International Journal of Smart Home
Vol. 9, No. 12, (2015), pp. 245-254
http://dx.doi.org/10.14257/ijsh.2015.9.12.25

Design and Implementation of a TV Schedule Editor for Internet TV Systems

Gyeyoung Lee[1], Jaegeol Yim[1, corresponding author]

[1]*Dept. of Computer Engineering Dongguk University, Gyeongju, Gyeongbuk, Korea*
{yim, lky}@dongguk.ac.kr

Abstract

This paper introduces our design and implementation of an editor with which Internet TV staff can create and modify TV schedules. This editor allows authorized staff to retrieve contents by accessing the content management system and to send a created schedule to the broadcast management system. We apply the Microsoft Solution Framework/Component Design (MSF/CD) principle in our development. Our design of database and classes and experimental results are discussed in this paper. Most of local governments are running their own Internet TV systems in order to provide regional news and information of local products, festivals, events, weather, and points of interest to citizens and tourists. With our editor TV schedules can be efficiently created.

Keywords*: Internet TV; TV Schedule; Editor; Electronic Program Guide; Component Design.*

1. Introduction

Internet television is the digital distribution of television content via the Internet. One of the apparent advantages of Internet TV is that any user with an IP device can watch it anywhere and anytime as long as the user can access the Internet [1].

The authors of [2] claimed that the use of the Web in delivering warning messages to the people of expected natural disasters such as a tsunami, earthquake, eruption of a volcano, flood, storm, and so on demonstrated the usefulness of Web sites in preparing for a natural disaster. From this claim we can infer that the Web is efficient for delivering government announcements to the people.

There are many local governments running their own internet TV systems. Pohang Internet Broadcasting Station run by Pohang city, Mint TV run by Incheon city, GBTV, Daejeon TV, and so on are a few example local governments' internet TV systems. Public information such as regional news, travel guide, and information about culture, weather, agriculture, and festivals is the only subject most of these Internet TV web sites are concern.

The only ultimate goal of all projects performed by local governments should be upgrading quality of citizens' life. One of the most important factors that determines quality of life could be income. However, none of existing local governments' TVs provides shopping service. Since pc monitors are large enough to display the TV screen and other menus, there is no reason to exclude content to promoting regional products. This paper proposes an internet TV for a local government in which shopping regional products is integrated.

ISSN: 1975-4094 IJSH

2. Related Works

There are many published research results regarding development of Internet TV Web sites. The authors of [3] proposed a mechanism to enhance Web Site Design Method (WSDM) by applying Model driven Architecture technique. WSDM follows the audience-driven design philosophy consisting of the following phases [4]:

1) Mission statement specification: the purposes of the site are specified.

2) Audience modeling: targeted users are identified and classified.

3) Conceptual design: Conceptual specification requirements of the web application is described.

4) Implementation design: This phase contains three sub phases:

- Site structure design: conceptual structure of web application is grouped and mapped onto pages
- Presentation design: the layout of web pages are specified.
- Logical data design: this sub phase is used only if the web application need to process data from external sources (e.g. Databases).

5) Implementation: generate actual implementation automatically.

The authors of [5] introduced their design of a Web Based Instruction (WBI). WBI promotes distance learning economically, enables learners to attend classes at their homes or offices, and provides delivery medium, content provider and subject matter in one package for the convenience of use. The authors of [6] presented a simple web search engine for indexing and searching html documents using python programming language. The authors of [7] presented a Ubiquitous Web Services Framework which aims to hide the heterogeneity of hardware, software, data formats and communication protocols that is present in today's pervasive environments. The authors of [8] proposed an Open Contents Web-Based to be implemented using Protocol of REST Web Service.

The authors of [9] developed a new web application for realization of interactive teaching and learning using the advantages of wireless networks and personalized devices. The authors of [10] introduced a real-time video communication system that uses the HTTP/Web Socket and RTC API via a Web server. Using the system, users can make real-time video communication through Web browsers. The authors of [11] proposed a new view synthesis technique for coding of multi-view color and depth data in arbitrary camera arrangements.

