

## 7

### Referências Bibliográficas

ARIB. **ARIB STD-B24, Version 3.2, Volume 3: Data Coding and Transmission Specification for Digital Broadcasting**, ARIB Standard, 2002.

ARIB. **ARIB STD-B23, Application Execution Engine Platform for Digital Broadcasting**. ARIB Standard, 2004.

ATSC. **DTV Application Software Environment Level 1 (DASE-1) PART 1: Introduction, Architecture, and Common Facilities**. ATSC Standard. Março de 2003.

ATSC. **DTV Application Software Environment Level 1 (DASE-1) PART 2: Declarative Applications and Environment**. ATSC Standard, Março de 2003.

ATSC. **Advanced Common Application Platform (ACAP)**, A/101. Agosto de 2005.

BDA, **Blue-Ray Disc Format – 2.B Audio Visual Application Format Specifications for BD-ROM**. White paper. Março de 2005.

BERNERS-LEE, T. J. **The World-Wide Web**. Communications of the ACM, v. 37, n. 8, Agosto de 1994, p. 76-82.

BULTEMAN, D.; RUTLEDGE, L. **SMIL 2.0: Interactive Multimedia for Web and Mobile Devices**, Springer, 2004.

BROWN, A. W.; WALLNAU, C. K. **International Workshop on Component-Based Software Engineering**. International Conference on Software Engineering '99 – Los Angeles. 1999.

ECMA Standardizing Information and Communication Systems. **ECMAScript Language Specification**, Standard ECMA 262, 3rd Edition, 1999.

ETSI. Digital Video Broadcasting (DVB), **Globally Executable MHP (GEM)**, ETSI TS 102 819 v 1.2.1, Maio de 2004.

ETSI. Digital Video Broadcasting (DVB), **Multimedia Home Platform (MHP)**

**Specification 1.1.1**, ETSI TS 102 812, Maio de 2005. Disponível em: <[www.mhp.org/](http://www.mhp.org/)>. Acesso em 12/03/2007.

FERRETTI, S.; ROCCETTI, M.; ANDRICH, J. **Living the TV Revolution: Unite MHP to the Web or Face IDTV Irrelevance!** 15th International Conference on World Wide Web, p.899-900, 2006.

FUHRHOP, C.; HU, J.; GAFSOU, O. **Prototype of Applications**, User Interfaces and End User Tools, IST-2000-28298: ICE-CREAM. Abril de 2003.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J., **Padrões de Projeto**, Editora Bookman, Porto Alegre, 2002.

GEARY, D. M., MCCLELLAN, A. L. **Graphic Java: mastering the AWT**. SunSoft, Prentice Hall, 1997.

GOSLING, J.; MCGILTON H. **The Java Language Environment**. Maio de 1996. Disponível em: <<http://java.sun.com/docs/white/langenv/>>. Acesso em 15/01/2007.

FILHO, G. L. S. F.; LEITE, L. E. C.; BATISTA, C. E. C. F. **Ginga-J: The Procedural Middleware for the Brazilian Digital TV System**. Journal of The Brazilian Computer Society, Março de 2007, p. 47-56.

HAVi Organization, **HAVi-Home Audio/Video Interoperability**. 1999, Disponível em: <<http://www.havi.org>>. Acesso em 20/10/2005.

IERUSALIMSCHY et al. **Lua 5.0 Reference Manual**. Technical Report MCC - 14/03, PUC-Rio, 2003.

ISO/IEC 11172-1. **Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 1: Systems**. ISO Standard, 1993.

ISO/IEC 13522-5. **Information technology - Coding of multimedia and hypermedia information -- Part 5: Support for base-level interactive applications**. ISO Standard, 1997.

ISO/IEC 13818-6. **Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC**. ISO Standard, 1998.

