MHEG-5 객체의 인코딩/디코딩 클래스 라이브러리 설계

이 세 훈† · 왕 창 종††

요 약

MHEG-5는 이기종 통신망 상에서 멀티미디어/하이퍼미디어 정보의 실시간 상호교환, 프리젠테이션, 사용자 상호작용을 제공하기 위한 정보 표현 표준 및 전송을 위한 표준이다. MHEG-5에 기반한 응용 서비스를 제공하기 위해서는 MHEG-5 객체의 생성 및 편집이 필수적으로 요구된다. 따라서 본 연구에서는 MHEG-5 객체의 인코딩, 디코딩과 속성 설정을 제공하고, 이기종 간의 상호교환을 지원할 수 있도록 클래스 라이브러리를 설계하였다. 설계한 클래스 라이브러리는 이질적인 통신망을 통해 분산 멀티미디어/하이퍼미디어 서비스 제공을 위한 MHEG-5의 기반 기술을 제공한다는 점에서 의의를 갖는다.

Design of Encoding/Decoding Class Library for the MHEG-5 Objects

Seihoon Lee† · Changjong Wang††

ABSTRACT

The MHEG-5 standard defines representation of multimedia and hypermedia information. The MHEG-5 can make it possible to interchange and present information between heterogeneous networked environment. Also, it can provide real-time user interaction. MHEG-5 object generating is the essential requirements for the MHEG-5 applications. In this paper, we designed encoding/decoding class library for the MHEG-5 objects. We can use it for the interchange on the network. The designed class library can provide base technology for the MHEG-5 based distributed multimedia and hypermedia services on the networked environment.

1. 서 론

Network technologies and multimedia are being popular. We have required new mechanism to interchange and present multimedia and hypermedia information on the network. The MHEG standard defines the representation and encoding of multimedia and hypermedia information for interchange between various applications[1]. The MHEG-5 standard is the fifth subset of the MHEG standard. It defines some classes in detail. Those classes are appropriate to some applications such as video on demand, audio on demand, interactive TV and hypermedia navigation[2, 3].

The MHEG-5 standard uses the ASN. 1(Abstract

※이 논문은 1996년도 한국학술진흥재단의 공모 과제 연구비에 의하여 연구되었음
† 정 회 원:인하공업전문대학 전자계산기과
†† 정 회 원:인하대학교 전자계산공학과
논문접수:1997년 3월 17일, 심사완료:1997년 9월 12일

Syntax Notation One) for object representation. The ASN.1 defines data structures that are exchangeable. But it is difficult to assign a value to a variable and represented in binary form. It is difficult to analyze and detect an error when exchanging. So, we require what must support object generations and modifications without adding new module to analyze the MHEG-5 objects[4, 5].

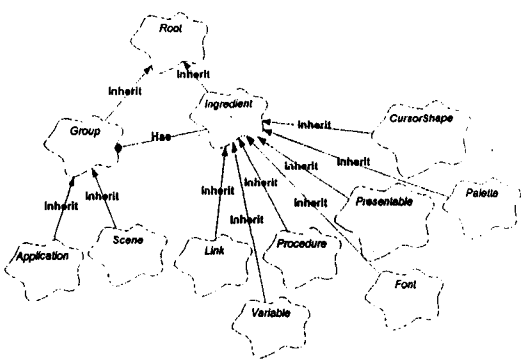
In this paper, we design encoding/decoding class library for the MHEG-5 objects. The class library can be used as the MHEG-5 encoder/decoder based on MediaADE, MHEG-based application development environment, in [6]. It can provide correct encoding and decoding of the MHEG-5 objects. We implement the MHEG-5 object generator for experimenting this class library. It can be used for developing other MHEG-5 applications. Using the class library, it is easier to modify encoder and decoder routines.

2. Related works

In this chapter, we review the MHEG-5 standard and object generation rules. In addition, we review the Snacc (Sample Neufeld ASN.1 to C/C^{++} Compiler) [4, 7]. The Snacc is a general purpose ASN.1 compiler. It is used for the MHEG-5 encoder and decoder.