3. Design of the Editor

Our user management system allows the system manager to register, delete, update, and retrieve user information. It also classifies users into groups depending on their roles and assigns resource access levels to groups. It is a unified ID management system in that a user may not log in again in order to access another sub-system. The system requirements are itemized as follows:

A schedule editor should allow users to create, delete, and update channels. Therefore, our design of the user interface of the channel management component has a Registration button for channel creation and an onAir button to disable a channel as shown in Figure 1. In the figure, we can find a combo box labeled onAir. From this box, we can designate onAir, offAir, or both. In this particular case, onAir is designated and all channels listed in the figure are active ones. By clicking an edit button in this figure, we can edit the TV schedule of the channel associated with the button. A typical user interface we can meet by clicking an edit button is shown in Figure 2.

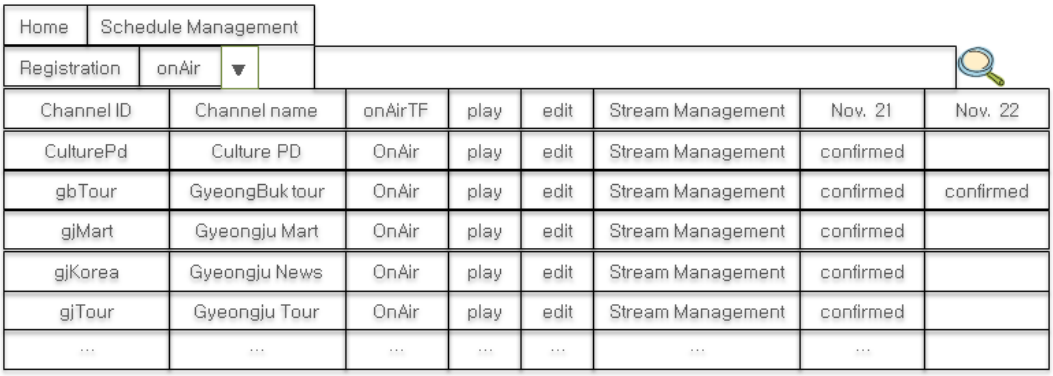

Figure 1. Our User Interface for the Channel Management Component

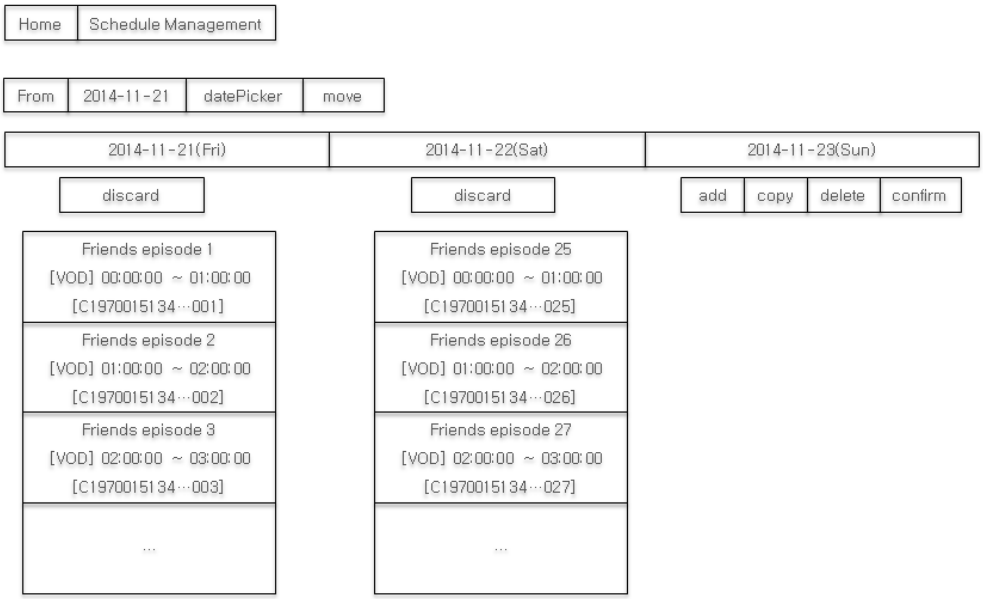

Figure 2. Our Design of the Main Page of the TV Schedule Editor

Using the datePicker button, we can designate a date, 2014-11-21, in this case. Then TV schedules for the days from the designated date, 2014-11-21, 2014-11-22, and so on in this case, are displayed on the window. We can discard existing schedules by clicking the discard button. TV schedule for 2014-11-23 is not created yet. We can add a content item to the TV schedule by clicking the add button. We can select content items and copy to another place of another or the same TV schedule. We can also select any number of content items in the TV schedule and delete them. Once a TV schedule is completed, we can click the confirm button and send the schedule to the stream management component. When the add button is clicked, a user interface similar to Figure 3 is popped up.

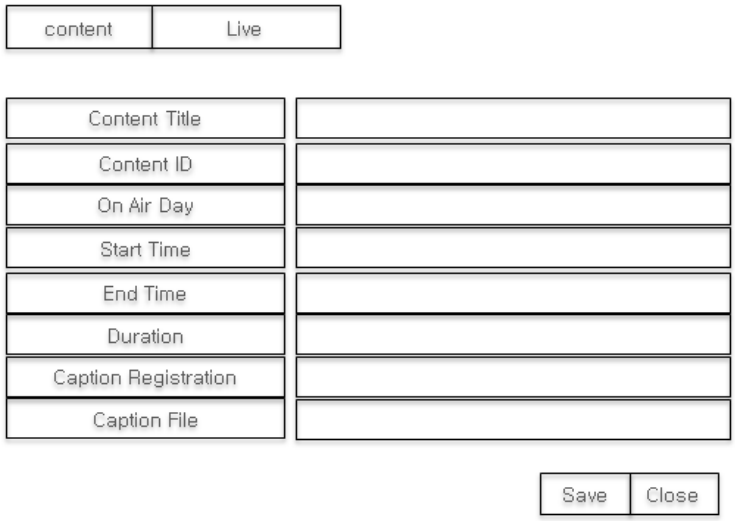

Figure 3. Our Design of the User Interface for the Add Menu

From Figure 3, we can see an element of a TV schedule can be either an archived file or a live stream. We can retrieve a content item through the content management system. After searching for a content item, we can click the save button in order to return to the edit-schedule page.

