

COMPONENTES INTERATIVOS PARA APRESENTAÇÕES EM MPEG-4

Tiago Nunes Tito dos Santos*

André Santanchè **

Resumo

O padrão MPEG-4 tem se difundido largamente entre os diversos dispositivos capazes de apresentar conteúdo digital multimídia. Apesar do maior uso do padrão MPEG-4 ser na apresentação de vídeos, este padrão permite a inserção de objetos complexos a serem apresentados em conjunto com o vídeo, incluindo software executável.

O presente artigo visa apresentar como um software executável interage com o sistema MPEG-4 e quais ferramentas (player, encoder e decoder) são utilizadas para gerar e executar este conteúdo multimídia.

Introdução

Com os diversos tipos de padrões abertos existentes no cenário atual, a exemplo dos formatos MPEG, está cada vez mais prático para autores de conteúdo, produzir, distribuir e consumir conteúdo digital.

A Evolução das tecnologias de redes e a disponibilidade de ferramentas para produção e modificação de conteúdo digital, facilitam a participação dos autores na produção e no consumo destes conteúdos. Por outro lado, deve se considerar a possibilidade de inserir nas apresentações multimídia, rotinas de software executável, principalmente quando existe o requisito de interatividade com o usuário. Tal Software interativo pode ser implementado utilizando um “Profile” definido pelo padrão MPEG-4, chamado MPEG-J.

* Curso de Sistemas de Informação da Universidade Salvador – UNIFACS

** Grupo de Pesquisa em Engenharia de Software e Aplicações (GESA/NUPERC)

MPEG-4

O Padrão MPEG-4 [ISO06] é um padrão de codificação, distribuição e apresentação de conteúdo multimídia. Ele tem sido adotado em diferentes dispositivos que apresentam conteúdo multimídia, tais como, computador, TVDI, dispositivos móveis etc.

O MPEG-4 utiliza o conceito de “Objetos de Media”, ou seja, um conteúdo multimídia é composto de objetos naturais, vídeo/áudio gravados por um câmera/microfone ou objetos sintéticos, objetos gerados por computador (ex.: cubo 2D). Para diferenciar estes objetos uns dos outros numa cena multimídia, é utilizado um descritor de cena.

Descritor de Cena

O Descritor de cena, é uma descrição textual dos objetos multimídia, relacionando o seu comportamento espacial e temporal. Por exemplo, em uma cena composta por um vídeo, áudio e uma imagem gerada por computador, o autor descreveria em que momento o vídeo seria iniciado (comportamento temporal), a posição que seria apresentada na tela (comportamento espacial) e que a imagem computadorizada é um retângulo ou um cubo.

A Descrição da cena é feita seguindo o conceito da linguagem VRML (Virtual Reality Modeling Language) [SIGN00]. Mas, a linguagem não oferece suporte a algumas características, como streaming de áudio e vídeo. Para suprir essas carências o padrão MPEG-4 define uma nova representação, que herda e estende o VRML, chamado de BIFS (Binary Format for Scene) [GOUL03].

O BIFS é a descrição textual da cena em um formato binário, está estruturado de forma hierárquica e pode ser representado como um grafo.

A Figura 1 mostra uma cena multimídia composta por um retângulo e uma imagem JPEG.