The MHEG standard provides representation of multimedia objects on heterogeneous networked environment. It also provides encoding/decoding for real-time interchange between applications and synchronized presentations[1, 2, 3, 8, 9]. It can support real-time user interaction when presentation. Multimedia information, which encoded by the MHEG standard, is the MHEG object. As using MHEG objects as inherited form in applications, we require an environment to present these objects. The MHEG standard defined such environment as engine[1]. The MHEG objects are final form representation. Thus the MHEG has some advantages. One is that the operation speed is high. The other is that it is possible to process user interaction easily[10, 11].

The MHEG-5 standard is the fifth subset of the MHEG standard. It defines some classes in detail. Those classes are appropriate to some applications such as video on demand, audio on demand, interactive TV and hypermedia navigation[2, 3]. The MHEG-5 system consists of server, client, the MHEG-5 objects, and the MHEG-5 applications which are sets of the MHEG-5 objects[2, 12, 13]. We can generate the MHEG-5 objects when create instance of the MHEG-5 classes, set the attributes, and then encode. (Fig. 1) is a diagram for the MHEG-5 class hierarchy.



(Fig. 1) Hierarchy for the MHEG-5 classes

The application object includes the object and other objects which shared by all scene objects. The scene object is a set of objects for presentation. It represents real multimedia information. It also includes the ingredient object. The ingredient object is responsible for event which occurred by user. The MHEG-5 applications are performed by transiting among scenes. The MHEG-5 applications are not procedural but declarative. We can execute all the MHEG-5 based application on any platform which can support the MHEG-5.

The MHEG-5 standard uses the ASN.1 and the BER (Basic Encoding R 'e) when it represents multimedia information[2, 8, 14]. The ASN.1 defines data structures that are exchangeable between heterogeneous machines. Data structures which defined by the ASN.1 are the ASN.1 modules[14]. The system, which uses the ASN.1 modules, assigns values to the ASN.1 modules. Then it converts to binary forms for transmitting other systems. The rules that convert the ASN.1 modules to binary forms are the BER. The MHEG-5 objects are multimedia information represented by the BER[1, 9, 14].

Many of the MHEG-5 application developers use Snacc for developing MHEG Objects encoder and decoder[4]. It is also designated to the ASN.1 compiler for GLASS(GLobally Accessible ServiceS) project by Digital CEC, GMD FOKUS, Grundig Multimedia Solutions, IBM ENC, and DeTeBerkom[12, 15, 16]. The Snacc is a general purpose ASN.1 compiler. But, some problems occurred when we use the Snacc for encoding and decoding of the MHEG-5 objects. (1) There are some bugs in hash table which the Snacc uses when decide data types. (2) Because Snacc is a general purpose ASN.1 compiler, it cannot be adopted by MHEG-5 client that targets system which has few resource, such as interactive TV. (3) In grammar, which defines for compilation, some bugs exist [7]. Because of these, errors can be occurred when compiling complex modules such as nested, multiple inheritance.

3. Design of encoding/decoding class library

In this chapter, we design encoding/decoding class library for the MHEG-5 Objects. It can support to generate, interpret, modify, and reuse the MHEG-5 objects.

The requirements for the MHEG-5 encoder and decoder are like these; (1) They must have portability. (2) They provide correct encoding and decoding method for all classes of the MHEG-5 ASN.1 modules. These are for supporting to interchange on the heterogeneous, distributed environment. In the design of class library, we use object oriented method for class representation. It has several advantages; (1) Then, we can use the class library to develop various MHEG-5 applications such as browser, engine, and server. (2) As using object oriented method, the library has flexibility when any modification needed.

To resolve problems which Snacc has, we design the class library which mapped into data structures of the MHEG-5 ASN.1 modules. The designed library can support correct encoding and decoding of all the MHEG-5 objects. In the design of class library, we design the base class which can encode and decode one BER item. Then we define other classes by inheriting from the base class.

