Integrated Device Control System using Google Maps

Takahiro KOITA¹, Kenta UEDA² and Kenya SATO¹

¹ Department of Information Systems Design, Doshisha University, Japan
² Intel, Japan

Abstract

This paper proposes an integrated device control system called Control Architecture for Networked Devices in Location Environments (CANDLE), which controls devices with Google Maps. Google Maps provides a global map interface and flexible APIs.

The main motivation for this study is to provide a search and control system for physically distributed devices using Google Maps. To achieve this, the requirements of the system are (1) to coordinate device information with map information on Google Maps and (2) all distributed information has to be integrated on one interface. Device information is a set of information to control devices on an interface of Google Maps.

CANDLE is an integrated device control system using Google Maps designed to meet these requirements. Using CANDLE, a user can control physically distributed devices through a web browser. In CANDLE, all device information is integrated into one map. CANDLE enables device control, searches, and display of information on distributed devices on an interface of Google Maps through a web browser. By using a web browser, a user can control devices from anywhere even if the devices are distributed globally. Thus, CANDLE is quite useful for distributed device control systems such as security, healthcare, and global monitoring systems.

Keywords: Google Maps, Ubiquitous Network, Ajax, Distributed Device Control

1. INTRODUCTION

The APIs for Google Maps [1] and Yahoo Maps [2] have been opened by and the companies to allow users to take advantage of maps on the Internet. These APIs can be easily applied to many systems or applications with JavaScript.

The main motivation for this study is to provide a search and control system for physically distributed devices using Google Maps. Thus, we propose an integrated device control system called Control Architecture for Networked Devices in Location Environments (CANDLE), which adopts device information in addition to text, picture and map information as content information. Device information is defined as the control screen, the name, and the place for the device concerned. CANDLE enables users to control devices using CANDLE’s functions for device information and device operation. CANDLE consists of four parts: a client, device servers, an interface server and devices. The device server manages the device information. Managing the device information involves collecting and updating the information on the device server. The interface server provides Google Maps and the additional proposed functions using the API of Google Maps.

In past few years, many studies have developed information systems using such map information. However, the focus of these studies was only sharing or showing content information such as text, pictures and movies on a map. Using CANDLE, users can control physically distributed devices through a web browser. In CANDLE, device information is written as hierarchical XML data. The XML data structure is designed to handle a building as a unit. Using hierarchical XML data structure, CANDLE can provide the hierarchical structure of a device from within a building. Using the XML data structure of CANDLE, a user can search among a number of devices in a building.

CANDLE also enables asynchronous communication of the user interface. However, existing web application functions adopt synchronous communication with the web browser and web server. Every time a user changes web pages the web browser communicates with the web server. Thus, classic web applications are not suitable for a user interface that requires asynchronous response. In contrast to classic web applications, CANDLE uses Ajax, and the web browser can communicate with the web server asynchronously.
2. CANDLE'S ARCHITECTURE

2.1 CANDLE

CANDLE can control many devices on one page. An overview of CANDLE is shown in Fig. 1.
Client (A): A user uses a web browser to access an interface server.
Interface server (B): An interface server provides JavaScript code of output device information and Google Maps for users, and gathers device information from device servers. This information is used to provide a device control screen for users.
Device server (C): A device server maintains and manages device information.
Devices (D): Devices connect to a device server and are controlled through the network.

2.2 Control sequence

The basic control sequence of CANDLE is shown in Fig. 2. The sequence is made up of the following seven steps.
(1) A user accesses an interface server using a web browser and gets map and device information from a device server. The web browser shows map and device information.
(2) The user selects one device displayed on the map.
(3) The web browser accesses the device server of the selected device to control.
(4) The web browser gets device information and a device control screen, and displays them to the user.
(5) The user controls the device and executes an action. The web browser sends a request to the device server with asynchronous communication.
(6) The device server accepts the request and controls the device. The results (responses) of controlling the device are sent to the web browser by the device server.
(7) The web browser shows results on the screen.