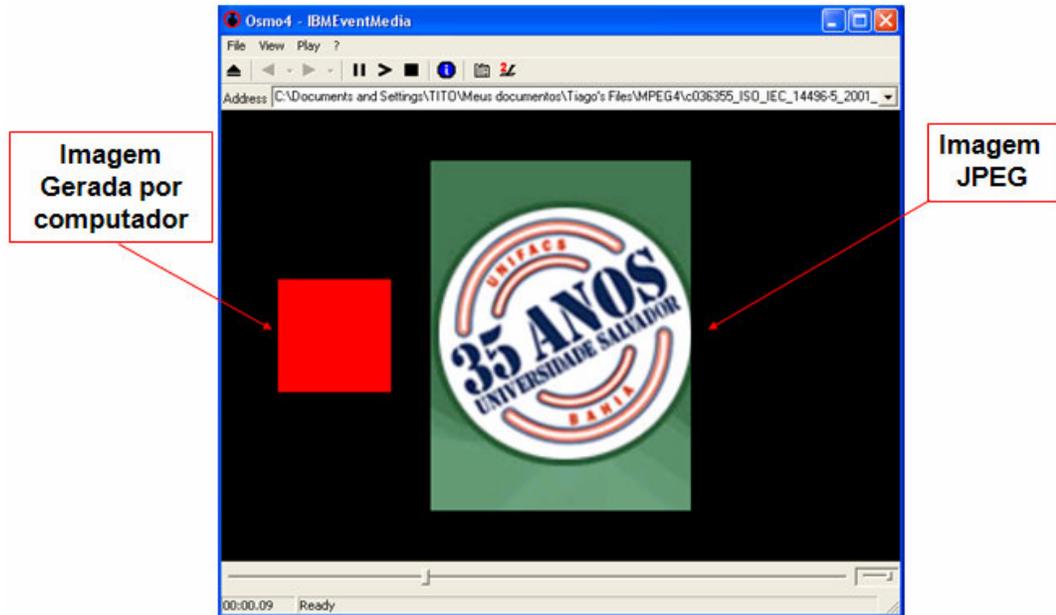


Figura 1 – Composição de uma cena multimídia

A Figura 2 representa a descrição da cena da figura 1 seguindo a sintaxe VRML estendida.

```

Group
{
  children
  [
    Transform2D {
      translation 50 50
      children [
        DEF ID0 Shape {
          appearance Appearance {
            material Material2D {
              emissivecolor 0 1 0
            }
          }
          geometry Rectangle {
            size 200 100
          }
        }
      ] ] }

    Transform2D {
      translation 0 0
      children [
        DEF ID1 Shape {
          appearance Appearance {
            material Material{
            }
          }
          texture ImageTexture {
            url 3
          }
          geometry Bitmap {}
        }
      ] ] }

  UPDATE OD
  [
    {
      objectDescriptorID 3
    }
    {
      objectDescriptorID 4
    }
  ]
}
  
```

Figura 2 - Descrição de uma cena multimídia

A Descrição Textual é composta basicamente por nós. Cada nó possui um tipo (Shape, Box, Sound, Transform, Appearance, Anchor etc.) , onde o nome ou tipo do nó,

indica o objeto que ele descreve (formato, iluminação, som, grupos, sensores etc.) [RODR03].

Analisando a figura 2, percebemos que a estrutura começa com o nó GROUP (nó raiz) e em seguida os seus correspondentes nós filhos.. No nó Transformed 2D, seus atributos determinam o comportamento espacial do objeto. No nó Shape vem a inscrição "DEF", que serve para identificar o nó. Portanto se fossemos manipulá-lo, acessaríamos o BIFS informando o ID do nó, por exemplo, ID1, e assim alterar alguma característica ou até mesmo deletar o nó.

O campo UPDATE OD referencia o Descritor de objetos (Object Descriptor – OD), que se encarrega de descrever, entre outras coisa, os descritores de fluxo presentes. Por exemplo, o "objectDescriptorID 3" está descrito no arquivo sceneapitest.scr e referisse a imagem JPEG.

Descritor Inicial de Objetos

O Padrão MPEG-4 trata as os dados como fluxos elementares (Elementar Streams - ES), portanto todo objeto multimídia (vídeo, áudio, imagem JPEG) são ESs [GOUL03]. Quando um player vai iniciar a execução do conteúdo multimídia, ele precisa ser informado em qual fluxo encontrar o descritor de cena e o descritor de objetos. Assim, a primeira informação a ser executada pelo player é o Descritor Inicial de Objetos (Inicial Object Descriptor – IOD).

O IOD, através do campo "ES_ID" consegue identificar qual fluxo corresponde ao descritor de cena e ao descritor de objetos, como mostra a figura 3.

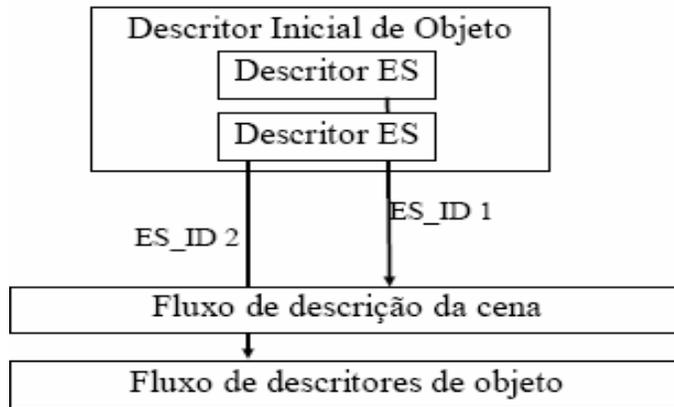


Figura 3 - Descritor Inicial de Objetos [GOUL03].

De posse destas informações, o player executa o fluxo referente ao BIFS para montar a cena. Como foi explicado anteriormente, o descritor de cena possui uma referência do fluxo elementar que corresponde ao objeto que deseja exibir. Portanto, através do campo “objectDescriptorID”, o player executa o fluxo referente aos objetos multimídia que utilizará na cena.

