Scalable Advertising Framework for Multi-screen Service

Tae-Beom Lim[1] and Kyoungro Yoon[2,3]

[1] *Korea Electronics Technology Institute*
[2] *Konkuk University*
tblim@keti.re.kr, yoonk@konkuk.ac.kr

Abstract

This paper presents an advertising framework to ensure the targeted advertisement service for heterogeneous consumer devices with different capabilities and subscribers with different preferences by providing advertisement content selection and filtering mechanism in real time. To support scalable advertisement service in multi-screen service environment, we suggest a new advertising framework using XML schema based on ISO/IEC 21000-2 Digital Item Declaration Language.

Keywords: *Scalability, Advertisement, Multi-screen, MPEG-21, XML schema*

1. Introduction

Recently, broadcast service providers and internet service providers are trying to deploy the multi-screen services that provide broadcast and internet services through various consumer devices such as high-definition smart TV, personal computers, mobile phones, etc. Also, for raising revenue, they are looking for the advertisement system that is able to present the suitable advertisement contents on various consumer devices with different capabilities and that is able to deliver the targeted advertisement contents to each user depending on user profile, user behavior, device capabilities, and/or their service policies. For the success of targeted advertisement services through different types of devices and different preferences of users, scalable advertisement contents should be provided.

ISO/IEC 21000-2 (a.k.a. MPEG-21) Digital Item Declaration (DID) [1] provides basic framework of choice-selection-condition through which individual multimedia resource can be selected to provide an adaptive consumption of multimedia contents. ISO/IEC 21000-7 Digital Item Adaptation (DIA) [3] provides various tools of describing terminal, network, and user environment as well as a tool for adaptation of contents, to achieve interoperable transparent access to distributed multimedia content by shielding users from network and terminal installation, management and implementation issues [2]. Generally, DIA tools could be used to satisfy various needs on transmission, storage, consumption constraints, and Quality of Service management. But, current DIA tools are not optimized to describe conditions for presentation of adapted advertisement content on user terminal by selecting and filtering Digital Items.

In this paper, we propose a new advertising framework to provide functionality of selecting and filtering advertisement contents based on device capabilities, user information, user viewing environment, and service policies in real time. Also, we design a new description language to describe the conditions for the advertisement contents structured as MPEG-21 DIDL [1] in a form of XML schema [5] called Scalable Advertisement Language (SAL). Advertisement contents described as MPEG-21 Digital Items are selected and reconstructed

[3] Corresponding Author.

in the adaptation process, to be optimally presented at each individual environment, based on the delivery context.

2. MPEG-21 Digital Item Declaration Model

MPEG-21 aims at defining an open framework for multimedia delivery and consumption for use by all the participating parties in the supply and consumption chain [3]. This open framework provides the technology needed to support users to produce, consume, exchange, and manipulate all the multimedia resources across a wide range of networks and devices in transparent and interoperable way. MPEG-21 Digital Item Declaration Model describes a set of abstract terms and concepts to form a useful model for defining Digital Item which is a structured digital object with a standard representation, identification and metadata within the MPEG-21 framework [1], [2].

3. Scalable Advertisement Language

SAL is a new description language in a form of XML schema, and can describe functions of selecting and filtering advertisement contents based on the factors such as device capabilities, user information, user viewing environment, or service policies. In addition, we design SAL to be harmonized with MPEG-21 DIDL easily.

We define a new complex type named *DCCondition* to describe conditions for selecting and filtering Digital Items to present an adapted advertisement content [3]. *DCCondition* type is extended from *StatementType* type of DIDL for the harmonization with DIDL. *StatementType* type provides a textual value that contains descriptive, control, revision tracking or identification information [1]. Also, to minimize the number of letters used in a XML document and the depth of the XML tree of the condition description, it is proposed to use the Reverse Polish Notation (RPN), which can be easily implemented using stack-based functions, to express complex conditions based on mathematical expressions such as Boolean expressions, comparison expressions or arithmetic expressions. Figure 1 shows the structure of *DCConditionType* type. *DCConditionType* type can contain *StackEntry* elements to describe condition expressions for selecting and filtering Digital Items as the RPN form.