2.3 Device server

A device server manages connected devices and works as a web server. Existing devices do not have the function of a web server. Thus, CANDLE uses device servers to control the devices through a web browser. If the devices have the function of a web server, CANDLE may not use a device server. However, if the devices exist in the same building, CANDLE uses the device server. The device server can use many devices and has device information on the following.
- Device information on a device server is provided as XML data.
- CANDLE needs to display device information data: a device control page, a device icon, etc.
XML data is written in the style described in subsection 2.5 and is put on a device server in advance. The XML data style is a tree structure. The XML data is updated with device information and rewritten XML data by the device server. Thus, users can get the latest device information when accessing CANDLE.

2.4 Interface server

An interface server supplies device information links to users. The interface server has the following functions.
- Map supply to users
- XML parses
- Device registration
- Device information link supply for users
The details of these functions are explained in the following subsections.

2.4.1 Map supply to user's function: Map supply function supplies an interface of Google Maps for users. Map supply function provides to show points of devices on Google Maps, to change the focus point according to the user selection and to communicate with other functions.
2.4.2 XML parse function: The XML parse function gets the XML data from the device server and extracts useful information from maps to supply to users. The XML data structure will be described in subsection 2.5.

2.4.3 Device register function: Device information is registered at the interface server by this function and registered information includes device longitude, device latitude, device name, device server’s URL and XML data file name (data.xml, data.php, etc.). CANDLE uses registered information and enables searching for devices. The interface server can register many device servers’ XML data.

2.5 XML data structures

In CANDLE, we define the XML data structure. The interface server can parse the XML data. The XML data structure is designed to handle a building as a unit. Using hierarchical XML data structure, place information can be described not only by geographical information (longitude and latitude) but also by relative information to a building or a floor. Thus, a user can search for a device within a building even if the geographical information of devices is the same. A user can also use the interface of Google Maps and can search a device in a wide area and inside a building. The XML data structure is shown in Fig. 3. In the figure, a parent node is pictured. The latter child node, attribute and content are text. The root node is named the server. The server node has a child node named place. The place node has a child node named floor. The floor node has a child node named floor_info and information on device existence. Floor_info has a child node named device and information on the floor image. The device node has device information on id, name, controller and picture.

3. IMPLEMENTATION OF SYSTEM

3.1 Environment of implementation

The interface server’s implementation of the web server is Apache-2.0.54, the program language is Ruby-1.8.2 and the database management system is MySQL-4.1.11. The device server’s implementation of the web server is Tomcat-4.0.6 and the program language is Java-1.5.0_06.

3.2 Description of XML

Figure 4 shows an example of the description of XML data. The XML data has the following nodes: Place node, Floor node, Floor_info node and Device node. Each node has information described as follows.

- Place node: describes building information
  - Location node: longitude and latitude
  - Name node: building name
  - Description node: building description
- Floor node: describes presence information of device
- Floor_info node: describes each floor’s information
  - Floor_img node: floor picture
- Device node: describes device information
  - Device_id node: device id
  - Device_name node: device name
  - Device_controller node: device control URL
  - Device_image node: presence information of device picture URL

This XML data structure has three rules.
1. The floor_info node can handle as many floors as there are in a building.
2. The device node can also handle as many devices as there are in a building.
are on a floor.

3. The device node, the child node of the floor_info node, can handle device presence information.

```xml
<device floor="1">
  <device_id>Ke115Light</device_id>
  <device_name>KE115 Light</device_name>
  <device_controller display="html">
    cfl/ke115.htm
  </device_controller>
  <device_image>img/light.png</device_image>
  <device_location left="260" top="90"/>
</device>

<device floor="1">
  <device_id>ke115Camera</device_id>
  <device_name>Web Camera</device_name>
  <device_controller win="window">
    http://119.22.96.1?page=Single
  </device_controller>
  <device_image>img/camera.png</device_image>
  <device_location left="260" top="55"/>
</device>

<device floor="1">
  <device_id>Ke115Light</device_id>
  <device_name>KE115 Light</device_name>
  <device_controller display="html">
    cfl/ke115.htm
  </device_controller>
  <device_image>img/light.png</device_image>
  <device_location left="260" top="90"/>
</device>
</floor_info>

<floor_info floor_name="second floor" info="0">
  <floor_image>img/ke2.png</floor_image>
</floor_info>

<floor_info floor_name="third floor" info="0">
  <floor_image>img/ke3.png</floor_image>
</floor_info>

<floor_info floor_name="fourth floor" info="0">
  <floor_image>img/ke4.png</floor_image>
</floor_info>
</floor>
</place>
</server>
```