- ITU-T. **Worldwide common core – Application environment for digital interactive television services**. ITU-T J.200. Março de 2001.
- ITU-T. **Harmonization of procedural content formats for interactive TV applications**. ITU-T J.202. Maio de 2003.
- ITU-T. **Harmonization of declarative content format for interactive television applications**. ITU-T J.201. Julho de 2004.
- LAMANDON, J. L.; CESAR, P.; HERRERO, C.; VUORIMAA, P. “**Usages of a SMIL Player in Digital TV**” VII IASTED International Conference on Internet and Multimedia Systems and Applications, Agosto de 2003, p. 579-584.
- LARMAN, C., **Utilizando UML e Padrões – Uma Introdução à análise e ao Projeto Orientados a Objetos**. Editora Bookman, 2002.
- LIANG, S.; BRACHA, G. **Dynamic Class Loading in the Java Virtual Machine**. Conference on Object-oriented programming, systems, languages, and applications (OOPSLA'98), Vancouver, Canadá, Outubro de 1998.
- MHP, **Who is using MHP?** A look around the world. Setembro de 2006. Disponível em: <[http://www.mhp.org/about\\_mhp/who\\_is\\_using\\_mhp](http://www.mhp.org/about_mhp/who_is_using_mhp)>. Acesso em: 14/01/2007.
- OPENCABLE, **Opencable Application Platform Specification - OCAP 1.0 Profile – OC-SP-OCAP1.0-I16-050803**. Issued Specification, Agosto de 2005.
- PENG, C.; VUORIMAA, P. **Digital Television Application Manager**. IEEE International Conference on Multimedia and Expo 2001, Tóquio, Japão, Agosto de 2001.
- PERROT, . **DVB-HTML: an optional declarative language within MHP 1.1**. EBU Technical Review. Setembro de 2001.
- RFC. Request for Comments: 3986. **Uniform Resource Identifier (URI): Generic Syntax**. 2005.
- ROBINSON, M; VOROBIEV, P. **Swing**. 2ed. Manning, 2003.

- RODRIGUES, R.F., **Formatação e Controle de Apresentações Hiperídia com Mecanismos de Adaptação Temporal**. Tese (Doutorado em Informática) – PUC-RJ, Rio de Janeiro 2003.
- SCHEIFLER, R. W.; GETTYS, J. **X Window System: Core and extension protocols: X version 11, releases 6 and 6.1**. Digital Press, 1996.
- SCHWALB E. M. **iTV Handbook - Technologies and Standards**, IMSC Press Multimedia Series, Prentice Hall PTR, 2004.
- SHEPPARD, Bill; MOLL, E. **Java Technology Goes to the Movies: Java Technology in Next-Generation Optical Disc Formats**. 2006 JavaOne Conference, Session TS-1685, Maio de 2006.
- SZYPERSKI, C. **Component Software**. 2ed. Addison-Wesley, 2002, 446p.
- SOARES, L.F.G.; RODRIGUES, R.F.; MUCHALUAT-SAADE, D.C. **Modelo de Contexto Aninhados – versão 3.0**. Relatório Técnico, Laboratório Telemídia PUC-Rio, Março de 2003.
- SOARES, L.F.G.; RODRIGUES, R.F. **Nested Context Language 3.0 Part 8 – NCL Digital TV Profiles**. Monografias em Ciências da Computação, PUC-Rio, Outubro de 2006.
- SOARES, L.F.G. et al, **Nested Context Language 3.0 Part 9 – NCL Live Editing Commands**. Monografias em Ciências da Computação, PUC-Rio, Dezembro de 2006.
- SOARES, L.F.G.; RODRIGUES, R.F.; **Mecanismo de Identificação de Recursos para Aplicações Interativas em Redes de TV Digital por Difusão**. 25º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Maio de 2007.
- SUN MICROSYSTEMS. **Applets**. 1994. Disponível em: <<http://java.sun.com/applets>>. Acessado em: 15/01/2007.
- SUN MICROSYSTEMS. **Java Media Framework, v2.0 API Specification**. 1999. Disponível em: <<http://java.sun.com/products/java-media/jmf/>>. Acessado em: 15/01/2007.

