11-2000

eXtensible MPEG-4 Textual Format (XMT)

Michelle KIM

Steve Wood

Lai-Tee CHEOK
Singapore Management University, LAITEECHEOK@smu.edu.sg

DOI: https://doi.org/10.1145/357744.357763

Extensible MPEG-4 Textual Format (XMT)

Michelle Kim
IBM T.J. Watson Research Center
30 Saw Mill River Rd
Hawthorne, NY 10532
(914) 784-7709
mykim@us.ibm.com

Steve Wood
IBM T.J. Watson Research Center
30 Saw Mill River Rd
Hawthorne, NY 10532
(914) 784-7309
woodsp@us.ibm.com

Lai-Tee Cheok
IBM T.J. Watson Research Center
30 Saw Mill River Rd
Hawthorne, NY 10532
(914) 784-7691
laitee@ee.columbia.edu

ABSTRACT

This paper describes the Extensible MPEG-4 Textual format (XMT), a framework for representing MPEG-4 scene description using a textual syntax. The XMT allows the content authors to exchange their content with other authors, tools or service providers, and facilitates interoperability with both the X3D, developed by the Web3D consortium, and the Synchronized Multimedia Integration Language (SMIL) from the W3C consortium.

Keywords

MPEG-4, textual format, scene description, authoring, X3D, SMIL

1. INTRODUCTION

MPEG-4 [1] is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) for communicating interactive audiovisual scenes. The standard defines a set of binary tools that provide the coded representation of individual audio-visual objects, text/graphics and synthetic objects. The interactive behaviors of these objects and the way they are composed in space and time to form an MPEG-4 scene is dependent on the scene description which is coded in a binary format known as BIFS (Binary Format for Scenes).

The MPEG-4 specification, in its binary form, basically provides a conformance point between the sender and the receiver of the content. As such, the coded form cannot be "reverse-engineered" in a consistent manner to represent the content author's original intentions.

The XMT has been designed to provide an exchangeable format between content authors whilst preserving the author's intentions in a high-level textual format. In addition to providing a suitable, author-friendly abstraction of the underlying MPEG-4 technologies, another important consideration for the XMT design was to respect existing practices of content authors such as the Web3D X3D, W3C SMIL and HTML.

2. TWO-TIER ARCHITECTURE: XMT-A AND XMT-Ω FORMATS

The XMT framework consists of two levels of textual syntax and semantics: the XMT-A format and the XMT-Ω format.

The XMT-A is an XML-based version of MPEG-4 content which closely mirrors its binary representation. The goal of the XMT-A format is to provide a deterministic one-to-one mapping to ISO/IEC 14496:1999 binary representations and to be interoperable with the X3D [2], which is being developed for VRML 200x (X3D) by the Web 3D Consortium. It contains a subset of the X3D, as well as the X3D-like representations of MPEG-4 specific features such as Object descriptors (OD), BIFS update commands and 2D composition. The ODs are used to associate scene description components to the actual elementary streams that contain the corresponding coded data while the BIFS update commands serve as a mechanism that allows a scene to be remotely manipulated, and portions of the scene to be progressively streamed in order to reduce bandwidth requirements.

The XMT-Ω is a high-level abstraction of MPEG-4 features designed based on the W3C SMIL [3], an XML-based language that allows authors to create dynamic, interactive multimedia presentations. Using SMIL, authors can describe the temporal behavior and layout of multimedia presentations, as well as associate hyperlinks with the media objects in the presentation.

For every XMT-Ω element, there is a mapping to a sequence of XMT-A elements. Note that there is no deterministic mapping between the two levels, for the obvious reason that a high-level author's intentions can be expanded to more than one sequence of low-level constructs. However, the XMT provides a standard mapping from XMT-Ω to XMT-A.

Moreover, for those authors who wish to control the implementation of certain portions of their presentation, the XMT provides an escape mechanism from XMT-Ω to XMT-A. The escape mechanism enables content authors to mix and match the two formats, XMT-Ω and XMT-A, overriding the default, standard mapping the XMT provides.