MPEG-J

O MPEG-4 especifica como uma aplicação Java deve ser recebida, decodificada e executada pelo player do terminal cliente. A parte do sistema que define estas especificações é o Profile MPEG-J [SWAM05].

Uma Aplicação MPEG-J (MPEGlet) funciona de forma semelhante a um applet Java, servindo-se da JVM (Java Virtual Machine) existente na máquina cliente. Com a utilização da API MPEG-J a aplicação dispõe de recursos como, por exemplo, acesso a cena gráfica, para manipular uma cena, acesso a características dinâmicas do player, a rede e ao controle dos decodificadores [TODE00].

Um MPEGlet após compilado, gera como saída um Java Class. Normalmente as classes são compactadas no formato “ZIP” ou “JAR”, sendo vinculado a eles um cabeçalho informando identificação, tipo, compressão, pacote e informação de dependência de classe. O Pacote é recebido pelo Player da mesma forma que um vídeo ou áudio, como um fluxo elementar [GOUL03]. Um Descritor de Objetos detém

informações de quando as classes devem ser carregadas e quais recursos devem ser alocados para executá-las.

Investigação de Ferramentas

Foi feita uma investigação de um conjunto de ferramentas (player, encoder e decoder) que pudessem ser utilizadas para gerar e executar conteúdo multimídia MPEG-4. Tais Ferramentas devem oferecer o suporte para autoria da descrição de cena, realizando a verificação de erros no código (debug) e codificando-o no formato BIFS, fazer o “debug” do MPEGlet e compactar suas classes e objetos e os empacotando, gerar os fluxos elementares dos objetos e por fim multiplexá-los.

Foi testado o conjunto de ferramentas disponibilizada pela IBM, o “IBMTToolkitForMpeg4”. Desenvolvido na linguagem Java, contém 3 ferramentas:

AVgen, que cria conteúdo áudio/vídeo para ISMA ou dispositivos 3GP. •

XmtBatch, gera conteúdo a partir da multiplexação do descritor de cena (na linguagem XMT), com uma amostra de vídeo ou áudio.

M4Play, aplicação (player) para executar o conteúdo MPEG-4.

A Figura 3 representa uma aplicação escrita em XMT, onde a partir do clique do usuário em uma das imagens é disparado um evento. Por exemplo, ao clicar no retângulo faz com que o triângulo fique de ponta cabeça.

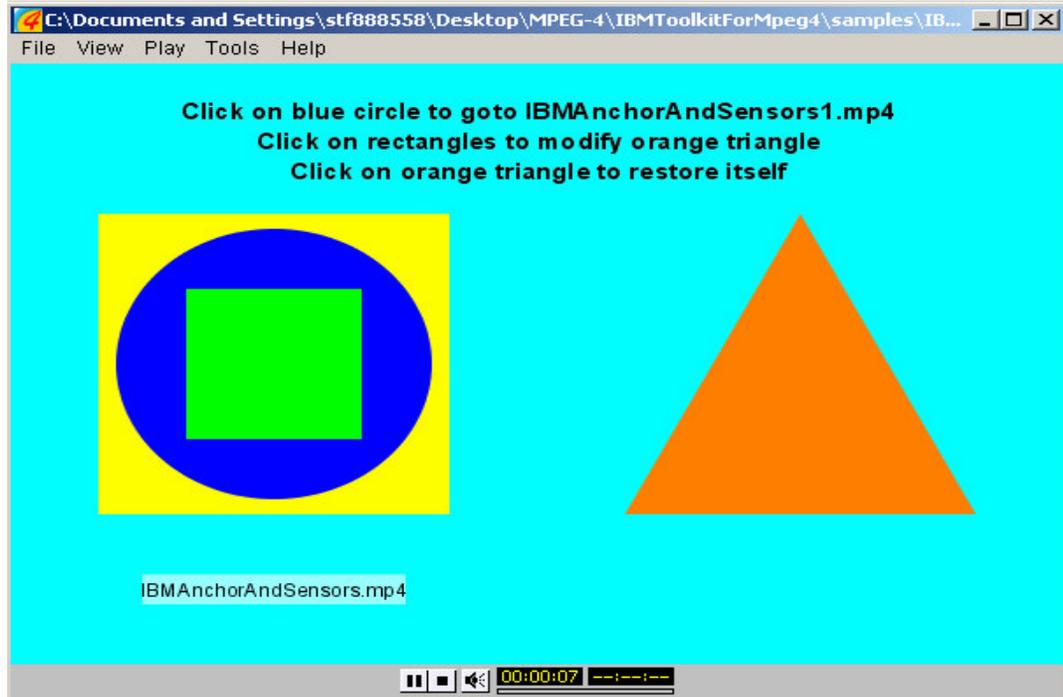


Figura 4 – Aplicação XMT

Os testes realizados com o Player foram bem sucedidos. Protótipos que utilizavam vídeo H263, áudio AAC e imagens JPEG não apresentaram erros de execução.