- SUN MICROSYSTEMS. **Object Serialization**. 1999. Disponível em: <<http://java.sun.com/j2se/1.3/docs/guide/serialization>>. Acessado em: 17/05/2007.
- SUN MICROSYSTEMS. **Permissions in the Java 2 SDK**. 2002. Disponível em: <<http://java.sun.com/j2se/1.4.2/docs/guide/security/permissions.html>>. Acessado em: 17/01/2007.
- SUN MICROSYSTEMS. **CDC: Java Platform Technology for Connected Devices**. Java Platform Micro Edition White Paper. Junho de 2005.
- SUN MICROSYSTEMS. **CDC Runtime Guide**. Reference Implementation. Agosto de 2005.
- TUCS; AXEL Technologies. **OpenMHP: Multimedia Home Platform**. 2004. Disponível em: <<http://www.openmhp.org>>. Acessado em: 15/01/2007.
- W3C. **Cascading Style Sheets, level 2 - CSS 2 Specification**, W3C Recommendation, 1998.
- W3C. **HTML 4.01 Specification**. W3C Recommendation, Dezembro de 1999.
- W3C. **XHTML 1.0 The Extensible HyperText Markup Language (Second Edition) – A Reformulation of HTML 4 in XML 1.0**. W3C Recommendation, Agosto de 2002.
- W3C. **Document Object Model (DOM) Level 3 Core Specification**. W3C Recommendation, Abril 2004.
- W3C. **XML Schema Part 0: Primer Second Edition**. W3C Recommendation, Outubro de 2004.
- W3C. **Synchronized Multimedia Integration Language (SMIL 2.1)**. W3C Recommendation, Dezembro de 2005.
- WEBER, J. W.; NEWBERRY, T. **IPTV Crash Course**. McGraw-Hill, 2006.

## Apêndice A

### Assinatura Digital de um *Xlet*

A execução de aplicações em receptores de TV digital domésticos permite o acesso a recursos em tais equipamentos que, no caso de um comprometimento de sua segurança, podem expor dados sigilosos de seus usuários. Para restringir o acesso a tais recursos sensíveis do receptor, foram criados mecanismos que permitissem a sua utilização por aplicações confiáveis.

O uso dos Arquivos de Requisição de Permissão (PRFs) é um mecanismo criado para permitir o acesso pelas aplicações a certos recursos sensíveis dos receptores em termos de falhas de segurança. Esses recursos incluem, por exemplo, acesso ao disco local, às preferências dos usuários, ao uso do canal de retorno, entre outros. Vale lembrar que o acesso a tais ativos deve ser controlado devido ao fato de as aplicações não necessariamente localizarem-se apenas no fluxo de transporte proveniente das difusoras; elas podem ser obtidas também via canal de retorno por provedores de conteúdo das mais diversas origens.

A especificação do GEM prevê que um receptor que suporte tal *framework* seja capaz, no mínimo, de processar os PRFs criados pelo padrão MHP. A Figura 37 ilustra um exemplo de arquivo e configuração que respeita tal padrão. O arquivo do exemplo permite a leitura das preferências do usuário; o uso do canal de retorno; bloqueia a sintonização; permite a seleção do serviço; permite a leitura e escrita de dois arquivos; entre outros.

```
<?xml version="1.0"?>
<!DOCTYPE permissionrequestfile PUBLIC "-//DVB//DTD Permission
Request File 1.0//EN"
"http://www.dvb.org/mhp/dtd/permissionrequestfile-1-0.dtd">
<permissionrequestfile orgid="0x000023d2" appid="0x4020">
  <file value="true"></file>
  <applifecyclecontrol value="true"></applifecyclecontrol>
  <returnchannel>
    <defaultisp></defaultisp>
    <phonenumber>+552135211500</phonenumber>
    <phonenumber>+552135211501</phonenumber>
    <phonenumber></phonenumber>
  </returnchannel>
```

```
<tuning value="false"></tuning>
<servicesel value="true"></servicesel>
<userpreferences read="true" write="false"></userpreferences>
<network>
  <host action="connect">hostname</host>
</network>
<persistentfilecredential>
  <grantoridentifier id="0x000023d2"></grantoridentifier>
  <expirationdate date="24/12/2032"></expirationdate>
  <filename read="true"
write="true">23d2/4020/formatterConfig/context/context.ini</filena
me>
  <filename read="true"
write="true">23d2/4020/formatterConfig/players/mimedefs.ini</filen
ame>
  <filename read="true"
write="true">23d2/4020/formatterConfig/players/ctrldefs.ini</filen
ame>
  <signature>023203293292932932921493143929423943294239432
</signature>
  <certchainfileid>3</certchainfileid>
</persistentfilecredential>
</permissionrequestfile>
```