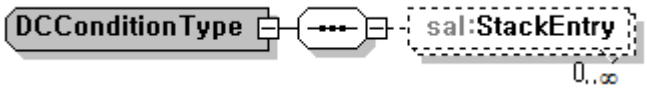

Figure 1. The structure of *DCCondition* type

StackEntryType type is a base abstract type to represent an operator or an operand for describing stack entries that are members of the stack operation. In SAL, an operand of the stack operation may contain a constant value, or a DCFunction that is a function for taking the characteristics of the device, service policy, user information, user viewing environment, and any other aspects that affect execution and presentation of advertisement contents on a client device from the system. The stack entities are classified into three groups of types based on the data type of the stack entry or the return value type of the DCFunction. These three new groups of entry types that are extended from *StackEntryType* type are *BooleanEntryType* type, *StringEntryType* type, and *NumericEntryType* type. If these types are used in SAL documents, these types shall be extended as operator types, constant value types, and *DCFunction* types because these are abstract types.

3.1. Operator Types

The operator types are subclasses of the stack entries, and allow building the stack operation trees. There are three groups of operator types: the Boolean operator group, the string operator group, and the numeric operator group. Each group is composed of the operators that have the same return value type as a result of the operation. In SAL, only the return value type is considered as the criteria of grouping operators, as it is not straightforward to identify the operand data types of a specific operator before actual stack operations are performed due to the characteristics of stack operations. Table 1 shows operation types in SAL XML schema.

Table 1. List of operator types

Class	Operator Type	Return Value
Boolean	*AND, OR, NOT, XOR*	Boolean
Comparison	*EQ, NEQ, GT, GTE, LT, LTE*	Boolean
String	*Contains*	Boolean
String	*UpperCase, LowerCase*	String
Numeric	*Add, Subtract, Multiply, Divide, Modulus, Abs, Ceiling, Floor, Round*	Numeric

3.2. DCFunction Types

The *DCFunction* types define the functions that check and return input factor values that may influence the application adaptation of the SAL filtering engine in the server, the intermediary, or the client. Table 2 shows the characteristics of *DCFunction* types. It is not an exhaustive list of the *DCFunction* types, but is a list of minimum set of *DCFunction* types used in our system.

Table 2. The Characteristic of the *DCFunction* types

Characteristics	Information
Device Capability	Client Device type, Operating System, Supporting Codec (Video, Audio, Image, etc), Input Interface, Output Interface, Screen Resolution, Virtual Machine, etc
Device State	Power Level, Orientation of Screen, etc
User Information	User Profiles, User Preference, Usage History
Connection	Bandwidth, Network Protocols, Latency, etc
Location	Geographic Coordinates, Time of Day
Locale	Local Language, Local Time Zone
Environment	Noise, Light, Temperature, etc
Service Policy	Subscription Status, Content Restriction, Security, Privacy, etc

Because there are many aspects of the input factors, we categorize the *DCFunction* types for the easy extensibility and manageability of the SAL schema. In the first level, we divide the *DCFunction* types into two categories of *Client* and *Sever*, according to the device expected to process the function described in the DCFunction type. In the second level, each category of the DCFunctions is subdivided into two subcategories of *Static Return Value* and *Dynamic Return Value* based on the flexibility of the return value of function. The DCFunction is categorized as *Dynamic Return Value* if the return values can change dynamically. Otherwise, it is categorized as *Static Return Value*. In the third level, each of the categories at the second level is categorized further into four subcategories of *Device Capabilities/State, User Information, Viewing Environment/*

State, and *Service Policy* based on the characteristics of the functions. In the last level, the *DCFunction* types are categorized into three categories of Boolean DCFunctions, String DCFunctions, Numeric DCFunctions, based on the type of the return value, in the same way as the constant value type, and the operator type. The DCFunctions are classified into sub-types using the typing mechanism of the XML schema to enable the validation mechanism of the SAL parser for the validity check of the function types in the process of parsing a SAL instance document. Figure 2 shows a snippet of the Boolean *DCFunction* types in SAL XML schema, and Figure 3 shows a snippet of SAL instance document, in which the condition given in the following example is described.

