EMSS Week 6

ARCHITECTURE

Amol Shandilya, Minghui Jin, Yanyan Jiang

The Booch Group

Architectural Styles

Model-Driven and Model-View-Controller

The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction and passes these interactions to the View and the Model.

1. View and Model is separate, so the system can change the view code without re-compiling the model and controller code. For example, when EMSS changes the power supply or adjusts an electricity ratio, the system logic section is the only one module needs to change.

   The model is self-contained and separated from the controller and view, making it easy to change the application's data and execute rules. Just changing the model can replace the database. Once the model is implemented correctly, the view can display them correctly, whether the data comes from the database or other servers. Since the three components of the application using MVC are independent of each other, changing one of them does not affect the other two, so a good loosely coupled component can be constructed based on this design idea.

2. MVC mode allows the use of a variety of different styles of view to access the same server-side code, because multiple views can share a model which includes any WEB side, App and PC side. For example, user can use any laptop or mobile-phone to check the status and change the model through the same model. The data returned by the model is not formatted, so the same component can be used by different interfaces. For example, the electricity power company changes the data, or the user data updates, and these only change the view to achieve the way, while the controller and model without any changes. Since data and business rules have been separated from the presentation layer, the reuse code can be maximized. The model also has the function of state management and data persistence processing.

3. Because different modules perform different duties, different applications have some of the same characteristics, which is conducive to engineering and tool management program code. The controller also provides the advantage that it can be used to connect different models and views to meet the needs of the user, so that the controller can provide a powerful means for constructing the application. Given some reusable models and views, the controller can select proper model for processing according to the needs of the user, and then select the view to display the results to the user.

Event Driven and Observe and React Pattern:

EMSS is a real-time system which should keep monitoring changes both in the system and the house, such as the battery level, the status of the system linked devices, and react to these changes in certain time according to the system timing analysis.

1. System shall observe that solar power is available or not, and react to different battery levels. The system settings about the battery level are as follows.
   1) When the battery level is under 80% at daytime, all the solar power is going to charge the battery until its electric quantity is up to 80%.
   2) When the battery level is in the range of 80% and 100% at daytime, then the solar power is firstly going to charge the battery, if there is any additional power left, then it goes to charge the working devices in the house.
   3) When the battery is full charged at daytime, then the solar power will charge the working devices in the house firstly. The extra power will be sold to the electricity company.
   4) In the case of nighttime power consumption, the system shall automatically change the method of power supply to solar power. Once the battery stored power is empty, then the system shall automatically change to electricity grid power supply.

2. System shall observe the device status and the whole power usage. It shall be able to send usage report to the user as asked at any time.
   1) The system shall keep accepting the status information of system linked devices, and keep these information in the system DB, as well as sending messages to the user to notify the status changes.
   2) The system shall monitor the real-time power consumption, and keep this information in the system DB. Once the user asks for a usage report, then the system goes to its DB, extracts the related information, and send a report to the user.

Advantages of this pattern:
In this pattern, the value of sensors in the system is observed and, if within expected ranges, no action is taken. If these sensors indicate some change has occurred, then action, such as changing the power supply, is taken.

What's more, with timing analysis, this pattern can react to changes in short time without affecting the user experience and system performance.

Architectural Pattern

For our project, we will be using the master-slave architecture. The master-slave is used where guaranteed interaction response times are required. It is often used in real-time systems. In our case, the sensors will feed the data to the central control unit in real time. This is important since the various features like the power consumption, switching of devices, and battery status are heavily dependent on the real-time functionality of the system.

One can also argue that the system uses two-tier client-server architecture since the central unit interacts with subunits like microcontrollers and battery system which have sensors and hardware units as their slaves.