3.1 Base class

We can inherit any other classes from the base class. In our class library, we define the base class to inherit subclass. The CMH5Object class, the base class, has method which can encode and decode one BER item. Each member variable of the CMH5Object class is mapped into octets of the BER. The CMH5Object class has methods for processing octets. It decodes the BER items, and the provides decoded information to child classes. It also encodes information, which child classes specify, to the BER. (Fig. 2) shows members and methods of the base class, CMH5Object.

```
CMH5Object {
  IDTYPE {
    UNIVERSAL = 0x00, APPLICATION = 0x40,
    CONTEXT_SPECIFIC = 0x80, PRIVATE = 0xC0};
  ...
methods :
  ReadOneItem, WriteOneItem, GetOneOctet,
  GetIDType, IsPrimitive, GetTagNumber,
  GetItemLength, GetContents,
  GetNoOfRequiredTagOctets,
  GetNoOfRequiredLengthOctets,
  WriteIDOctet, WriteLengthOctet, WriteContents;
members :
  Length, IDType, Prim, TagNo, Content, Buff;
...
};
```

(Fig. 2) Members and Methods of CMH5Object

The methods, ReadOneItem and WriteOneItem, operate real encoding and decoding BER. Algorithm

1 is BER decoding algorithm ReadOneItem.

Algorithm 1 BER decoding

```
CMH5Object::ReadOneItem(BerBuff)
Begin Method
oct := GetOneOctet;
IDType := GetIDType;
Prim := IsPrimitive;
TagNo := GetTagNumber;
oct := GetOneOctet;
Length := GetItemLength;
If (Prim == TRUE) Then Content := GetContents
Else Content := NULL;
EndIf
Return Buff;
End Method
```

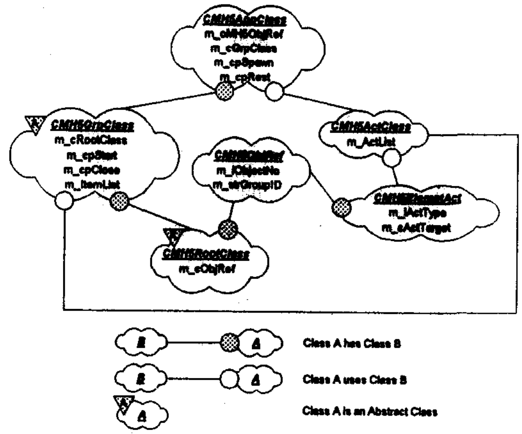
Each child class which inherits from the CMH5O bject class is mapped into the MHEG-5 classes. Child classes decode the MHEG-5 object by invoking ReadOneItem. While information of all member variables is gathered, parent classes invoke ReadOneItem for each object. Algorithm 2 is BER encoding algorithm, WriteOneItem.

Algorithm 2 BER encoding

```
CMH5Object::WriteOneItem()
Begin Method
  Leng := 0;
  Leng := Leng + GetNoOfRequiredTagOctets;
  Leng := Leng + GetNoOfRequiredLengthOctets;
  Leng := Leng + Length;
  If (Buff != NULL) Then
    Delete Buff;
    Allocate Buff with Length;
  EndIf
  WriteIDOctet;
  WriteLengthOctet;
  If (Prim == TRUE) Then
    WriteContents;
  Else
    Return Leng;
  EndIf
  Return Leng;
End Method
```

3.2 Application class group

The application class and the scene class inherit from the base class. There are members such as identifier and methods for grouping the ingredient classes in the application class. In the group class, the root class exists as member variable. The root class is parent class of all other MHEG-5 classes. The root class has an identifier of the application class. The group class includes the action object which must be executed after the MHEG-5 application activated. Users can generate an application object with the BER file name as parameter. (Fig. 3) shows class diagram of application class group.



(Fig. 3) Application class group

Users can decode the application object by invoking BerDecode. During encoding of the application object, we must add the BER lengths of lower level objects to length octet of higher level object. Thus, we need two encoding phases. In first phase, from the lowest level objects to the application object, each object is converted to the BER and length of each BER is returned for updating length octet of higher level object. In second phase, from the application to the lowest level objects, each BER stored to file. Algorithm 3 and 4 is encoding/decoding algorithm for application object.