Figura 37 – Exemplo de arquivo PRF.

O arquivo PRF permite ainda a leitura de três arquivos localizados no dispositivo de armazenamento local no receptor: *context.ini*, *mimedefs.ini* e *ctrlrdefs.ini*.

Os dois diretórios iniciais do caminho de acesso a tais arquivos consistem do identificador da organização e o identificador da aplicação responsáveis por realizar sua gravação em disco. Isso é parte de uma padronização proposta pelo MHP e absorvida pelo GEM que permite a uma aplicação criar arquivos apenas em diretórios que contenham em seu caminho inicialmente tais identificadores.

Vale ressaltar que é permitido que diferentes organizações/aplicações acessem um arquivo gravado em disco. Contudo, é necessário apresentar no arquivo PRF o identificador da organização responsável pela gravação original do arquivo (através do elemento “*grantoridentifier*”) e uma assinatura (através do elemento “*signature*”). Com isso, o receptor terá garantias que esta nova aplicação tem permissão de ler/alterar um arquivo que não foi originalmente por ela criado.

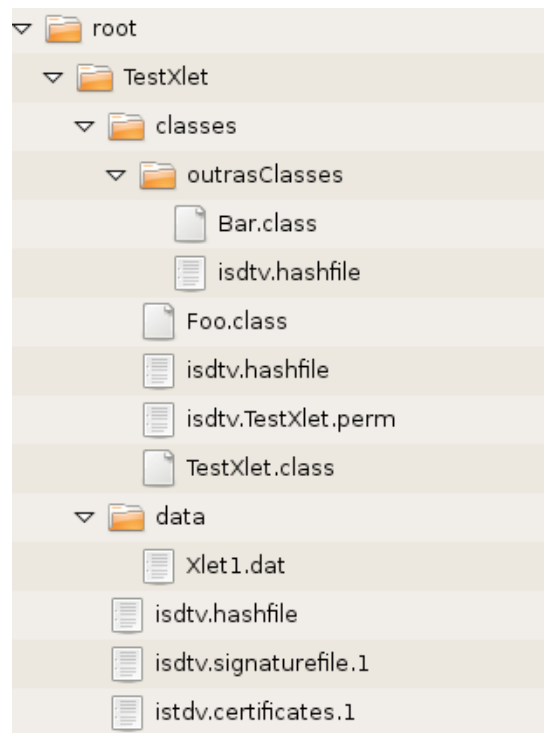


Figura 38 – Estrutura de diretórios de um Xlet.

Uma aplicação corretamente certificada possui uma estrutura semelhante à mostrada na Figura 38:

- O PRF localiza-se no mesmo diretório da classe inicial (que no exemplo é *TestXlet.class* e o arquivo PRF é *isdtv.TestXlet.perm*); e
- Todos os diretórios que tiverem seu conteúdo assinado possuirão um arquivo contendo uma assinatura do tipo hash de todo ele (utilizando o algoritmo MD5, por exemplo). No exemplo da Figura 38, o diretório “data” não tem seu conteúdo assinado.



## Apêndice B

### Sinalização de Aplicações

O processo de sinalização de aplicações recomendado pelo GEM e adotado pelo MHP envolve o envio periódico de uma tabela denominada Tabela de Informação de Aplicações (AIT – *Application Information Table*). Essas tabelas constituem um fluxo elementar do fluxo de transporte MPEG-2. Elas são um tipo de PSI (*Program Specific Information*). Esse apêndice explica os principais atributos presentes na AIT e relevantes para a implementação de uma aplicação DVB-J.