((Horizontal pixel resolution of screen > 1600) & (Color range > 256))

```
<complexType name=" DCHPixelResolution">
 <complexContent> <extension base="sal:NumericEntryType"/> </complexContent>
</complexType>
...
<complexType name=" DCColorRange">
 <complexContent> <extension base="sal:NumericEntryType "/> </complexContent>
</complexType>
```

Figure 2. Snippet of the numeric *DCFunction* types

```
<Selection select_id="HighProfile">
 <Descriptor>
  <sal:DCCondition>
   <sal:DCCondition>
    <sal:StackEntry xsi:type="sal:DCHPixelResolution"/>
    <sal:StackEntry xsi:type=
                 "sal:NumericValueType" value="1600"/>
    <sal:StackEntry xsi:type="sal:GT"/>
    <sal:StackEntry xsi:type="sal:DCColorRange"/>
    <sal:StackEntry xsi:type=
                 "sal:NumericValueType" value="256"/>
    <sal:StackEntry xsi:type="sal:GT"/>
    <sal:StackEntry xsi:type="sal:AND"/>
  </sal:DCCondition>
 </Descriptor>
</Selection>
```

Figure 3. An example of using the numeric *DCFunction* type

5. Scalable Advertisement Platform

Figure 4 shows the structure of real-time scalable advertisement platform using SAL documents. The SAL advertising content authoring tool generates an SAL document which wraps legacy advertisement contents for enabling this content suitable for various users and various client devices that have different device capabilities. When a user wants to consume an application, the client sends a request message with server specific delivery contexts to the server. Then, an SAL document which includes requested application should go through at least one SAL filtering engine before being

presented to the requesting client. SAL filtering engine can be located at any of the servers, or a client. Generally, the SAL advertisement content filtering engine in a server performs the process of selecting and filtering the SAL document based on delivery contexts that are input factors concerned with service policies and user subscription information. The filtering engine in a client will be operated if it is necessary that it should perform the process of selecting and/or filtering by real-time device state information and user information which user does not want to deliver to any servers. It determines which specific component or item of MPEG-21 DIDL to be selected and/or filtered by interpreting *DCCondition* elements of SAL and Condition elements of DIDL. For the selection and/or filtering of appropriate component or item, the delivery context shall be examined and matched against the DCCondition. Once the selecting and filtering process is performed, the application including the set of selected advertisement contents can be rendered by a parser and/or a renderer in a client.

