



**Rafael Diniz**

**O Perfil NCL-DR e o Middleware Ginga para  
Receptores do Sistema Brasileiro de Rádio  
Digital**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para  
obtenção do grau de Mestre pelo Programa de Pós-  
Graduação em Informática do Departamento de  
Informática da PUC-Rio.

Orientador: Prof. Luiz Fernando Gomes Soares

Rio de Janeiro  
Julho de 2015



**Rafael Diniz**

**O Perfil NCL-DR e o Middleware Ginga para  
Receptores do Sistema Brasileiro de Rádio  
Digital**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Luiz Fernando Gomes Soares**

Orientador

Departamento de Informática – PUC-Rio

**Prof. Sérgio Colcher**

Departamento de Informática – PUC-Rio

**Prof. Marcio Ferreira Moreno**

Departamento de Informática – PUC-Rio

**Prof. José Eugenio Leal**

Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 10 de julho de 2015.

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Rafael Diniz**

Recebeu seu título de Bacharel em Ciência da Computação pela Universidade Estadual de Campinas em 2009. Atualmente integra o grupo de pesquisadores do Laboratório Telemídia, desenvolvendo pesquisas na área de Sistemas Hiperídia e Rádio Digital.

### Ficha Catalográfica

Diniz, Rafael

O Perfil NCL-DR e o Middleware Ginga para Receptores do Sistema Brasileiro de Rádio Digital / Rafael Diniz; orientador: Luiz Fernando Gomes Soares. - 2015.

135 f. : il. (color.) ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2015.

Inclui bibliografia

1. Informática – Teses. 2. Rádio Digital Interativo. 3. NCL perfil Digital Radio. 4. Ginga. 5. Sistema Brasileiro de Rádio Digital. 6. Digital Radio Mondiale. 7. HD Radio. I. Soares, Luiz Fernando Gomes. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para os meus pais, Constantino e  
Fabiola, por todo o amor, carinho e  
suporte na minha formação.

## Agradecimentos

Tenho muita alegria de ter podido concretizar o sonho de realizar um trabalho que considero relevante na área de comunicações eletrônicas.

Agradeço inicialmente aos guerreiros e guerreiras que mantêm há mais de 30 anos a Rádio Muda FM, rádio livre que é minha inspiração e dá sentido para os trabalhos que deenvolvo na área.

Muitas pessoas colaboraram para que este trabalho existisse, tanto na questão técnica, política e social, como nas questões pessoais durante esses dois anos e meio aqui no Rio de Janeiro. Agradeço de coração a todas elas, e em especial:

Aos amigos de muitos anos, muitos dos quais estão na mesma luta que eu por um país mais livre: Rhatto, Chico, Novaes, Pajeh, Letícia, Ratitu, Zollner, Davi Corturnada, Helena, Nina, Jupagul, Juh, Lua, Goa, Mendigo, Samuca, Didi, Fran, Lu, Djahjah, Nils, Foz, Elisa, Drebs, Diogo, Belisário, ...

Ao guru Luiz Fernando Gomes Soares, meu orientador e personalidade da academia brasileira que mais admiro, por toda a clareza para ensinar e compartilhar seu conhecimento, pela colaboração em todas as iniciativas que tive no campo político e acadêmico, e também por toda a esperança que sua Ginga renovou nos pesquisadores do Sul Global, mostrando como se faz e implementa uma tecnologia aqui em baixo.

Aos amigos do Telemídia, Alan, Guilherme, Roberto, Rodrigo, Nagato, Eduardo, Amparito e Ricardo, com quem aprendi muito, e com os quais compartilhei vários bons momentos, entre festas, lanchinhos e discussões técnicas, políticas, filosóficas e tudo mais.

Ao Marcio e Álvaro, que se incluem no grupo anterior também, mas que

toparam realizam um evento sobre Rádio Digital em 2013, a Conferência Internacional Espectro, Sociedade e Comunicação II, que foi muito importante para a discussão sobre o rádio digital no país.

A todos que me ajudaram com o DRM, seja com software, equipamento ou informações. Principalmente ao pessoal da BBC, Ruxandra Obreja, Lindsay Cornell, Julian Cable, do Fraunhofer IIS, Alex Zink, e o autor do software Spark, Michael Feilen.

Ao Ministério das Comunicações do Brasil e à EBC, por estarem realizando um trabalho consistente na análise dos distintos padrões de rádio digital. Principalmente ao Flávio Lima, Ismar, Sartorello e Bráulio.

À Bruna, que conviveu comigo durante boa parte do mestrado, minha amiga e companheira no ativismo, com a qual vivi muitos momentos felizes.