Algorithm 3 BerEncode for application object

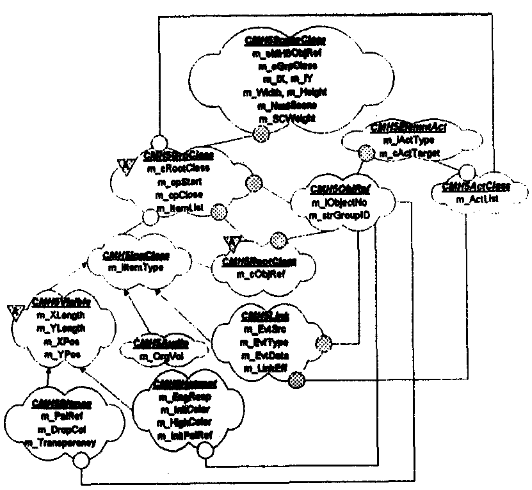
```
CMH5AppClass::BerEncode(m_strBerName)
Begin Method
  m_iLength := WriteOneItem();
  subLength := m_cGrpClass.BerEncode();
  If(OnSpawnDown is Encoded) Then
    subLength := subLength + m_cpSpawn.BerEncode();
    m_iLength := m_iLength + subLength;
  EndIf
  If(OnRestart is Encoded) Then
    subLength := subLength + m_cpRest.BerEncode();
    m_iLength := m_iLength + subLength;
  EndIf
  BerWrite(m_strBerName);
  m_cGrpClass.BerWrite(m_strBerName);
  If(OnSpawnDown is Encoded) Then m_cpSpawn.BerWrite(m_strBerName);
  If(OnRestart i Encoded) Then m_cpRest.BerWrite(m_strBerName);
  return TRUE:
End Method
```

Algorithm 4 BerDecode for application object

```
CMH5AppClass::BerDecode(BerBuff)
Begin Method
  m_szBerBuffer := ReadOneItem(m_szBerBuffer);
  m_oMH5ObjRef := m_oContent;
  m_szBerBuffer := m_cGrpClass.BerDecode(m_szBerBuffer);
  If(m_szBerBuffer is NULL) Then return TRUE;
  Else
    temp := m_szBerBuffer;
    ReadOneItem(m_szBerBuffer);
    m_szBerBuffer := temp;
    If(on-spawn-close-down is encoded) Then
      m_szBerBuffer  := m_cpSpawn.BerDecode(m_szBerBuffer);
      If(m_szBerBuffer is NULL) Then return TRUE;
      Else // decode on-restart
        m_szBerBuffer := m_cpRest.BerDecode(m_szBerBuffer);
        return TRUE;
      EndIf
    Else // decode on-restart
      m_szBerBuffer := m_cpRest.BerDecode(m_szBerBuffer);
    EndIf
    return TRUE;
  EndIf
End Method
```

3.3 Scene class group

The scene class group defines the ingredient classes activated simultaneously. Unlike the application class, the scene class includes member variables which are required for presentation such as position and size. It contains a member which points to next scene object. The group class which is included as member variable by the scene class has item list. Encoding and decoding of the scene object is same as them of application class.

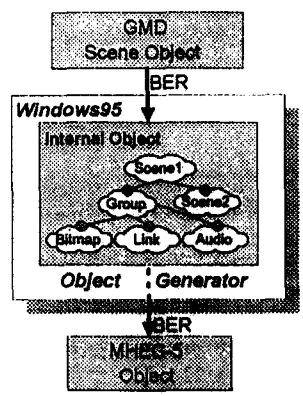


(Fig. 4) Scene class group

4. Experimentation

In this chapter, we make some experimentation to show that the class library is applicable to distributed multimedia environment. For this experimentation, we implement the MHEG-5 object generator. The object generator is able to decode, modify, and encode the MHEG-5 objects. Then we compare it to Snacc. Since we have not implemented the application programming interface for end user presentation systems and interface modules, we except experimentation about presentations.

In this experimentation, we use the MHEG-5 objects that are used by the MHEG-5 engine of GMD[13]. We choose scene objects rather than application objects because scene objects are more complex than application objects. The scene object has multilevel nested structures, and it has the link object. Thus the scene objects are applicable to our experimentation. (Fig. 5) shows the environment for experimentation.



