2017 International Conference on Applied Mechanics and Mechanical Automation (AMMA 2017)
ISBN: 978-1-60595-471-4

An Effective Device Integration Middleware in Prison IoT

Wei WEI*, Yang LIU, Huan-huan MAI and Wen-qiang ZHANG

Henan University of Technology, New and high-tech zone, Zhengzhou, China

*Corresponding author

Keywords: Middleware, Prison IoT, Intel atom, Device integration.

Abstract. Prison IoT (Internet of Things) is the application of IoT technology in prison environment. The monitoring platform is the core infrastructure in prison IoT. As more devices are integrated into prison internet to carry out some simplest tasks, the monitoring platform is required to communicate with more and more heterogeneous devices. The vast amount of devices increased the complexity in each stage of prison work flow, from deployment to operation. Prisons are experiencing more and more inconvenience and frustration as this situation become more complex. To overcome these problems, a middleware architecture is developed to help integrate heterogeneous surveillance devices from different vendors. As a result, prisons only focus on device function, thus help build more robust upper-level applications. The Java based on Intel Atom hardware implementation shows that devices can be easily introduced and managed, thus can improve prison work flow significantly..

Introduction

As prison Internet debut since 2010, more and more techniques and architectures in Internet of Things (IoT) have been applied in prison field. The initial objective is to enhance status monitoring of personnel inside and outside prison. With IoT techniques [1][2][3], we can build condition monitoring systems in prisons and integrate them into a global monitoring system based on the wireless sensor network and cloud computing technology. But devices in prison Internet are mostly designed, operated and maintained by different organizations, with different communication protocols and hardware standards. So integration is the first hard job.

As a restricted area, in prison environment, all prisoners, people and cars in and out are required to be monitored. In order to meet the above monitoring requirement, different kinds of sensors are needed, e.g., positioning devices, video cameras, perimeter monitor, are deployed and integrated into central platform. And these device need to provide control interface, e.g., video cam-eras could turn to given direction, and gateway could be closed on demand in case of emergency. The integration of various devices has become a greatly challenge in building prison monitoring systems. In this paper, we designed a middleware system to decrease the integration cost and promote the interoperation ability of the different devices. The goal is to provide a middleware to the national prison monitoring network that follow prison Internet standard.

Related Work

Many WIFI and RFID based solutions have been used for several years for monitoring or controlling purposes. To monitor the status of prisoners' hand-cuffs, H. Xu et al. at Jiangnan University developed a small wireless sensor network, in which sensors were integrated into handcuffs and were used to collect position and status information of prisoners [4]. To identify and locate prisoners in pre-established gateway, [5] developed an electronic watch which provides needed functions. In [6], S. Li provided a detailed illustration of a RFID application in public security domain of China. Real-time prisoner locating system based on RFID technology has been created [7].

For middleware, we distinguish two important functionalities in the integration of heterogeneous devices: (i) the software level middleware which tries to erase the 'soft' difference between devices and subsystems and provide universal routing and messaging services among them. (ii) the hardware level middle that is compatible with most of interfaces and facilitate hardware based real-time communication among devices.

Design of the Middleware

The software level middleware exploits semantics to support the integration and operational collaborations between different devices. In this paper, we de-scribe PIM - a middleware that seamlessly integrates devices with different prison surveillance technologies, manages message routing, data format and protocols translation, and provides event manage and data transfer services based on an operator's intent. By using devices' capabilities and locations, da-ta, PIM is able to present to the operator a standard and centralized view of devices that they can operate on, and give platform developer a unified inter-face to program on. In addition, PIM provide an interface standard that different vendors can follow and make their devices ready to be plugged into PIM with little effort. The software level architecture is shown in Figure 1.

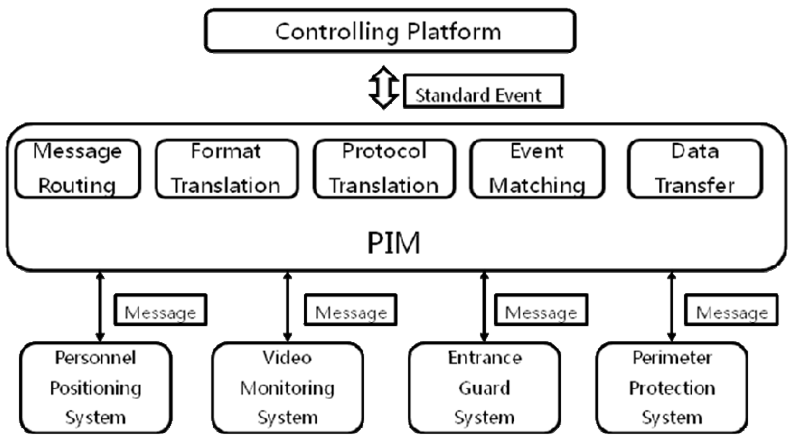

Figure 1. Software level of middleware.