Aos meus pais, Fabíola e Constantino, por toda a excelente criação que me proporcionaram, pelo amor, carinho e pelo suporte que até hoje me dão para seguir com meus sonhos.

## Resumo

Diniz, Rafael; Soares, Luiz Fernando Gomes. **O Perfil NCL-DR e o Middleware Ginga para Receptores do Sistema Brasileiro de Rádio Digital**. Rio de Janeiro, 2015. 135p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Em 2010, o Ministro das Comunicações instituiu o Sistema Brasileiro de Rádio Digital (SBRD). No entanto, um modelo de referência para o sistema ainda não foi definido. Este trabalho apresenta uma análise da radiodifusão sonora terrestre no Brasil e a partir desta são inferidos alguns requisitos para um rádio digital interativo. Esses requisitos são então utilizados para nortear a pesquisa. A relevância e importância do uso da NCL e do Ginga no rádio digital brasileiro, assim como acontece no Sistema Brasileiro de TV Digital (SBTVD), são discutidas, e a definição da forma como deve ser feito o transporte de aplicações NCL nos dois sistemas de rádio digital sendo considerados para adoção pelo país é realizada. Um novo perfil da linguagem NCL para o rádio digital é definido. Esse novo perfil recebeu o nome de perfil *Digital Radio*, ou simplesmente DR. O Ginga também é definido para uso em receptores de rádio digital, sendo que novos exibidores de mídia e recursos adaptados ao contexto do rádio digital são introduzidos. A implementação do Ginga da PUC-Rio é apresentada com as modificações necessárias para o rádio digital. Adicionalmente, um ambiente para validar a interatividade no rádio digital com o Ginga foi montado e testes exercitando toda a cadeia de transmissão e recepção de rádio digital com aplicações NCL embutidas foram realizados. As definições e conclusões que resultaram da pesquisa visam contribuir para a definição de um Sistema Brasileiro de Rádio Digital que seja poderoso em termos comunicacionais e agregue os recursos mais relevantes para o meio na era digital.

## Palavras-chave

NCL; Rádio Digital Interativo; NCL perfil *Digital Radio*; Ginga; Sistema Brasileiro de Rádio Digital; Digital Radio Mondiale; HD Radio.

## Abstract

Diniz, Rafael. Soares, Luiz Fernando Gomes (Advisor). **The NCL-DR Profile and the Ginga Middleware for the Brazilian Digital Radio System.** Rio de Janeiro, RJ, Brasil, 2015. 135p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In 2010, the Minister of Communications instituted the Brazilian Digital Radio System (SBRD), however a reference model for the system has not yet been set. This text presents an analysis of the terrestrial radio broadcasting in Brazil and presents some requirements for an interactive digital radio. These requirements are then used to guide the research. The relevance and importance of the use of NCL and Ginga in the Brazilian digital radio as in Brazilian Digital TV System (SBTVD), are discussed, and the definition of how the transport of NCL applications should be done in the two digital radio systems being considered for adoption by the country is performed. A new profile of NCL for use in digital radio is defined. This new profile was named Digital Radio Profile, or just DR. Ginga is also defined for use in digital radio receivers, and new media players and features adapted to the digital radio context are introduced. PUC-Rio's implementation of the Ginga middleware is presented with the necessary modifications for digital radio use. In addition, an environment to validate the interactivity in digital radio with the Ginga was assembled and tests exercising the entire digital radio chain of transmission and reception with embedded NCL applications were performed. The definitions and conclusions that resulted from research activities are expected to contribute to the definition of a Brazilian Digital Radio System that is powerful in communicational terms and aggregates the most relevant technologies for the medium in the digital age.

## Keywords

NCL; Interactive Digital Radio; NCL Digital Radio profile; Ginga; Brazilian Digital Radio System; Digital Radio Mondiale; HD Radio.



## Sumário

1 Introdução	17
1.1. Motivação	17
1.2. Objetivos	19
1.3. Organização da dissertação	20
2 Trabalhos relacionados	22
2.1. Transporte	22
2.2. Aplicações	24
2.3. Tecnologias utilizadas	28
3 Requisitos para a interatividade na radiodifusão brasileira	30
3.1. Quem são emissoras e os ouvintes de rádio no Brasil?	30
3.2. Aplicativos vislumbrados para o rádio digital	34
3.3. Requisitos levantados	35
4 Transporte de aplicações via rádio digital	37
4.1. Digital Radio Mondiale	37
4.2. HD Radio	50
5 O Perfil NCL Digital Radio	57
5.1. Elementos de aparência e leiaute	60
5.2. Elementos de efeitos de transição	62
5.3. Elementos para importação de entidades externas	63
5.4. Elementos de Estrutura e Conteúdo	64
5.5. Elementos de ligação	67
5.6. Elementos conectores	69
5.7. Switches e regras	74
5.8. Elementos de interface	76
5.9. Variáveis globais	80