3. INTEROPERABILITY OF XMT

The XMT format can be interchangeable between SMIL player, Virtual Reality Modeling Language (VRML) [4] player, and MPEG-4 player. The XMT-Ω format can be preprocessed and

played directly by a SMIL player, preprocessed to the corresponding X3D nodes and played back by a VRML player, or compiled to an MPEG-4 representation such as mp4 (an exchangeable binary file format defined by MPEG-4), which can then be played by an MPEG-4 player. Figure 1 presents a graphical description of the interoperability of the XMT. MPEG-7 [5], as shown in the figure, is an emerging standard for describing multimedia content. The integration of MPEG-7 with XMT, which is currently work in progress, will enable the content-based retrieval of MPEG-4 objects.

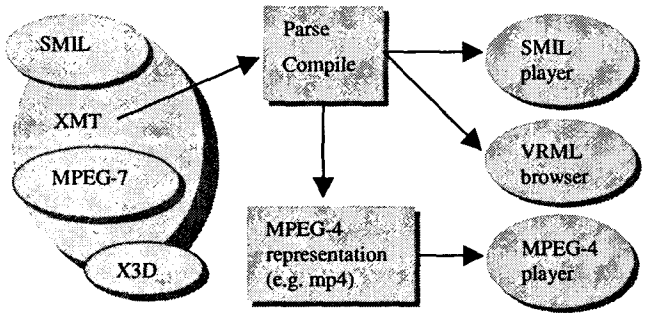

Figure 1. Interoperability of XMT

4. XMT-Ω FORMAT

The XMT-Ω format provides the ease of use and facilitates content interchange and interoperability with SMIL Boston, a follow on of SMIL 1.0. It is an effort by the W3C Synchronized Multimedia Working Group (SYMM WG) to partition SMIL functionality into sets of reusable modules. The XMT can be seen as an extension of SMIL.

In MPEG-4, objects are represented as nodes and their interactive behaviors described using routing mechanism that associates an event source with an event sink. The XMT-Ω format describes audio-visual objects and their relationships at a higher level, where content requirements are expressed in terms of the author's intent rather than by coding explicit node and route connections in MPEG-4. This permits authors to offer constructs at this level and to exchange them at this level with other authors. An authoring tool would compile this into MPEG-4 content by mapping the constructs into BIFS, OD, media streams, etc. with any appropriate compression or media conversions that may be required. Media sources in this format can be of a variety of formats native to the machine where the authoring tool is executing, and it is the responsibility of the tool, during the compilation phase, to convert media to suitable formats, bit-rates, and so on.

In this high-level format, there is not necessarily only one mapping for each authoring construct. MPEG-4 nodes and routes are very powerful tools and there can be more than one way to represent authoring constructs. Indeed, as MPEG-4 nodes can be 'wired' together with routes in many combinations, it is often difficult to reverse-engineer an author's intent from the nodes and routes. Faced with a presentation containing many nodes and routes, the re-authoring and maintenance of content can often prove to be challenging, as the high-level view of that presentation must be inferred. It is thus to provide such a high-level view and facilitate exchange and rapid content re-purposing or re-authoring that this textual format has been designed.

Recognizing though that some authors may wish to access low-level nodes and routes, this format allows the embedding of the XMT-A textual node and route definitions within an identified low-level nodes section. Interoperation between the two formats is also permitted.

4.1 Re-Using SMIL for XMT-Ω

SMIL is an XML-based language that allows authors to write interactive multimedia presentations. The main strengths of SMIL are that its constructs are self-describing, it is based on XML which provides an excellent format for interchange of data among different applications, it is relatively easy to author, and it is a language familiar to HTML users. It is also extensible so that new objects or meta-data can easily be inserted in the representation.

The XMT- Ω format provides a new set of elements that expresses the high-level view of MPEG-4, while re-using a subset of modules defined by SMIL where the semantics are compatible. It is not specifically designed as a playback format, and is to be preprocessed to SMIL, VRML, or compiled to an MPEG-4 representation, such as mp4, via appropriate translation software.