Vendor Interface

Vendor Register Interface. The vendor register interface is the first part of interface in the vendor side of PIM and provides the main APIs for the vendor to expose their devices' capabilities. The registered devices' information is unified and stored, then it is used by center platform to organize global device view and publicize it to other devices.

Vendor devices' information is provided as device description schema, which contains three sections: the core description, capabilities description and the properties description. The core description part consists of device in-formation, such as device ID. The capabilities description describes what functions the device could provide, e.g., what order the device could accept and the output, the input and output data format and their corresponding software translation module, in the form of WSDL [8]. The properties description section lists all available features of the device, e.g., the attribute for a positioning device may include positioning type, such as RFID, WIFI, a sample device description shown in Figure 2.

Vendors Request Interface. The vendor request interface is used to obtain information about other devices by given conditions, e.g., neighboring area, then composes a request and finally executes the desired task. When the plat-form asks for task information, the middleware prioritizes the available tasks in progress and given a global review. For each specific device, the interface will find corresponding vendor agent module and translate data format and protocol grammar to a unified format, then translate it back to the format of requested device.

```
<rdf:RDF>
<PIM:Device rdf:ID="POS_A2_C11.1_R2">
<hasName>Positioning Handcuff</hasName>
<CAPS>
  <PosInfo>
   <inMIMEType>application/octet-stream</inMIMEType>
  </PoSInfo>
  </CAPS>
   <hasAttribute>
<NAttribute rdf:ID="PosType">
      <rdf:type rdf:resource="#PoSAttribute"/>
      <attributeValue>2</attributeValue>
      <modifier>1</modifier>
</NAttribute>
 </hasAttribute>
</PIM:Device>
</rdf:RDF>
```

Figure 2. Device description schema.

Middleware Core

The PIM core provides support for message delivery, contents transfer and event management with heterogeneous devices using different networking technologies. PIM is architected with three sub-layers.

Core Sub-layer. In a heterogeneous environment, we must provide interoperability for devices. It's the main aim of PIM. The layer is a shared infrastructure that provides lower-level functionalities that hide device heterogeneity from each other. It provides basic system-level functionalities, such as event management. Enabling device interoperability allows us to focus device abstraction and data aggregation in upper layers.

Platform Sub-layer. The platform sub-layer abstracts devices, aggregates de-vices and functions into a central place where operators and devices have a single view on them. One is Info Hierarchy which contains metadata of devices that have been discovered in Prison. It provides an interface for monitoring devices' states. And it also contains information regarding its location in the prison. Another is Management Hierarchy which contains management preferences, user access control policies on devices and contents. It is a central repository where components can obtain specific references for devices.

Service Sub-layer. The service sub-layer is designed for interactions with control platform and other devices. The design goals of this sub-layer are to provide (1) query interfaces for device information, (2) control interfaces for devices, (3) message delivery service among devices, (4) data transfer among devices, (5) event publish and subscribe.

Device Manager - Other parts in a prison use interfaces from the content man-ager. The content manager allows other parts to browse devices, query application level information based on certain criteria, access control policies. The device manager obtains the information from the lower layers.

Transfer Manager - the transfer manager take charge of message delivery and data transportation among devices. It provided connection oriented and connectionless communication across different low level hardware bus and proto-cols.

Event Manager - The event manager unifies different kind of event and publishes events to each subscriber. The original event will be filter according a preset logic and then unified into the delivery center. If delivered, the unified event will be translated back to device specific event. The logic is given as in Figure 3.

