

# Design and Implementation of Monitoring Control System Based on Internet of Things

Ziyan Zhong<sup>1, a</sup>, Bo Cheng<sup>1, b</sup>, Shuai Zhao<sup>1, c</sup>

<sup>1</sup>State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

<sup>a</sup>13717812875@163.com, <sup>b</sup>chengbo@bupt.edu.cn, <sup>c</sup>zhaoshuai@bupt.edu.cn

**Keywords:** integrated visualization, intelligent warning, local storage, personalized module management

**Abstract.** The article takes integrated visualization of IoT (Internet of Things) monitoring data and friendly user interaction as research objects, relying on the IoT platform aggregated by lightweight business, to research on how to turn the heterogeneous data collected from various sensors to synthetic situation form, and how to provide a new scheme, which is characterized by personalized module management and intelligent warning based on local storage, to the user, thereby offering the user effective assistant decision information and friendly user experience to the maximum extent.

## Introduction

Nowadays, IoT technology has penetrated into every aspect of people's lives. Taking advantage of IoT technology to monitor objects provides people with great convenience and high efficiency. However, the realization of multiple monitor means is based on different sensor information. The traditional discrete monitoring systems are separate independent systems, which need to invest lots of manpower to manage and maintain the system. What's more, the system reliability is so poor that it would reduce the efficiency of operation and the level of emergency disposal, especially when realizing complex device linkage in the cases of emergencies and facing serious damages. Therefore, a mature IoT monitoring system requires not only overall design and accurate hardware configuration, but also an intuitive and comprehensive data presentation mode, to present the entire dynamic updating data view to users, which makes the most area of data share and integration from the whole monitoring system.

From the perspective of system optimization, a new generation of excellent Monitoring Control system would provides people with personalized and intelligent information service. Personalization here means that users can participate in the website design directly, through configuring the amount, position and size of display modules according to their own needs and preference. Intelligence refers to the initiative warning in the light of different degrees of danger of IoT monitoring system detected from devices' perception and judgment. In addition, the secure storage and convenient usage of data is an important issue of the application of Information Technology [1], especially in the monitoring system based on public network. Due to the network congestion and communication breakdown, data loss happens occasionally. Thus the local storage of data becomes more crucial. Taken together, the combination of these characteristics could boost the level of user experience, and is essential for users to analyze synthetic situation and make more rational and scientific decisions.

## System Architecture

**B/S-MVC Architecture.** The Monitoring Control system studied in this paper is implemented based on B/S-MVC architecture, which applies Web SSH (Struts+Spring+Hibernate) framework and the mode of separation the MVC (Model View Controller) on the server side. The browser side, namely the user interfaces that any user who has access to the system network can visit. The browser side mainly involves the real-time display and interaction control of each single device, and diversified warning display and trace approach of fusion information.