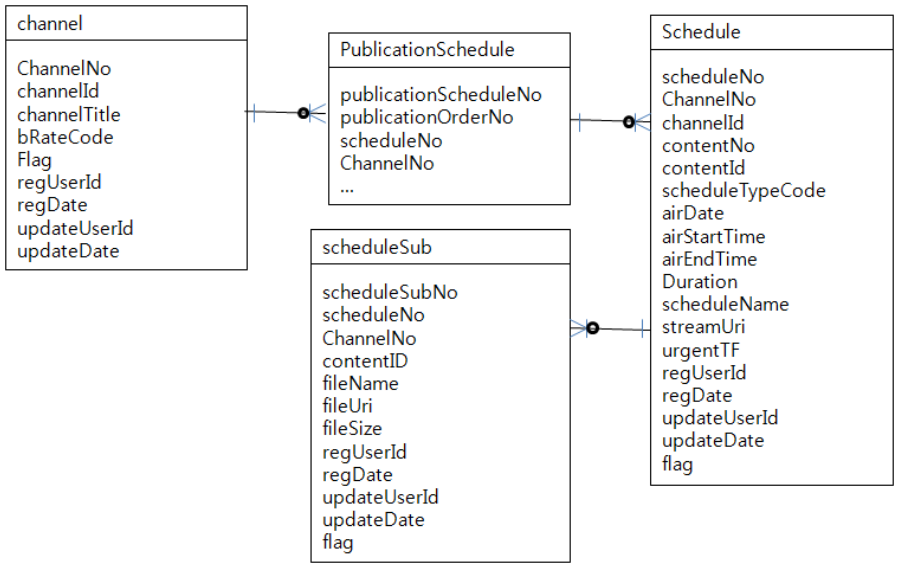

Figure 4. Our Design of a Database for the Editor

4. Design and Implementation of the System

We design a database as shown in Figure 4. It consists of Channel, Schedule, PublicationSchedule, and scheduleSub tables. In the Channel table, we record channel number, channel identification, channel title, bit rate, the user ID of the person who created the record, the date of creation, and history of update. In the Schedule table, we record information of all schedules. Schedule number of the schedule, the channel number of the channel on which this schedule is applied, and information of content items consisting of the schedule is recorded in this table. A schedule cannot be utilized until it is published. Publication information is recorded in the PublicationSchedule table. A

schedule can be associated with caption files. Information of caption files associated with a schedule is recorded in the ScheduleSub table.

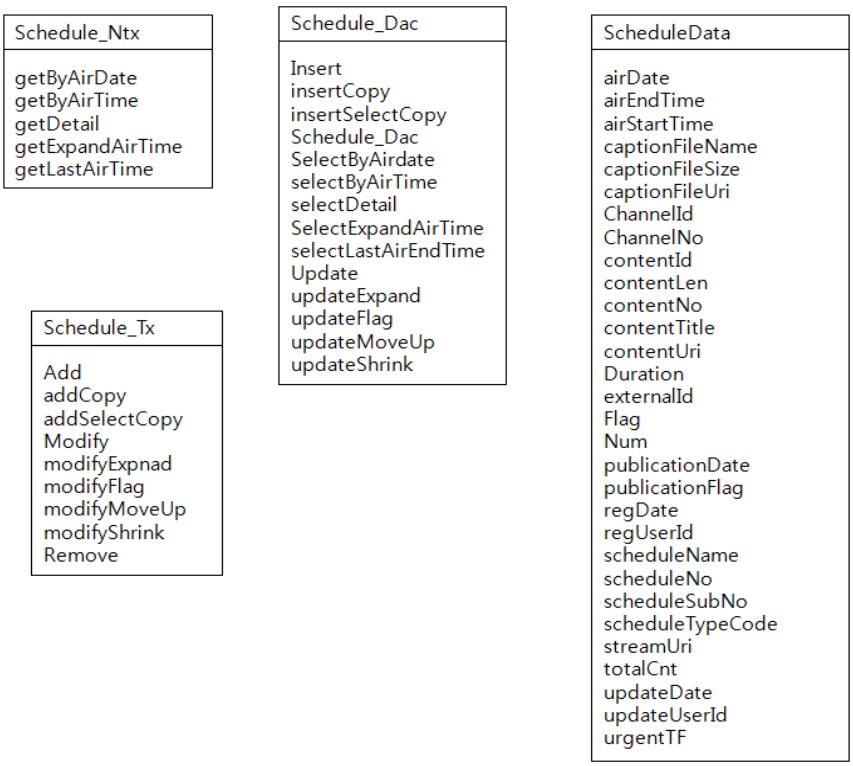

Figure 5. Description of the Classes to be Developed

We are exercising MSF/CD development strategy. Based on the system requirement, we develop business model, indentify components, develop components, and finally develop the application system. In the business layer, we have transactional processes and non-transactional processes. A transactional process must be rolled-back if the process is not committed. Get methods are non-transactional whereas add and modify methods are transactional. In the data access layer, we have methods to access (select, delete, update and create) the database. Various data types are declared in ScheduleData class as shown in Figure 5.

We implemented InterfaceAddChannel method that creates an XML file for Wowza set-up. This file can be sent to the stream management component that finds an available Wowza server and assigns a channel to it. Similarly, InterfaceRemoveChannel creates an XML string that is to be used by the stream manager inactivating a channel.