Apesar do bom aproveitamento da ferramenta, ela não oferece suporte para aplicações MPEG-J, inviabilizando a utilização de código Java em conjunto ao MPEG-4.

O Software IM1 disponibilizado pela ISO (International Organization for Standardization), é oferecido em várias versões. A maioria das ferramentas testadas continha erros no código fonte, o que dificultava a realização de testes.

O Pacote “c039417_ISO_IEC_14496-5_2001_Amd_4_2004” foi o que continha o menor índice de ”bugs”, retornando os melhores resultados.

Implementado em C ++, depende da instalação de outros softwares para sua execução. Abaixo segue uma listagem dos softwares:

- Microsoft Visual C++ 6.0.
- Visual C++ Service Pack 5 e Processor Pack.
- DirectX SDK
- JDK 1.4.2

A partir da instalação destes softwares, poderá realizar a compilação dos projetos, gerando as ferramentas BifsEnc, Mp4Enc e IM1 player.

Com a ferramenta “MP4Enc”, foi possível gerar arquivos multimídia de extensão “mp4” a partir da multiplexação de todos os fluxos elementares referenciados pelo descritor de fluxos, por exemplo, vídeo, áudio, descrição da cena, descritor de objetos e aplicação mpeg-j. Executando o MP4Enc, passa-se como parâmetro o arquivo que corresponde ao BIFS e outro de extensão “scr”, que contem informações sobre os fluxos elementares (equivalente ao IOD e o OD). As Figuras 4 e 5 representam a descrição do fluxo OD, BIFS, áudio G723 e aplicação MPEG-J, informando nome do arquivo, formato e tipo do fluxo, tamanho do buffer e suas configurações de codificação.

```

InitialObjectDescriptor {
  objectDescriptorID      0
  ODProfileLevelIndication 1
  sceneProfileLevelIndication 1
  audioProfileLevelIndication 2
  visualProfileLevelIndication 1
  graphicsProfileLevelIndication 1
  esDescr {
    ES_ID 1
    muxInfo {
      fileName sceneapitest.od
      streamFormat BIFS
    }
    streamPriority 5
    decConfigDescr {
      streamType 1 // OD Stream
      bufferSizeDB 200
    }
    slConfigDescr SLConfigDescriptor{
      useAccessUnitStartFlag TRUE
      useAccessUnitEndFlag TRUE
      useRandomAccessPointFlag TRUE
      useTimestampsFlag TRUE
      timeStampResolution 1000
      timeStampLength 14
    }
  }

  ES_ID 2
  muxInfo {
    fileName sceneapitest.bif
    streamFormat BIFS
  }
  streamPriority 5
  decConfigDescr {
    streamType 3 // BIFS Stream
    bufferSizeDB 1000
    decSpecificInfo BIFSConfig {
      nodeIDbits 10
      routeIDbits 10
      isCommandStream TRUE
      pixelMetric TRUE
      pixelWidth 352
      pixelHeight 288
    }
  }
  slConfigDescr SLConfigDescriptor{
    useAccessUnitStartFlag TRUE
    useAccessUnitEndFlag TRUE
    useRandomAccessPointFlag TRUE
    useTimestampsFlag TRUE
    timeStampResolution 100
    timeStampLength 14
    //OCR_ES_Id 1
  }
}
}

```

Figura 5 - Descrição do fluxo OD e BIFS.

```

{
  objectDescriptorID 3
  esDescr {
    ES_ID 2113
    muxInfo {
      fileName ..\audio\linda.g723
      streamFormat G723
      streamAttr "5.3"
    }
    streamPriority 4
    decConfigDescr {
      streamType 5 // Audiostream
      objectTypeIndication 0xc1 // G723
      bufferSizeDB 300
      decSpecificInfo DecoderspecificInfostring {
        info "obsolete string"
      }
    }
    slConfigDescr SLConfigDescriptor{
      timestampResolution 1000
      compositionUnitDuration 30
      OCRRResolution 1000
      OCRLength 24
    }
  }
}
{
  objectDescriptorID 4
  esDescr {
    ES_ID 2114
    muxInfo {
      fileName JSJ
      streamFormat MPEGJ
      streamAttr "t.info"
    }
    decConfigDescr {
      streamType 9 // MPEGJstream
      objectTypeIndication 0xc2 // MPEGJ
      bufferSizeDB 216000
      decSpecificInfo DecoderspecificInfostring {
        info "obsolete string"
      }
    }
    slConfigDescr SLConfigDescriptor{
      useAccessUnitStartFlag TRUE
      useAccessUnitEndFlag TRUE
      useRandomAccessPointFlag TRUE
      useTimestampsFlag TRUE
      timestampResolution 1000
      timestampLength 10
      packetSeqNumLength 3
      AU_seqNumLength 8
      //OCR_ES_Id 2113
    }
  }
}