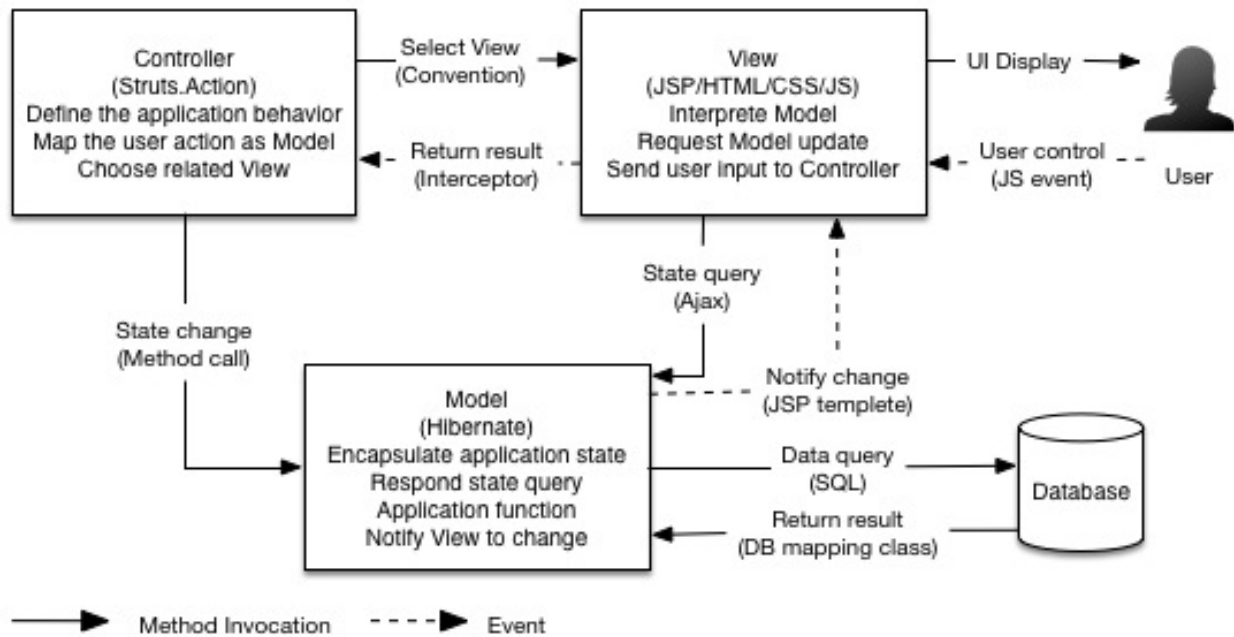


Fig.1. System B/S-MVC Architecture

The system is organized by MVC pattern—one kind of software design patterns. [2] It separates the storage, logic and presentation to achieve the high cohesion and low coupling between each component. If apply that pattern, the data operation, business logic and interface display part of the Web application could be modified separately and would not influence each other, which improves the reusability of the software component.

The interaction between each layer of the MVC architecture is shown as Figure 1, in which identified the technology and data interchange format applied in the Monitoring Control system. The meaning, function and the corresponding technology of Model, View and Controller are as follows:

- **Model:** Used to encapsulate data related to business logic and handling methods of data. In the system, it mainly embodies the mapping class of Hibernate database table and the corresponding DAO function.
- **View:** Used to render and display data. There are two approaches to render data. One is to directly render data from the server via JSP (Java Server Page) template language. Another is to realize the flicker-free partial refreshment by means of getting JSON format data through Ajax request from Model. The View part employ CSS to define the display style, and use JavaScript to create dynamic effect and the related user operation events.
- **Controller:** Used to organize Model and View, to control application flow, and to make corresponding treatment to events. The events here contain the interactive behaviors of the user, as well as the data change of Model layer. The Action defined by Struts framework plays the role of Controller in the system.

**Data Flow.** In the monitoring control system, the data flow mainly includes two directions: the upstream data flow (data upload) and the downstream data flow (send control command). The data upload flow chart is shown as figure 2.

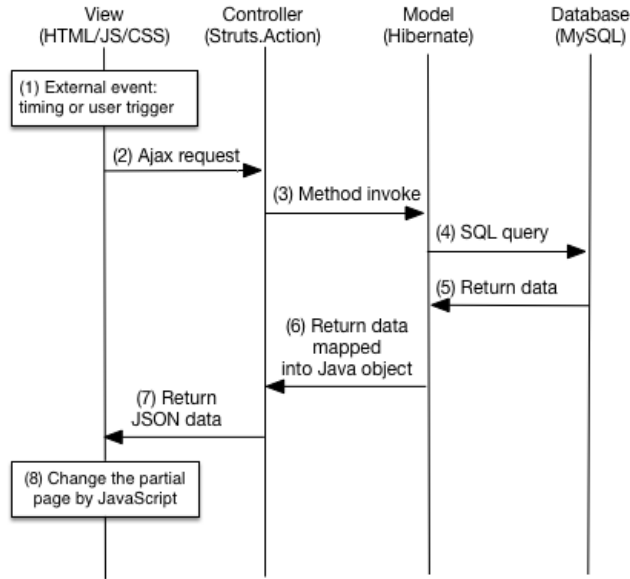


Fig.2. Data upload flow chart

If only take Web display & control system and the components of direct interaction into account, the data upload indicates the data flow that the data go through database and finally represent to the user by Web display and control system. The downstream flow refers to send users' operations to the system through user interface, and then send the command to the hardware equipment via publish-subscribe system and protocol stack framing. [3]