4.2 Extensible Media (xMedia) Objects

Although SMIL provides a useful abstraction for media objects in its Media Object Module, the concern was that SMIL is more about multimedia player composition rather than multimedia object composition. MPEG-4 is more focused on audio-visual (multimedia) objects. An example is that SMIL would render text and let the text player concern itself about font, style, kerning, colors etc., whereas an MPEG-4 text object includes font, style, alignment, and colors and as such is intimately aware of such detail. A combination of HTML+SMIL would include these text attributes but again, HTML has its own media content model.

MPEG-4 also contains 2D elements similar to those that are described in Scalable Vector Graphics (SVG) [6], such as the <Rectangle> and <Circle> elements. SVG in fact uses modules from SMIL, namely the Content Selection and Animation modules. The Animation module was jointly developed between SMIL and SVG.

The XMT- Ω syntax and semantics have been designed using extensible media (xMedia) objects as basic building blocks. The XMT- Ω defines an xMedia object by an element, such as <img> <rectangle> and <mesh>. An xMedia element abstracts geometry and behavior of the corresponding object, while representing the media specific attributes and timing attributes. An xMedia element also abstracts BIFS and OD commands, media streams, etc.

Behavior that are associated with an xMedia element can be defined by a set of animation and timing elements, e.g. <set attributeName="myRect.color" begin="click" to ="#ff00ee">. This example would map in MPEG-4 to a TouchSensor on the xMedia object that has a route that, via a valuator, sets the color of a Rectangle object, named myRect, to the color #ff00ee.

Spatial properties of an xMedia element can be further defined by a set of common child elements, such as,

<transformation>, <material>, <outline>, <chromakey>, <texture>, <light>, and <hotspots>.

Finally, events that are associated with an xMedia element (e.g. a mouse click) can be expressed using timing attributes as in SMIL, e.g. <video begin= "foo.click"/>. In this example the video will begin when (and if) the foo (button) is clicked. Elements and attributes defined in the XMT-Ω namespace is a superset of what is defined in SMIL.

4.3 Examples

4.3.1 Color Animation

Figure 2 provides a sample XMT-Ω fragment that demonstrates color change of a circle over a two-second duration when the mouse button is pressed. Using the XMT-Ω format the circle has been defined and an animate child element is added to it. The animate child element describes a linear interpolation using three-color values that starts on a mouse click event and has duration of 2 seconds.

```
<circle id="myCircle" radius="20">
  <transformation visibility="true"
                  translation="24 50"/>
  <material color="#ee0000" filled="true>"
     <animateColor attributeName="color"
                  dur="2s" begin="click"
                  values="#ee0000; #ffcc45;
                  #ffffff" keyTimes="0; 0.3; 1"
                  calcMode="linear" />
  </material>
</circle>
```

Figure 2. Sample XMT-Ω fragment for color animation

This XMT-Ω fragment can be mapped to MPEG-4 nodes using a VRML-like syntax as shown in Figure 3. The `Switch` and the `Choice` nodes are used for hiding and showing the circle based on the value of the `visibility` attribute in the XMT-Ω fragment. The `Transform2D` node contains a `translation` field for positioning the circle. The `Shape` node contains node for specifying the circle geometry and an `Appearance` and `Material2D` node for specifying the color of the circle. A `TouchSensor` is used to sense mouse activity while a `TimeSensor` is used to define the duration of the color change. To start the color animation, the `TouchSensor` is routed to the `TimeSensor` so that the `TimeSensor` is started when the mouse is pressed. The fractional output of the `TimeSensor` is then routed to the input of the `ColorInterpolator` and finally the output of the `ColorInterpolator` to the `emissiveColor` field of the `Material2D` node.