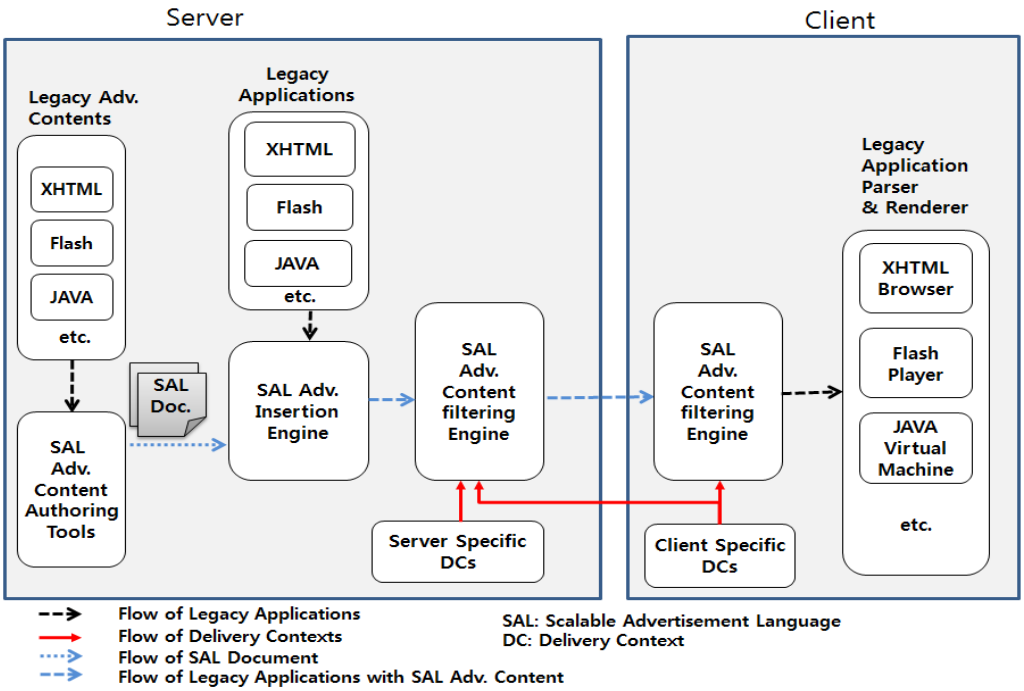

Figure 4. The Structure of Scalable Advertisement Platform

6. Conclusion

In this paper, we propose a new advertising framework to provide functionality of selecting and filtering advertisement contents based on device capabilities, user information, and service policies in real time. Also, we design a new description language to describe the conditions for the advertisement contents called Scalable Advertisement Language.

By using this advertising framework, service providers can apply various targeted advertisement services (e.g. user-targeting, device-targeting, location-targeting, etc) to multi-screen service environment easily. Furthermore, this advertising framework provides scalability of an advertisement content which is presented on the various client devices simultaneously.

References

[1] ISO/IEC International Standard 21000-2: Information technology – Multimedia framework (MPEG-21) – Part 2: Digital Item Declaration **(2005)**.

[2] I. S. Burnett, S. J. Davis and G. M. Drury, "MPEG-21 digital item declaration and identification-principles and compression", IEEE Trans. Multimedia, vol. 7, no. 3, pp. 400-407 **(2005)**.

[3] ISO/IEC International Standard 21000-7: Information technology – Multimedia framework (MPEG-21) – Part 7: Digital Item Adaptation **(2007)**.

[4] A. Vetro and C. Timmerer, "Digital item adaptation, overview of standardization and research activities", IEEE Trans. Multimedia, vol. 7, no. 3, pp. 418-426 **(2005)**.

[5] ISO/IEC International Standard 21000-1, Information technology – Multimedia framework (MPEG-21) – Part 1: Vision, Technologies and Strategy **(2004)**.

[6] J.-Y. Lee and Y.-C. Kil, "Advertisement Management System utilizing On-Line Music Services in Communication Networks", Journal of IWIT, vol. 10, no. 6, pp. 187-193 **(2010)**.

[7] B.-I. Kwon and N.-M. Moon, "A study of Metadata design for Digital Content Marketplace based on Interactive Media", Journal of IWIT, vol. 9, no. 3, pp. 155-164 **(2009)**.

Authors

Tae-Beom Lim

He received a B.S degree in Physics from Sogang University, Seoul, Korea in 1995, a M.S.E degree in Computer Science and Engineering from Sogang University, Seoul, Korea in 1997, and a Ph.D. degree in Computer Science and Engineering of Konkuk University in 2012, Seoul, Korea. He was a research engineer in Imaging Technology Lab, Daewoo Electronics Institute of Technology from 1997 to 2002. He joined Korea Electronics Technology Institute in 2002 as researcher and is a principal researcher now. Also, He served as an editor of ISO/IEC IS 15938-12 (MPEG Query Format).

Kyoungro Yoon

He received a B.S. degree in Computer and Electronic Engineering from Yonsei University, Seoul, Korea in 1987, a M.S.E. degree in Electrical Engineering/Systems from University of Michigan, Ann Arbor in 1989, and a Ph.D. degree in Computer and Information Science from Syracuse University in 1999. He became a Member (M) of IEEE in 2000. He was a principal researcher and a group leader in Mobile Multimedia Research Lab, LG Electronics Institute of Technology from 1999 to 2003. He joined the school of Computer Science and Engineering of Konkuk University, Seoul, Korea in 2003 as an assistant professor and is an associate professor now. He served as a chair of Ad Hoc Groups on User Preferences and MPEG Query Format of ISO/IEC JTC1 SC29 WG11 (a.k.a. MPEG) and is currently serving as a chair of JPSearch Ad Hoc Group of ISO/IEC JTC1 SC29 WG1 (a.k.a. JPEG).