### Realization of the System

The function modules of the comprehensive monitoring system divided by business logic are listed in Figure 3.

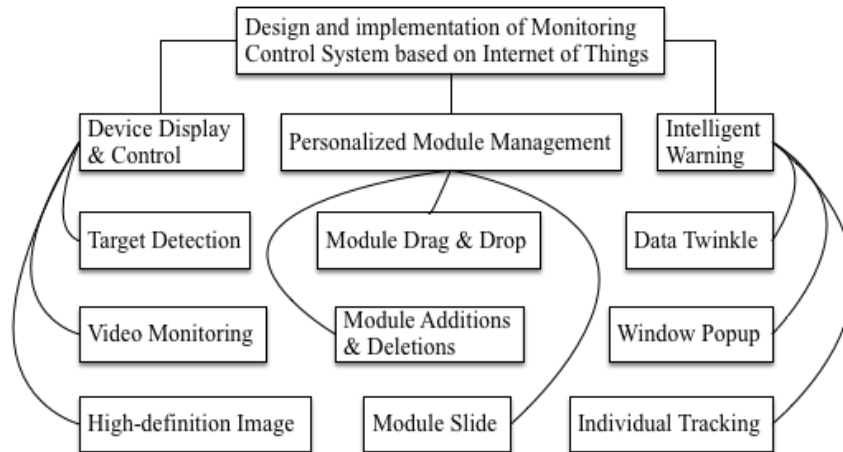


Fig.3. System functional module chart

The modules mainly contain display & control function of each device, personalized module management modules, and intelligent warning modules. The features and the difficulties of realizing the function modules and their sub-modules will be specified in the following sections.

**Device Display & Control.** The device display & control module involves target detection devices, video monitoring devices and high-definition image.

The display of target detection devices divided from bottom to top according to DOM tree mainly includes the rendering of background, reference line, historical route, plot, and the panel of the plot detailed information. The main function of dynamic curve drawing is realized through HTML5 Canvas. [4] The main idea of utilizing HTML5 Canvas to realize 2D dynamic effect is to achieve the graphical representations of motion model through JavaScript delay function, the partial or full scale

clearance, and the canvas redrawing. To render the plot, the DOM element corresponding to the plot icon would use absolute positioning, which is different from historical track that use relative positioning. The panel with detailed information is the invisible element ranking only under body layer when initialized. Once the mouse hover over the plot icon, the data in the panel would be dynamic updated, with the float position being settled down, and the non-display attribute being cancelled.

The display and control functions of video monitoring devices can be realized via ActiveX, applet or other video plug-ins. ActiveX plug-in is equivalent to a client application integrated in the browser. It adopts desktop software programming language (i.e. MFC, C#, etc.), and could step out of the HTTP protocol limit, communicate with device more flexible and direct, and realize real-time video display efficiently as well as take up less system resources. [5] Similarly, Applet plug-in is compiled via Java programming language. It has good compatibility in that it not only compatible with IE browser.

The display and control functions of high-definition image devices involve real-time image display function and historical image display function. Both of them are based on Ajax short polling to realize the picture updating. For the purpose of saving the Ajax request, and reducing the network load, the update strategy we can adopt is that, once a picture need to be updated, move and insert the last but one picture into the historical area. On account of automatic caching mechanism provided by browser, this kind of movement wouldn't bring about redundant server request.