As AITs são identificadas na PMT como um fluxo do tipo 0x05. Além disso, opcionalmente, a PMT pode conter um descritor de sinalização contendo o tipo de aplicação e a versão da AIT. Dessa forma, o processo de sinalização fica otimizado pois, receptores que não possuam capacidade de processar um determinado tipo de aplicação não necessitam decodificar toda a AIT, e mudanças na AIT são mais facilmente detectadas através do indicador de versão.

Cada tabela AIT possui, principalmente, zero ou mais descritores comuns, contendo informações compartilhadas por todas as aplicações nela descritas, uma lista com os descritores de informações específicas de cada uma dessas aplicações, além dos campos tipo da aplicação, os códigos de controle da aplicação, os identificadores das aplicações e a versão da tabela.

O campo “**tipo de aplicação**” leva o número 0x0001 para aplicações DVB-J.

O campo “**versão da tabela**” funciona como uma otimização para que uma atualização em seu conteúdo possa ser mais facilmente detectada.

O “**código de controle**” serve para controlar o ciclo de vida a aplicação. O que significa dizer que através dele a emissora pode mandar comandos para encerrar uma aplicação, auto iniciar ou apenas informar a sua disponibilidade para o usuário.

Os “**identificadores das aplicações**” são um par contendo o identificador da organização que produziu a aplicação e o identificado da aplicação.

A Tabela 4 mostra os principais campos, citados anteriormente, e seus respectivos valores a fim de exemplificar uma AIT para o carregamento do Formatador NCL.

Tipo de Aplicação	0x0001 (DVB-J)
Versão	0x0
<i>Descritores comuns()</i>	
for(i=0;i<N;i++){	
Identificador da aplicação	0x00001111 . 0x4001
Código de controle	0x01 (Auto iniciar)
<i>Descritores da aplicação()</i>	
}	

Tabela 4 – Exemplo resumido de uma tabela AIT.

Dentre os descritores possíveis para a AIT destacam-se aqueles descritos, de forma consolidada, na Tabela 5. São eles:

- O descritor geral da aplicação que informa o perfil ao qual aquela aplicação se destina e a versão desse perfil. No caso da tabela é informado o *Enhanced Profile* com versão mínima igual a 1.1.1. Nesse descritor pode haver mais de uma par perfil/versão. Se o receptor der suporte a qualquer dos pares informados, estará apto a executar a aplicação;
- O descritor do protocolo de transporte, que pode ser um descritor comum entre as aplicações e apresenta o identificador do protocolo e informações específicas do mesmo. No caso da utilização do carrossel de objetos o identificador é 0x0001 e as outras informações disponibilizadas são: um identificador da rede original DVB-SI, um identificador do fluxo de transporte e o identificador do serviço DVB-SI. Esses são basicamente os mesmos elementos formadores do localizador DVB;
- Os descritores de informações para o usuário que incluem o descritor de nome da aplicação (informando nome e idioma) e o descritor de ícones da aplicação (informando uma figura pela qual o usuário terá acesso à aplicação). Esses descritores são específicos de

cada aplicação não podendo ser declarados juntamente com os descritores comuns;

- Os descritores específicos de aplicações DVB-J que informam: os parâmetros para a aplicação; o diretório base que representa o caminho (*path*) para execução da aplicação; caminhos adicionais para classes a serem inicializadas junto com a aplicação (*classpath extensions*); a classe inicial da aplicação.

Perfil da aplicação	0x1( <i>Enhanced Profile</i> )
Version.Major	0x1
Version.Minor	0x1
Version. Micro	0x1
Protocolo de transporte(){	0x0001
Identificador rede original	0x12
Identificador fl. De transporte	0x1
Identificador serviço	0x12
}	
ISO_639_language_code	pt-br
Nome da aplicação	Ginga-NCL
Diretório base	/app/
Extensão ao classpath	/players
Classe inicial	GingaXlet

Tabela 5 – Parâmetros adicionais da AIT para sinalização do Formatador *Xlet*.