```

Figura 6 - Descrição do fluxo de áudio e MPEG-J.

Utilizando o IM1 Player foi possível executar o conteúdo multimídia, tendo o suporte de outras ferramentas para fazer o “debug” da descrição de cena e do MPEGlet.

Considerações Finais

Neste artigo foram passadas as características principais do padrão MPEG-4, abordando como um conteúdo multimídia é gerado. Foram detalhados os passos que um autor de conteúdo deve seguir e quais ferramentas utilizar para a realização do seu trabalho.

Fica em destaque, a possibilidade de desenvolver softwares na linguagem Java que interajam com o sistema MPEG-4. Tal evolução, traz a vantagem de desenvolver aplicações complexas capazes de prover interatividade entre usuário e servidor ou com outros usuários.

Bibliografia

[GOUL03] GOULARTE, Rudinei. Personalização e adaptação de conteúdo baseadas em contexto para TV Interativa. Tese (Doutorado) – ICMC-USP, 2003.

[HERP02] HERPEL, C. e ELEFTHERIADIS, A. MPEG-4 Systems: Elementary Stream Management. Dept. of Electrical Engineering, Columbia University. New York, 2000. Disponível em <http://www.chiariglione.org/mpeg/tutorials/papers/icj-mpeg4-si/04-ESM_paper/4-ESM_paper.htm>. Acesso em: outubro de 2007.

[ISO06] ISO/IEC. Overview of the MPEG-4 Standard (ISO/IEC JTC1/SC29/WG11 N4668). Disponível em: <<http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm>>. Acesso em: 16 nov. 2006.

[JAVA07] The Java TV Technology Home Page <<http://java.sun.com/products/javatv/>>. Acesso em: maio 2007.

[MONT02] MONTEIRO, Marcelo Souto. TV interativa e seus caminhos. Ed. Campinas, SP: 2002.

[RODR03.2] RODRIGUES, Paula MPEG-4: Linguagens para Descrição de Cenas. Monografia Apresentada na Disciplina “Fundamentos de Sistemas Multimídia”, Pontifícia Universidade Católica, 2º Semestre de 2003.

[SANT06.1] SANTANCHÈ, André. Fluid Web e Componentes de Conteúdo Digital: da visão centrada em documentos para a visão centrada em conteúdo. 2006. 1 v. Tese (Doutorado) – Unicamp, Campinas, 2006.

[SANT06.2] SANTOS, Tiago Nunes Tito. Estudo da Integração de Dispositivos Móveis em Aplicações de TV Digital e Interativa. Projeto de Iniciação Científica, 2006.

[SANT07.1] SANTANCHÈ, André e MEDEIROS, Claudia Bauzer. User-author centered multimedia building blocks. *Multimedia Systems*, 12(4):403–421, Março de 2007.

[SANT07.2] SANTANCHÈ, André e MEDEIROS, Claudia Bauzer. A Component Model and Infrastructure for a Fluid Web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341, Fevereiro de 2007.

[SIGN00] SIGNÈS, J. FISHER, Y. e ELEFTHERIADIS, A. MPEG-4's Binary Format For Scene Description. *Signal Processing: Image Communication*, Volume 15, Number 4, January 2000, pp. 321-345(25) Disponível em <http://www.chiariglione.org/mpeg/tutorials/papers/icj-mpeg4-si/05-BIFS_paper/5-BIFS_paper.htm>. Acesso em: setembro de 2007.

[SWAM05] SWAMINATHAN, V. e BOURGES-SÉVENIER, M. MPEG-J White Paper. Disponível em <<http://www.chiariglione.org/mpeg/technologies/mp04-j/>>. Acesso em: setembro de 2007.

[TAKA00] TAKAHASHI, Tadao. Sociedade da Informação no Brasil – Livro Verde. Brasília, 2000.

[TODE00] TODESCO, Glauco. Criação de Ambientes Virtuais Multiusuário através do Padrão MPEG-4. Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos, 2000.