**Personalized Module Management.** Because of too much devices accessed into the monitoring control system, the system contains only a few key modules on initialization to achieve the priority watch of the key area. The administrator can drag and reposition the monitoring module of key area or key point, add or delete the module, and slide around the module smoothly according to the need.

The flow chart of drag-and-drop and reposition the grid is shown as figure 4.

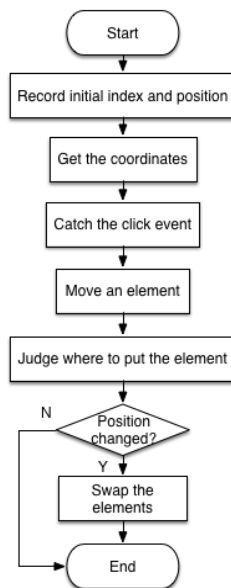


Fig.4. Flow chart of drag-and-drop and reposition the grid

To realize the function, we ought to use array to record the initial index and the position information of each monitoring module (i.e. corresponding div element) at first. Secondly, we get the mouse's coordinates to obtain the mouse movement information. After that, we catch the mouse click event, and then we'll know when the mouse would be clicked and when it released. [6] If we skip this step, we will never know the hover action during drag-and-drop, which will be annoying and counterintuitive. Now we really get down to business. We move an element and know how to capture the movement. We define `item.style.position = 'absolute'`. Every operation changes the left and top coordinates, it moved. When mouse released, judge the index of position that the element wants to reach, and save the index at the same time. Find the corresponding index in the array by means of

using attribute selector. [7] If two indexes are different, swap the index of array and reposition the element, as well as shift the subsequent elements one module width to the right.

For the add module function, the realization processes are as follows:

- (1) Click the add button.
- (2) One panel popup, realized by toggle function. The popup panel has listed all of the device monitoring module options. Choose the modules required.
- (3) If the module exists, popup the alert message “The monitoring module is already existing.” Otherwise, add the modules behind, which means append the elements of the module into the HTML body.

Analogously, in order to delete a module, we need to add a delete icon at the time when mouse hover over the module by means of adding mouseover and mouseout EventListener. After that, if user selects a monitoring module to delete, and confirm that operation, then the corresponding nodes will be deleted from DOM, and the subsequent modules will be autocompleted.

The function of sliding the module to left or right is of equal importance. In that case, user can glance over all items conveniently. We add previous and next icon to each row just like the delete icon. Once we click the next button, the system would judge whether the module is animated. If not, the index recorded as the move step of the row will plus one. Then if the index is equal or less than the module numbers of the row, the position of the whole row will move one module width to the right. Otherwise, the row will be reset and the index will be set s the module numbers of the row. Vice versa, he process of slide to the left (i.e. press the previous button) do the opposite thing.

**Intelligent warning function.** A friendly monitoring control system can make full use of the detected information collected by all kinds of sensor equipment. It arranges the symptom information according to certain rules or protocols, and matches the information with the given threshold value, so as to make the fault diagnosis decision for the user.

Once the system monitored the device alarm, the event display area would update the newest information with text twinkling hint, accompany with alarm sounds and the shake of the related monitoring modules. After that, a window embodies the corresponding modules will popup. The administrator could slide the module and double click any one to zoom in, so that he can observe the individual detailed monitoring information. The flow chart of intelligent warning is shown as figure 5.

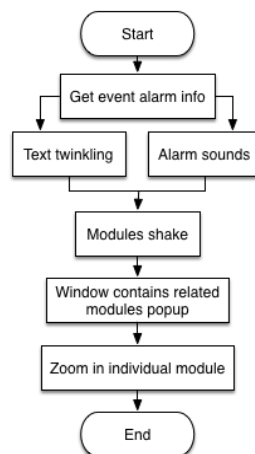


Fig.5. Flow chart of intelligent warning

The event information is gained from the database. The text twinkling function is realized by JavaScript setInterval function. We set the blink color as yellow alternates with red, and the blink duration as 2,000 millisecond.

