 3, no. 1, pp. 66–74, Jan.–Mar. 2004.

[30] W. Drytkiewicz, I. Radusch, S. Arbanowski, and R. P. Zeletin, "PREST: A REST-based protocol for pervasive systems," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Sens. Syst.*, 2004, pp. 340–348.

[31] R. Glitho, F. Khendek, N. Y. Othman, and S. Chebbine, "Web services-based architecture for the interactions between end-user applications and sink-less wireless sensor networks," *ACM SenSys*, pp. 865–869, 2007.

[32] F. Rosenberg, F. Curbera, M. Duftler, and R. Khalaf, "Composing RESTful services and collaborative workflows: A lightweight approach," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 24–31, Sep.-Oct. 2008.

[33] Y. K. Huang, A. -C. Pang, H. -P. Cheng, and W. -H. Zhuang, "Distributed throughput optimization for ZigBee cluster-tree networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 513–520, Mar. 2012.

[34] P. Cheong, K.-F. Chang, Y.-H. Lai, S.-K. Ho, I.-K. Sou, and K.-M. Tam, "A ZigBee-based wireless sensor network node for ultraviolet detection of flame," *IEEE Trans. Ind. Electron.*, vol. 58, no. 11, pp. 5271–5277, Nov. 2011.

[35] H. S. Ahn and W. N. Yu, "Environmental-adaptive RSSI-based indoor localization," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 4, pp. 626–633, Oct. 2009.

Bo Cheng received the Ph.D. degree in computer science from the University of Electronics Science and Technology of China, Chengdu, China, in 2006.

He is a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include service computing, Internet of Things, and multimedia communications.

Shuai Zhao is a Postdoctoral Fellow with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include Internet of Things and service computing.

Shangguang Wang received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

He is an Associate Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include service computing, cloud services.

Junliang Chen is a Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests are in the area of service creation technology.

Prof. Chen was elected as a member of the Chinese Academy of Science in 1991 and a member of the Chinese Academy of Engineering in 1994.