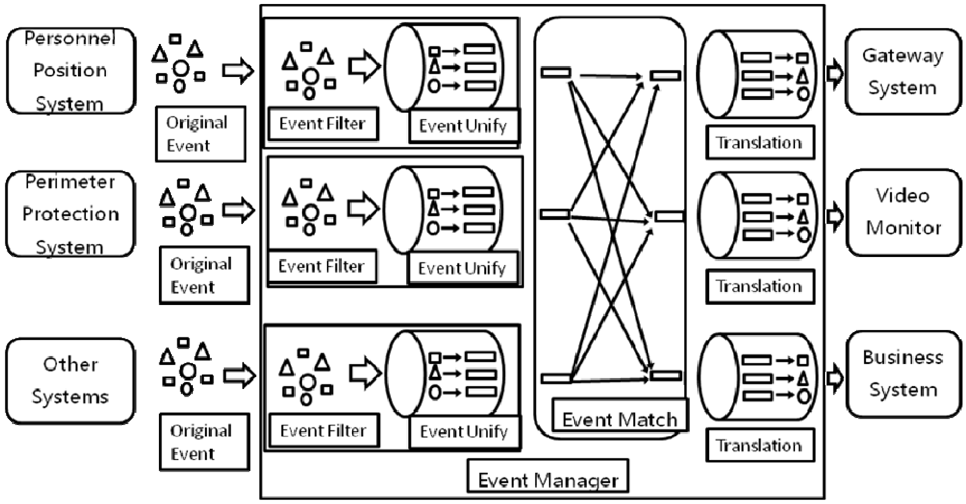

Figure 3. Event management process.

Prototype Implementation

We implemented a prototype of the PIM system in Java based on Intel Atom hardware platform. We used the Web Ontology Language (OWL) [9] and Re-source Description Framework (RDF) [10] to define the device and task description schemas. We chose OWL/RDF as they support elaborate description capabilities and make for flexible prototyping. We use two-level switching module for network packet switch, which support common network functions, like Vlan Mapping, QoS, ACL et al. To process these descriptions we used the Java Expert System Shell (JESS) [11] and implemented rules across our descriptions. Our implementation keeps the use of JESS to a minimum by as-signing only those operations to JESS that involve rule invocations. We implemented USB based plug-ins to support the UPNP protocol with some video cameras. We tested the prototype with a variety of vendor device available in our organization in order to verify our system in the presence of various de-vices and device types. The monitor UIs are given in Figure 4.

Figure 4. Middleware based perimeter monitor.

Conclusions and Future Work

The seamless integration of heterogeneous devices can greatly improve the utilization of the potential of data resources on existed prison monitoring in-formation systems. The unified integration and interoperability is a basic requirement for automatic search, retrieval and processing of grain storage sensor data. This paper is a step further towards the object from a way of middleware. The major contributions of our work are to create prison device standards and give a basic architecture of middleware in prison Internet.

In the near future, we are planning to investigate more type of devices and make PIM could support more heterogeneous devices and more specific requirement. It will be a further step towards national prison monitoring system.

Acknowledgement

This research was financially supported by the National Natural and Science Foundation of China (No.61472460,U1504607), Natural Science Research of the Education Department of Henan Province(17A520004,16A520006), Program for Innovative Research Team (in Science and Technology) in University of Henan Province (17IRTSTHN011), Plan of Nature Science Fundamental Research in Henan University of Technology (2014JCYJ04, 2015XTCX03), Key Laboratory of Grain Information Processing and Control (Henan University of Technology), Ministry of Education (KFJJ-2016-104), Doctor Foundation of Henan University of Technology (No. 2012BS011), Science and technology project of science and technology department of Henan province (No.172102110013).

References

[1] Katasonov, A., Kaykova, O., Khriyenko, O., Nikitin, S., Terziyan, V.: Smart Semantic Mid-dleware For The Internet Of Things. In: 5th International Conference on Informatics in Control Automation and Robotics, pp. 169-178. INSTICC, Madeira, Portugal (2008).

[2] Yan, L., Zhang, Y., Yang, L.T.: The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems. Auerbach Publications, FL (2008).

[3] Brock, D., Schuster, E.: On the Semantic Web of Things. In: Semantic Days 2006, Stavanger, Norway (2006).

[4] Xu,H., Fan, X.X., Wang, P., Peng, L.: Design of Things Network Node Used in Prison, Journal of Computer Research and Development 12(4), pp. 317-320 (2010).

[5] Tian, M., The Application of Intelligent Sensor Monitoring System in Prison Managment, In-ternet of Things Technologies 3, pp. 28-29 (2011).

[6] Li, G.S.: The Application of RFID Technology in Public Security Domain, China Security and Protection 4, pp. 41-44 (2011).

[7] Li, W.F.: The Real Time Personel Location System in Shushan Prison, Internet of Things Tehnologies 4, pp. 30-31 (2011).

[8] "Web Services Description Language (WSDL) 1.1", http://www.w3c.org/TR/wsdl

[9] Web Ontology Language (OWL). http://www.w3.org/TR/owl-features.

[10] Resource Description Framework (RDF). http://www.w3.org/RDF .

[11] Java Expert System Shell, http://herzberg.ca.sandia.gov/jess.