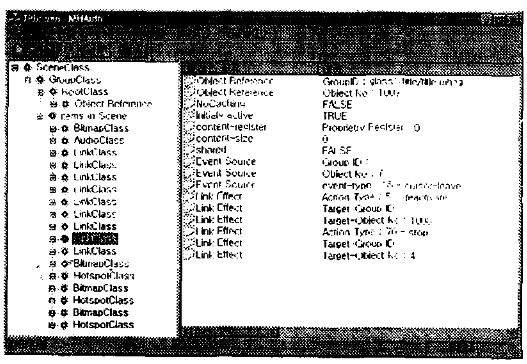
(Fig. 5) Experimentation environment

We make two experiments like these; (1) Whether the C^{++} class which created from the MHEG-5 objects by BER is correct or not. (2) Whether the modified objects correctly encoded or not. The MHEG-5 standard provides textual notation for representing the MHEG-5 objects [2]. (Fig. 6) shows the textual notations of a scene object which we plan to examine.

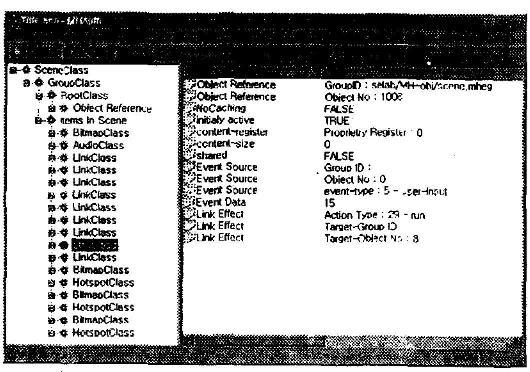
```
{:scene
  :object-identifier ( "glass1/title/title.mheg" 0 )
  :group-items
  ( ......
  {:link
    :object-identifier 1009
    :event-source 7
    :event-type CursorLeave
    :link-effect (:deactivate 1008
      :stop 4 ) } ......
  {:hotspot
   :object-identifier 7
   ......}
```

(Fig. 6) Textual notation for a scene object

(Fig. 7) shows that the object generator analyzes a scene object, and shows attributes and information of items in it. As shown below, the object generator decodes the scene object correctly (in Fig. 8).



(Fig. 7) Content of a scene object



(Fig. 8) Modified link object

After the object generator figured the MHEG-5 object to screen, user can modify attributes by using dialog boxes. We modified the link object and group identifier of the scene object and then stored. In this experimentation, we show that the designed class library resolves encoding error which the Snacc has.

(Fig. 9) shows that the Snacc make incorrect encoding at underlined octet. When encoding event-type, 01 is length octet, 11 is content octet to represent event, CursorEnter. Tag value 0A is identifier octet to represent enumeration type. Thus, octet 0D has no meaning. Because of this, length of the link object may not equal to which specified in itself. This error can make miss-reading problem in decoding phase. In addition, it cannot decode event-type correctly.

```
...
A4 08 <-- LinkCondition [4]
30 03 <-- event-source ObjectReference
02 01 05 <-- object-number
0D 0A 01 11 <-- event-type CursorEnter
...
```

(Fig. 9) Error in link object made by the Snacc

As shown above, we show that the object generator using designed class library can support encoding, decoding, and modification of the MHEG-5 object correctly. Also we show that the Snacc is inapplicable to interchanging the MHEG-5 objects because it can make miss-reading problem when encoding object.

5. Conclusion

In this paper, we designed the MHEG-5 encoding/decoding class library. We use the library for various distributed multimedia and hypermedia applications such as video on demand, audio on demand, interactive TV, and hypermedia navigation. We are currently implementing the run-time presentation system and interface units.

The designed library can provide encoding, decoding, modifying, and efficient management of the MHEG-5 objects. The class library resolved problems which occurred with use of the Snacc. As using object oriented method, the library has flexibility when any modification needed. Also, we can use it as base technology when we develop other MHEG-5 applications. We have two plans in current. One is to develop integrated application development environment, which includes presentation. The other is to design the MHEG-5 class application programming interface to enhance this library.