No caso da utilização das facilidades proporcionadas pelos *plug-ins* do DVB-MHP ou *Application Storage* (Armazenamento de Aplicações), mais alguns descritores podem ser utilizados:

- Um descritor para delegação de aplicação. Esse descritor pode ser comum ou específico de uma aplicação e nele é especificado o identificador da aplicação responsável por interpretar aquele conteúdo;
- Um descritor de *plug-ins* que informa qual tipo de aplicação o plug-in é capaz de interpretar. Esse descritor deve ser informado para cada plug-in presente na lista de aplicações;

- Um descritor de aplicações armazenadas. Nesse descritor é informado, principalmente, se o ciclo de vida da aplicação é controlado pela difusora ou requer a interação do usuário (como jogos, por exemplo), e a versão da aplicação.

## Apêndice C

### A Linguagem NCL

Para a construção de aplicações interativas para sistemas de TV Digital faz-se necessário o sincronismo entre as mídias envolvidas neste sistema. Além do sincronismo do vídeo e áudio principal, o sistema de TV Digital deve ser capaz de lidar com a sincronização espacial e temporal de objetos de diferentes tipos de mídia.

O XHTML, usado pela maioria dos *middlewares* descritos neste capítulo, é uma linguagem baseada em um modelo hipermídia bastante simples. Apesar de sua simplicidade ser um dos fatores que contribuiu para a sua grande difusão, esta introduz limitações como, por exemplo, o fato de permitir apenas o sincronismo espacial. Para se obter sincronismos espaço-temporais é necessário recorrer a recursos fora do paradigma declarativo, como o uso do ECMAScript.

O modelo hipermídia no qual foi baseado o XHTML define apenas entidades do tipo nó e elo, sendo que existe apenas um tipo de elo (usado para representar relações de referência), diminuindo, assim, o poder de expressão da linguagem. Esse modelo limitado determina, ainda, que a definição dos elos esteja contida no nó, tornando impossível reusar um documento sem herdar os elos definidos dentro dele.

O W3C, órgão oficial que regula os padrões da *World Wide Web*, define o SMIL (W3C, 2005) como linguagem oficial para o sincronismo de mídias. Nessa linguagem, solucionando o problema do HTML, os elos não se encontram definidos dentro dos nós. É especificada nela, ainda, uma entidade denominada Composição. A Composição contém um conjunto de elos e nós e, no caso do SMIL, estabelece uma semântica de sincronização. As Composições podem ser de três tipos: paralela, seqüencial ou exclusiva. A atribuição dessa semântica às Composições, no entanto, pode dificultar o processo de autoria quando na criação de relacionamentos complexos. Outro fator limitante dessa linguagem é fato dela não oferecer suporte à edição/geração ao vivo de documentos.

A NCL (*Nested Context Language*) é uma linguagem XML para a autoria de apresentações hipermídia, incluindo programas de TV não lineares, baseada no modelo hipermídia NCM (*Nested Context Model*) (Soares et al, 2003). Seu foco é o sincronismo de mídias.

Por ser baseada no modelo NCM, a linguagem permite:

- o uso de composições para a estruturação lógica do documento;
- o uso de conectores hipermídia para a autoria de elos e fornecendo a semântica de sincronismo destes;
- a possibilidade de escolha entre um conjunto de nós alternativos; e
- a especificação da apresentação por meio de descritores.

A NCL define como os objetos de mídia estão estruturados e como se relacionam, no tempo e no espaço. Contudo, ao contrário do SMIL, os tipos de conteúdo desses objetos de mídia não são restritos. Ou seja, num documento NCL os objetos de mídia além de consistir de imagens estáticas, áudio, vídeo e texto, amplamente implementado pelas demais linguagens, podem representar objetos de execução (programas na linguagem Lua, JAVA, etc.) ou até documentos de outras linguagens (XHTML, SMIL, etc.). Por esse motivo a linguagem NCL é conhecida como uma linguagem de **cola**.