We defined InterfaceAddSchedule method that returns an InterfaceScheduleData with a given list of content items as shown in Figure 6. The given list of content items is represented by ds in the figure. It appends all contents to the param variable in JSON format. Then it transforms the JSON string into XML string. Finally, it transform the XML into InterfaceHeader.

```
public InterfaceHeader InterfaceAddSchedule(DateTime airdate, string channelId, DataSet ds)
...
  for(int i = 0; i < ds.Tables[0].Rows.Count; i++)
```

```
{
  DataRow dr = ds.Tables[0].Rows[i];
  if (i > 0) param.AppendFormat(",");
  ...
  param.AppendFormat("\"program_id\":\"{0}\",",
                     dr["PUBLICATIONSCHEDULENO"]);
 ...

 HttpWebResponse response =
     SendGetHttpWebRequest(param.ToString(), url, "POST");
 ...
 string xml = readStream.ReadToEnd();
 ...
 InterfaceScheduleData data =
        JSONHelper.Deserialise<InterfaceScheduleData>(xml);
 ...
```

Figure 6. A Part of our Implementation of InterfaceAddSchedule( )

In the Schedule_Ntx, many non-transactional methods are defined. A part of our definition of the GetByAirdate method is shown in Figure 7. It simply invokes SelectByAirdate defined in Schedule_Dac class. The other methods are all defined in the similar manner - simply invokes a method in Schedule_Dac class.

Many transactional methods are defined in Schedule_Tx class. As an example, we show a part of our implementation of the Add method in Figure 8. As we did in the Schedule_Ntx class, a method defined in the Schedule_Tx class performs what it has to do by invoking a method in Schedule_Dac. The Insert method defined in the Schedule_Dac in this case. However, we have to create an object of CreateTransactionScope( ) before we invoke a Schedule_Dac method and perform roll-back if the invoked method fails.

```
public class Schedule_Ntx : BizNTxBase
...
  public DataSet GetByAirdate(DateTime airdate, int channelNo)
  ...
   using (Schedule_Dac schedule_Dac = new Schedule_Dac())
   {
    return schedule_Dac.SelectByAirdate(airdate, channelNo);
   ...
```

Figure 7. A Part of our Implementation of the GetByAirdate( ) Method


```
public class Schedule_Tx : BizTxBaseForOracle
...
  public int Add(int channelNo, ... captionFileSize)
  ...
  using (var tx = CreateTransactionScope())
  ...
    using (Schedule_Dac schedule_Dac = new Schedule_Dac())
    ...
     scheduleNo = schedule_Dac.Insert(channelNo, ...);
     if (contentNo < 0)
     {
       CurrentRollBack();
    ...
       tx.Complete();
       bizResult = contentNo;
  ...
```

Figure 8. A Part of our Implementation of the Add Method

As an example of the methods defined in the Schedule_Dac class, a part of our implementation of the SelectDetail method is shown in Figure 9. The SQL sentence designated by "SP_DPS_SCHEDULE_DETAIL_Q" is a select sentence. The SetInParameter method assigns the scheduleNo as the in-parameter of the select sentence.

```
public DataSet SelectDetail(int scheduleNo)
...
    DbCommand command = OracleDataBase.GetStoredProcCommand
                              ("SP_DPS_SCHEDULE_DETAIL_Q");
    SetInParameter(command, "p_SCHEDULENO", DbType.Int32, scheduleNo);
    SetOutParameter(command, "CUR_OUT", OracleType.Cursor);
    ds = OracleDataBase.ExecuteDataSet(command);
    return ds;    ...
```

Figure 9. A Part of our Implementation of the SelectDetail( ) Method

5. Experiments

We have performed experiments of testing our user management system. User registration has been tested as shown in Figure 10.