Fig. 4 Example description of XML data

### 3.3 System internal constituent

CANDLE’s internal constituent is shown in Fig. 5. U-MAP (Ubiquitous-MAP API) [3] is our basic API for CANDLE. U-MAP functions as an interface server of the map supply function. U-MAP consists of four objects: mapper, parser, controller, and switcher. The mapper supplies user device information on an interface of Google Maps and extends the Google Maps API. The parser parses XML data and hands over information on the mapper. The switcher changes screen device information and device controls. The controller controls devices. Users call the U-MAP, which uses each object to supply each function to users.

3.3.1 Device registration function of actual environment:
The device registration screen is shown in Fig. 6. A finger icon appears when a user clicks on a map and inserts longitude and latitude into the registration form. Other registration information is the building name, the device server’s URL and the XML data file name. By using the device registration function, device information is registered to the interface server.

3.3.2 Getting the XML data function of actual environment:
Getting the XML data function shows a marker on Google Maps. The marker indicates that a device exists at the marker place on Google Maps. When a user clicks on a marker (Fig.7), Google Maps shows device information in an info window with the mapper. After clicking the marker, the info window displays the message “Loading…” and the mapper sends a request to the device server. The device server makes a response to the web browser. The web browser gets a response and call for the switcher. The switcher changes the window shown in Fig. 7. In this figure, the window shows floor information to control changed by the switcher.

3.3.3 Device control function of actual environment:
When a user selects a device on the browser and clicks the “send” button, the controller sends a control request to the device server. The device server receives the request, controls the device, and sends a result message to the web browser. When it receives the result message, the web browser displays the result on Google Maps.
4. RELATED WORKS

This section discusses related works focusing on Google Maps or map information on the Internet.

First, MapWiki [5] is a system that expresses wiki content on a common map. MapWiki enables writing and updating the content to communicate. An example of implementation is GMap Wiki that uses Google Maps and RDF for data communication. MapWiki manages content information at a central data server. However, CANDLE uses not only content information but also device information. CANDLE manages device information in a distributed manner. This can be suitable as a device control system when the number of target devices increases.

Second, studies of distributed map information using DHT (distributed hash table) [5] and an information navigation system [6] have been conducted. The DHT system can shorten search time of content. Moreover, the DHT system can share the contents on maps with a fixed keyword or fixed key information. Thus, the system is not suitable to search for picture contents or to search for keywords. The information navigation system searches are distributed on a network database and the Web, and a map is selected from many available maps. These systems treat only information related to maps, text and pictures, but CANDLE treats information not only on maps, text, and pictures but on device information as well. Using hierarchical XML data structure and the design, place information can be described not only by geographical information (longitude and latitude) but also by relative information to a building or a floor. Thus, a user can search for a device using an interface of the hierarchical structure in case of inner within a building even if the geographical information of devices is the same.

5. CONCLUSION

This paper proposes an integrated device control system, CANDLE, which controls devices with an interface of Google Maps. Using this system, users can control devices on an interface of Google Maps with a web browser. Therefore, the system registers devices to the map, and users can search for and control devices with Google Maps. Additionally, CANDLE uses XML data that describes device information and can display the position of a device in a layered structure.

We are planning to implement the system to extend the search function to a query on device state. In addition, device servers will update device information and reflect XML data. Therefore, the device server dynamically updates device information on the longitude and latitude or its status when the device moves. Thus, users are able to search with a query on the most recent device information and state.

6. REFERENCES