4.3.2 Slideshow

Figure 4 provides another sample XMT-Ω fragment to demonstrate the use of SMIL Timing and Synchronization module in XMT. The slideshow starts with an MPEG-1 video that is replaced by a JPEG image 50 seconds later. The <excl> and <par> elements are SMIL timing containers. The <excl> element ensures that only one of its <par> elements is "played" at any one time. The fragment is mapped to MPEG-4 nodes and commands as shown in Figure 5. The <video> and <img> elements are mapped to the `MovieTexture` and `ImageTexture` nodes respectively. The two groups of children nodes for the `choice` node contain positioning and source information each for the video and image.

```
Switch {
   whichChoice 0
   choice  [
    Transform2D  {
        translation 24 50
         children [
           Shape {
               geometry   DEF   myCircle   Circle
                  {radius 20}
               appearance Appearance {
                  material DEF CMat
                     Material2D {
                       EmissiveColor 0.93 0.0 0.0
                         filled TRUE
                     }
               }
            }
            DEF CI ColorInterpolator {
               key [0.0 0.3 1.0]
               keyValue[0.93 0.0 0.0,
                        1.0 0.93 0.27,
                        1.0 1.0 1.0 ]
            }
            DEF TS TouchSensor {}
            DEF T TimeSensor {cycleInterval 2}
        ]
       }
    ]
}
Route TS.touchTime to T.startTime
Route T.fraction_changed to CI.set_fraction
Route CI.value Changed to CMat.emissiveColor
```

Figure 3. Mapping of the sample XMT-Ω fragment for color animation to MPEG-4 nodes

```
<excl>
    <par begin="0s">
        <video url="c:\clips\video.mpg"
            <transformation translation="50 50"/>
    </video>
    </par>
    <par begin="50s">
        <img url="c:\clips\img.jpg"
            <transformation translation="100
100"/>          </img>
    </par>
</excl>
```

Figure 4. Sample XMT-Ω fragment for slideshow

```
DEF SW Switch {
   whichChoice 0
   choice  [
       # whichChoice=0 :
    Transform2D  {
       translation 50 50
        children [
          Shape {
              geometry Bitmap {}
              appearance Appearance {
                    texture MovieTexture {
                    url "c:\clips\video.mpg"}
              }
              }
          ]
       }
       # whichChoice=1:
    Transform2D  {
       translation 100 100
        children [
          Shape {
              geometry Bitmap {}
              appearance Appearance {
                    texture ImageTexture {
                     url "c:\clips\img.jpg"}
              }
              }
          ]
       }
   ]
}
AT 50 { REPLACE SW.whichChoice BY 1 }
```

Figure 5. Mapping of sample XMT-Ω fragment for slideshow to MPEG-4 nodes and command

The `MovieTexture` node is displayed at the beginning of the presentation. The command at the end of the `Switch` node causes the scene to be switched and replaced with the `ImageTexture` node at 50 seconds from the beginning of the presentation.

5. CONCLUSION

In this paper we described the Extensible MPEG-4 Textual Format (XMT) framework. The XMT framework consists of two levels of textual syntax and semantics: the XMT-A format, which provides a one-to-one deterministic mapping to MPEG-4 binary representation, and the XMT-Ω format which provides a high-level abstraction of XMT-A to content authors so they can exchange the content with other authors while preserving the original intent. The XMT-A provides interoperability between VRML and MPEG-4, and the XMT-Ω provides interoperability between SMIL and MPEG-4.

6. REFERENCES

[1] ISO/IEC FDIS 14496, Information Technology -- Generic Coding of Audio-Visual Objects – Part 1: System, Part 2: Visual, Part 3: Audio, Part 6: DMIF, International Organization for Standardization, 1998.

[2] ISO/IEC FDIS 14772:200x, Information Technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML).

[3] "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification" W3C Recommendation, http://www.w3.org/TR/REC-smil/ (June 1998).

[4] ISO/IEC FDIS 14772-1:1997, Information Technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification and UTF-8 encoding.

[5] "MPEG-7: Context, Objectives and Technical Roadmap, V.12", ISO/IEC JTC1/SC29/WG11 N2861 (July 1999).

[6] "Scalable Vector Graphics (SVG) 1.0 Specification", W3C Working Draft, http://www.w3.org/TR/2000/03/WD-SVG-20000303/index.html (March 2000).