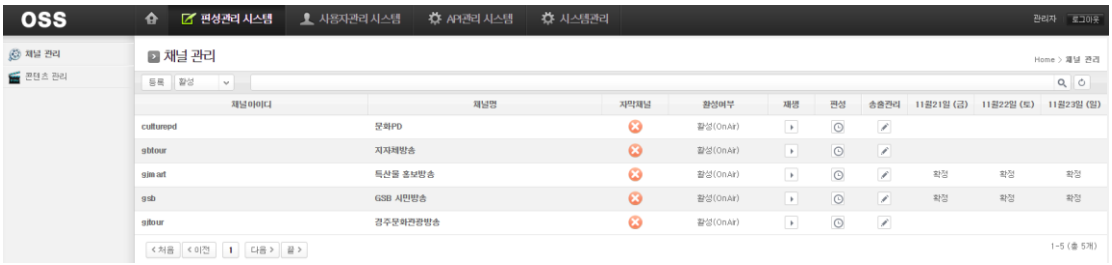

Figure 10. A Screenshot of the Main Page of the TV Schedule Editor

We have performed experiments to test our implementation. A screenshot of the first page of our TV schedule editor is shown in Figure 10. We reach this page after the login process. A list of channels is displayed in this figure. We can add a new channel and inactivate a channel at this page. If we click the edit button associated with the gjmart channel, then we have a window similar to Figure 11.

TV schedules for days of 23rd, 24th, and 25th are shown in the figure whereas the TV schedule for 26th is still empty. If we click the discard button associated with an existing TV schedule, then the schedule is deleted from the list. We can copy an existing TV schedule and paste it in an empty place. When we click the add button associated with 26th, a pop-up window shows as shown in Figure 12. If we know the content title of the item we want to add to the TV schedule, then we can write it in the search box in the window. Otherwise, we can click the lens symbol to see the list of all content items as shown in Figure 13. When we select an item in the list, we can see detail information of the selected item as the result of selection. If we click the save button, the selected item is added into the TV schedule as shown in Figure 14. We can select some items in TV schedule to copy or delete. We can always paste copied content items into any empty place. Once the TV schedule for the day is completed, we have to click the confirm button in order to save the work.