As for the shake effect, we apply the csshake plug-in to realize a variety of shakes. The critical code is shown in figure 6.

```

<link rel="stylesheet" type="text/css" href="csshake.min.css">
<div class="shake-hard">
  
```

Fig.6. Code of shake effect

The realization of the window popup is much more complex. After getting the event information, the popup window attribute turns to display. We need to overlay two layers on the existing interface. One transparent layer masking, and one pop window contains the corresponding monitoring modules by setting the opacity in the css file and use CSS3 keyframes technology respectively.

Last but not least, the zoom in effect ought to apply double click function and set window.location.href attribute so as to link to the individual module monitoring subsystem.

**Application scenarios.** The system is broadly applicable to the cases where the administrator intends to monitor the entire IoT system.

Take a campus application scenario as an example. The level 1 target detection devices can be distributed outside the external wall, at the campus gateway, at every entrance of the important buildings, and so on. The devices accessed to the IoT system could be heat-sensing equipment, electrical fence, entrance guard system, or any other alarm devices.

The monitoring range of level 2 video monitoring devices covers the main roads on campus, active recreation area, and in-building monitor. On the whole, we need up to a hundred surveillance cameras, which are all fixed point monitoring in teaching buildings, laboratory buildings, canteens, dormitories and any other people concentrated area. The moving point monitoring outside the building need pan-and-tilt rotary head to monitor the main street and gateways.

The level 3 high-definition image equipment include HD camera, 3G infrared camera, laser monitor and so on. Level 3 devices sometimes require linkage to Level 2 devices. The system screenshots is shown as figure 7.