REFERENCES

[1] ISO/IEC IS 13522-1 Information technology-Coding of Multimedia and Hypermedia information-Part 1 : MHEG object representation-Base notation(ASN.1), 1996.

[2] ISO/IEC IS 13522-5 Coding of Multimedia and Hypermedia Information-Part 5 : Support for Base-Level Interactive Applications, 1997.

[3] Robert Joseph, MHEG-5 : An Overview, http://www. fokus. gmd. de/ovma/mheg/rd1206. html.

[4] Rober Joop, Design and Implementation of a Generic ASN. 1 Editor, Diploma Thesis, Technische Universität Berlin.

[5] M. Lockyer, M. Badham, MHEG Authoring, http://www.octacon. co.uk/proj/diamond/tee 1. htm.

[6] S. H. Lee, C. J. Wang, "MediaADE : The MHEG-based distributed multimedia/hypermedia Application Development Environment," in Proc. of World Conference on Educational Multimedia and Hypermedia, pp. 378-383, June, 1996.

[7] M. Sample, Snacc 1.2rj : A High Performance ASN. 1 to C/C++/IDL Compiler, Dept. of Computer Science, Univ. of British Columbia, 1995.

[8] ISO/IEC DIS 8824 Specification of Abstract Syntax Notation One(ASN. 1), 2nd Ed., 1992.

[9] ISO/IEC DIS 8825 Specification of Basic Encoding Rule for Abstract Syntax Notation One(ASN.1), 2nd Ed., 1992.

[10] R. Steinmetz, K. Nahrstedt, Multimedia : Computing, Communications & Applications, Prentice Hall Inc., 1995.

[11] D. Solvoll, G. Ivars, and P. E. Dybvik, Information exchange in MultiTorg, http://www.nta.no/telektronikk/4.93.dir/Solvoll_D.html.

[12] K. Hofrichter, "MHEG5-Standardized Presentation Objects for the Set Top Unit Environment," in Proc. of European Workshop IDMS '96, Spring-Verlag, pp. 33-44, March, 1996.

[13] K. Hofrichter, H. W. Bitzer, GLUE-GLobal User Endsystem, http://www.fokus.gmd.de/ovma/glue/.

[14] B. S. Kaliski, Jr., A Laymans Guide to a Subset of ASN. 1, BER, and DER, RSA Data Security, Inc., 1991.

[15] H. Cossmann, C. Griwodz, G. Grassel, M. Puhlhofer, M. Schreiber, R. Steinmetz, H. Wittig, L. Wolf, "GLASS : A Distributed MHEG-based Multimedia System," in Proc. of Second International COST237 Workshop, Spring-Verlag, pp. 61-78, Nov., 1995.

[16] B. Heumann, T. Leidig, P. Rosch, "GLASS-Studio : An Open Authoring Environment for Distributed Multimedia Application," in Proc. of European Workshop IDMS'96, Spring-Verlag, pp. 45-58, March, 1996.

이 세 훈

1985년 인하대학교 전자계산학과 졸업(이학사)

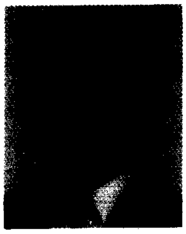
1987년 인하대학교 전자계산학과 대학원(이학석사)

1996년 인하대학교 전자계산학과 대학원(공학박사)

1987년~1990년 해병대 전산실 분석장교

1993년~현재 인하공업전문대학 전자계산기과 조교수

관심분야 : 멀티미디어/하이퍼미디어, 개방형 시스템, 소프트웨어 공학, 교육용 소프트웨어



왕 창 종

1964년 고려대학교 물리학과(학사)
1975년 성균관대학교(경영학 석사)
1981년~1990년 인하대학교 전자계산소장
1992년~1993년 정보과학회 부회장, 전산교육연구회 위원장
1979년~현재 인하대학교 전자계산공학과 교수
관심분야: 소프트웨어공학, 멀티미디어 컴퓨팅, 원격교육