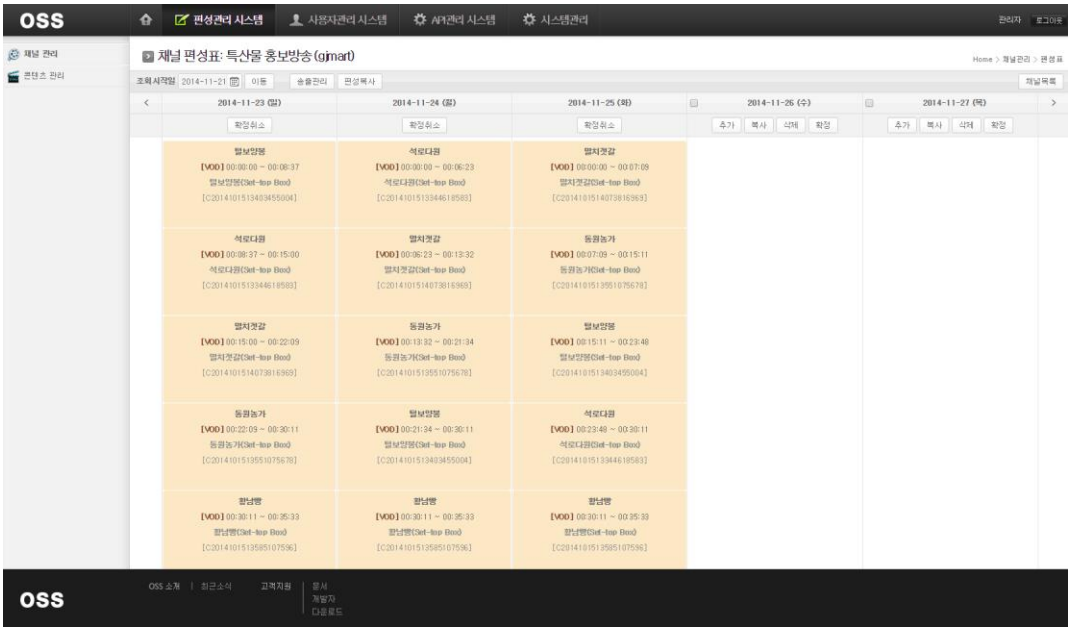

Figure 11. A Screenshot of the Window for Editing TV Schedules

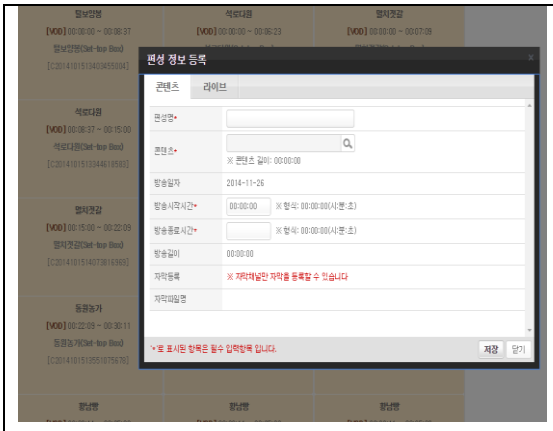

Figure 12: The Pop-up Window to Add an Item into a TV Schedule

Figure 13: A List of All Content Items

Figure 14. A Screenshot Showing the Result of the Add Command

7. Conclusions

We introduced our design and implementation of a TV schedule editor. This editor provides the users with all functions needed to edit TV schedules. Retrieving all channels, adding a new channel, and deactivating existing TV channels can be done with the editor for channel management. For editing TV schedules for TV channels, copying and pasting an existing TV schedule, copying and pasting content items shown in TV schedules, adding a new content item or live stream to a TV schedule, deleting content items from TV schedules can be done with the editor. The editor is closely related to the content management system (CMS) in that the editor retrieves available content items by invoking the retrieval methods provided by the CMS.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2011-0006942), by 'Development of Global Culture and Tourism IPTV Broadcasting Station' Project through the Industrial Infrastructure Program for

Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393) and the Dongguk University Research Fund of 2015.

References

[1] J. Yim, G. Lee, T. Lee and J. Jeon, "Implementation of Mobile Internet TV Channels", IJSEIA, vol. 8, no. 6, **(2014)**, pp. 273-286.
[2] C. Chou, F. Zahedi, and H. Zhao, "Ontology for Developing Web Sites for Natural Disaster Management: Methodology and Implementation, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 41, no. 1, **(2011)**, pp. 50 – 62.