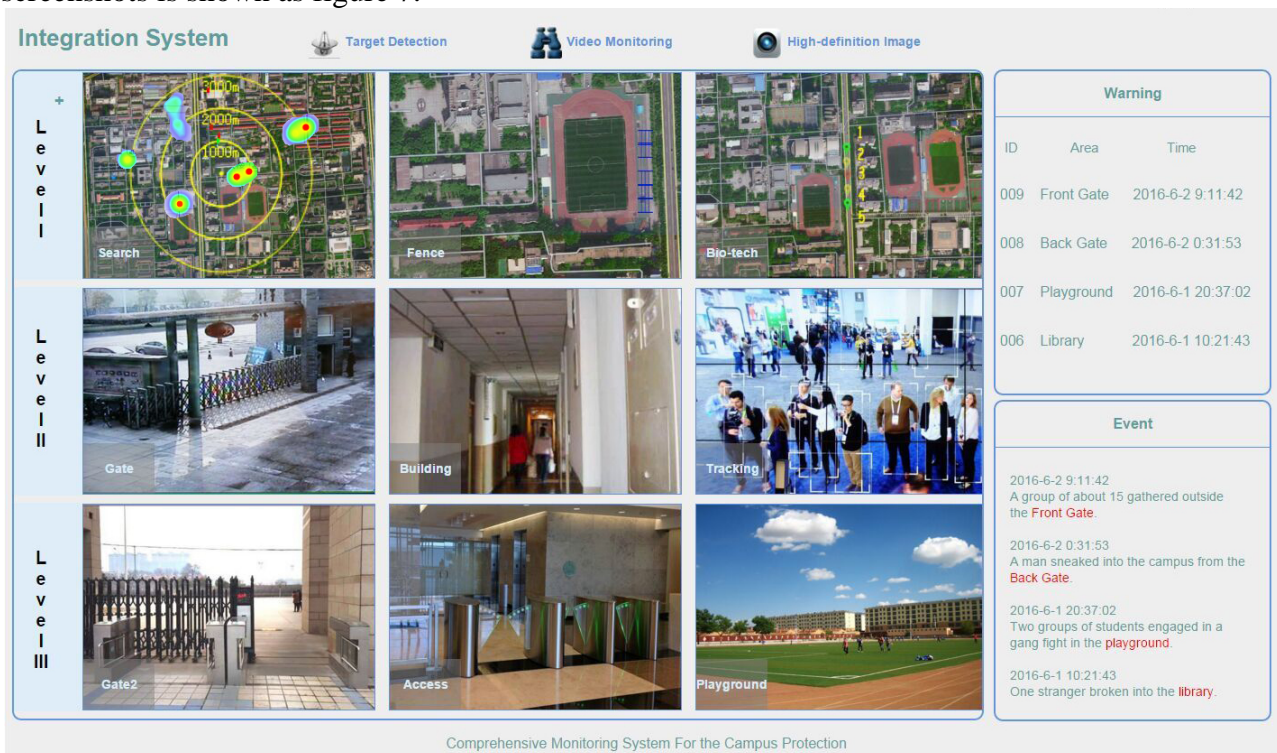


Fig.7. The system screenshots

The integration system includes three levels of device display and control. Each row can be slide to left or right to observe all the important items smoothly. Furthermore, we can click the add icon to add a module or hover over one module to delete the unnecessary one. The right side shows the warning and event details of when and where what occurred. In that case, the administrator can clearly be informed things happened. In general, the system characteristic of personalized modules management and intelligent warning works very well.

## Secure Storage

In order to realize the website personalization, we need to remember the configuration information of the user. For that reason, SessionStorage, which valid until session ends is not suitable for the system. In addition, considering we only need the configuration information during browser display, there's no need for verification between server side and client side, so it's also unnecessary to apply Cookie that would cause internal storage waste and influence the performance.

Based on the above consideration and test, the system adopts HTML5 LocalStorage as the client side storage mode. The configuration parameters, individual monitoring module page number, alarm event object, and modules' sequence index object all use LocalStorage technology. In this way, when the user visit the website again, the modules displayed are custom defined instead of the original state, not influenced by page refreshing. Other variables would be taken out from local computer for the front-end application use.

A set of LocalStorage read-write operation codes is as figure 8.

```
range = (window.localStorage && window.localStorage.getItem('range')) || 1000;
if(window.localStorage){
    window.localStorage.setItem('range', range);
}
```

Fig.8. LocalStorage read-write operation

What needs to be paid attention that localStorage is saved as key-value form. When we want to save objects, the parse and stringify method provided by JSON object can be used to transfer other data types into string, and then store into localStorage. [8]

## Summary

This article offers an implementation plan for the monitoring control system based on Internet of Things. It designed and constructed an integrated system possessed with functions of device display and control, personalized module management, and intelligent warning. Furthermore, taken a campus security and protection scenario as an example, the plan can be perfectly applied at lower cost. The system plays an important role for keeping campus safety, and also has certain significance to other construction of area security and protection.

## References

- [1] Zhang Yuliang, Du Keming, Sun Zhongfu, Man Qingkui, Zhao Wei, and Li Hongliang, "Integrated Application of Local Data Storage Technology in Agricultural Remote Monitoring System", Agriculture Network Information, 2010:doi:10.3969/j.issn.1672-6251.2010.04.003.
- [2] Information on <https://en.wikipedia.org/wiki/Model-view-controller>
- [3] Margara A, Cugola G. High Performance Publish-Subscribe Matching Using Parallel Hardware[J]. IEEE Transactions on Parallel & Distributed Systems, 2013, 25(1):126-135.
- [4] Grady M. Functional Programming Using Javascript And The Html5 Canvas Element[J]. Journal of Computing Sciences in Colleges, 2010, 26:97-105.
- [5] Information on [https://msdn.microsoft.com/en-us/library/aa751968\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa751968(v=vs.85).aspx)
- [6] Information on <http://blog.csdn.net/anthonywanted/article/details/3525813>
- [7] Information on <http://jingyan.baidu.com/article/ed15cb1b7a051e1be36981df.html>
- [8] Information on <http://my.oschina.net/crazymus/blog/371757?fromerr=Aadiq6bt>