Outra vantagem da linguagem NCL é que esta permite a autoria de documentos em tempo de apresentação, ou seja, ela permite a alteração documento durante a sua execução. Esta funcionalidade é de grande valor em sistemas de TV digital onde a transmissão de programas ao vivo, como eventos esportivos, requerer uma alteração em tempo real da apresentação. Nesse ambiente, a autoria em tempo de apresentação é realizada através do envio de comandos de edição via NCL *stream events* (eventos DSM-CC).

A versão atual do NCL encontra-se especificada usando XML Schema (W3C, 2004b). Nela é prevista a existência de dois perfis: um Perfil Básico e um Perfil Avançado para TV digital. Esses perfis podem ser construídos graças à característica modular da linguagem através da seleção de módulos específicos. Estes módulos que compõem a linguagem são um conjunto de elementos, atributos e valores de atributos XML que representam uma unidade funcional.

As áreas funcionais da linguagem e seus respectivos módulos são os que seguem:

- Estrutural – módulo *Structure*;

- Leiaute – módulo *Layout*;
- Componentes – módulo *Media* e módulo *Context*;
- Interfaces – módulo *MediaContentAnchor*, módulo *CompositeNodeInterface*, módulo *AttributeAnchor*, e módulo *SwitchInterface*;
- Especificação da Apresentação – módulo *Descriptor*;
- Elos – módulo *Linking*;
- Conectores – módulo *XConnector* e módulo *CompositeConnector*;
- Controle da Apresentação – módulo *TestRule*, módulo *TestRuleUse*, módulo *ContentControl* e módulo *DescriptorControl*;
- Temporização – módulo *Timing*;
- Reuso – módulo *Import*, módulo *EntityReuse* e módulo *ExtendedEntityReuse*;

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="coisaDePele"
xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile
http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd">
<head>
  <regionBase>
    <region height="480" id="window1" title="None"
width="640" >
      <region height="200" id="imgRegion1" left="0"
top="0" width="80%"/>
      <region height="280" id="logoRegion1" left="0"
top="200" width="100%"/>
      <region height="50" id="fraseRegion1" left="0"
top="220" width="100%"/>
    </region>
  </regionBase>
  <descriptorBase>
    <descriptor id="img_d1" region="imgRegion1"/>
    <descriptor id="logo_d1" region="logoRegion1"/>
    <descriptor id="frase_d1" region="fraseRegion1">
      <descriptorParam name="color" value="#FF0000" />
    </descriptor>
  </descriptorBase>
  <connectorBase>
    <importBase alias="connBase"
documentURI="coisadePele30.conn"/>
  </connectorBase>
</head>
<body>
  <port id="inicio" component="logotele1" />
  <media type="image" descriptor="logo_d1"
id="logotele1" src="img/logo.jpg" >
    <area begin="5.1s" end="10.5s" id="part1"/>
  </media>
</body>
</ncl>
```

```
</media>
  <media type="image" descriptor="img_d1" id="img2"
src="img/MIDIA.jpg"/>
  <media type="text" descriptor="frase_d1" id="frase"
src="frase.txt" />
  <link id="link1" xconnector="connBase#onBeginStart">
    <bind component="logotele1" interface="part1"
role="onBegin"/>
    <bind component="img2" role="start"/>
  </link>
  <link id="link4" xconnector="connBase#onEndStop">
    <bind component="logotele1" interface="part1"
role="onEnd" />

    <bind component="img2" role="stop"/>
  </link>
  <link id="link3" xconnector="connBase#onEndStart">
    <bind component="logotele1" interface="part1"
role="onEnd" />
    <bind component="frase" role="start"/>
  </link>
</body>
</ncl>
```

Figura 39 – Exemplo de documento NCL 2.0

A Figura 38 ilustra o código de um documento NCL para um melhor entendimento da linguagem. Para maiores informações sobre essa linguagem o leitor deve consultar (Soares & Rodrigues, 2006).