[3] M. Mukhtar, M. Hassan, J. Bin and L. Rahim, "Enhanced approach for developing web applications using model driven architecture", International Conference on Research and Innovation in Information Systems, **(2013)**.
[4] J. Yim, "Design of a Web Application for Local Government Internet TV System", Advanced Science and Technology Letters, vol. 66, **(2014)**.
[5] H. Hwang, "Design of an Effective Learning Evaluation Component in Web-Based Instruction", IJMUE, vol. 6, no. 4, **(2011)**, pp. 1-12.
[6] A. Mirzal, "Design and Implementation of a Simple Web Search Engine", IJMUE, vol. 7, no. 1, **(2012)**, pp. 53-60.
[7] H. Yim and K. Lee, "A Ubiquitous Web Services Framework for Interoperability in Pervasive Environments", IJMUE, vol. 7, no. 3, **(2012)**, pp. 43-50.
[8] H. Chong and F. Gaol, "ULa Lab: Ubiquitous Open Contents Web-Based Language Laboratory using REST Protocol Web Service", IJMUE, vol. 8, no. 4, **(2013)**, pp. 199-206.
[9] S. Shin, J. Ku and J. Lee, "Design and Implementation of Web App to Facilitate Interactions in Classroom", IJMUE, vol. 8, no. 4, **(2013)**, pp. 377-384.
[10] T. Ban, T. Cho and H. Jung, "A Study on the Hybrid Web-based Real-Time Video Communication System", IJMUE, vol. 9, no. 3, **(2014)**, pp. 11-20.
[11] S. Yoon and Y. Ho, "View Synthesis and Coding of Multi-view Data in Arbitrary Camera Arrangements Using Multiple Layered Depth Images", Journal of Multimedia and Information System, vol. 1, no. 1, **(2014)**, pp.1-10.

Authors

Jaegeol Yim, He received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.

Gyeyoung Lee, He received the M.S. degree in Computer Science from the Dongguk University in 1982 and the Ph.D. degree in Computer Engineering from the Dankook University in Korea in 1992, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju, Korea. His current research interests include Petri net theory, AI systems and Speech processing systems.