5.10. Elementos de metadados	89
5.11. Resumo das características do perfil DR da NCL	90
6 Ambiente de apresentação Ginga e novas mídias	92
6.1. Monomídias	92
6.1.1. Gráficos Vetoriais	93
6.1.2. Sintetização de voz	94
6.1.3. Áudio e imagem	95
6.2. Objetos imperativos NCLua	96
6.3. Acesso a informações de geolocalização do receptor	97
6.4. Base temporal	100
6.5. Comandos de edição	103
6.6. Suporte a múltiplos dispositivos	109
6.7. Resumo das características do middleware Ginga para rádio	111
7 Implementação de referência do Ginga	114
8 Demonstração de transmissão, recepção e execução de aplicação NCL	121
9 Conclusão	128
10 Referências Bibliográficas	131

## Lista de figuras

Figura 1. A cor verde identifica os países que utilizam o sistema ISDB-T variante internacional, cujo <i>middleware</i> é o Ginga.	18
Figura 2. Foto de aplicação Journaline sendo executada em receptor de rádio digital.	25
Figura 3. Receptor HD Radio apresentando uma imagem.	27
Figura 4. Tecnologias que fazem parte de um sistema In-Vehicle Infotainment (IVI). Fusca do ex-presidente do Uruguai Pepe Mujica meramente ilustrativo.	33
Figura 5. Exemplo de configuração do DRM mostrando um exemplo de mapeamento entre os serviços e os fluxos de transmissão.	38
Figura 6. Estrutura do MOT <i>Directory</i> .	42
Figura 7. Sintaxe de um parâmetro do <i>DirectoryExtension</i> ou <i>HeaderExtension</i> .	43
Figura 8. Parâmetros permitidos no <i>DirectoryExtension</i> .	44
Figura 9. Possíveis parâmetros do <i>Header extension</i> para objetos do carrossel.	47
Figura 10. Serviços do HD Radio.	51
Figura 11. Diagrama de blocos do sistema HD Radio.	52
Figura 12. Etapas de segmentação dos objetos MOT em <i>MSC data groups</i> .	53
Figura 13. Segmentação do <i>MSC data group</i> do MOT em <i>Data Transport Packets</i> (identificados como <i>ADS Data Packet</i> na figura) do protocolo AAT do HD Radio.	54
Figura 14. Arquitetura típica de uma máquina de apresentação NCL.	58
Figura 15. Aplicativo Ginga para rádio digital.	94
Figura 16. Exemplo de receptor de rádio com antena telescópica em região rural.	111
Figura 17. Componentes do Ginga.	114
Figura 18. Arquitetura do Ginga com destaque aos módulos adaptados para o rádio digital em verde, e módulo novo em vermelho.	116
Figura 19. Elementos que compõem a transmissão de rádio digital.	121

Figura 20. Receptor Dream recebendo uma aplicação NCL.	124
Figura 21. Esquema de teste que envolve os elementos relevantes (em linha contínua) da cadeia de transmissão/recepção DRM para aplicações NCL.	124
Figura 22. Foto da USRP.	125
Figura 23. Foto dos equipamentos de transmissão DRM da NTi.	126
Figura 24. Ambiente de teste para transmissão DRM. Da esquerda para direita: computador executando o software Spark, USB DAC 48, DiRaGen30 e antena.	126
Figura 25. Foto do receptor FUNcube Dongle Pro+.	127

## Lista de tabelas

Tabela 1. Divisão por tipo das emissoras de rádio do Brasil.	32
Tabela 2. Taxas de bits típicas do broadcast digital.	35
Tabela 3. Valor do <i>Application identifier</i> no FAC para sinalizar a presença de uma aplicação Ginga em um serviço de dados.	39
Tabela 4. Campos e valores da entidade <i>Application information</i> do canal SDC para transmissão de aplicativos NCL.	41
Tabela 5. Parâmetros do <i>DirectoryExtension</i> .	44
Tabela 6. Sintaxe do campo de dados do parâmetro <i>DirectoryIndex</i> .	45
Tabela 7. Possíveis sintaxes para o <i>index_name</i> .	46
Tabela 8. Valor de <i>ContentType</i> e <i>ContentSubType</i> possível para cada entrada do carrossel.	47
Tabela 9. Parâmetros importantes do <i>Header extension</i> para os objetos do carrossel.	49
Tabela 10. URI para referenciar serviços no multiplex DRM.	49
Tabela 11. Atribuição do número das portas do HD Radio.	53
Tabela 12. Campos do cabeçalho dos pacotes da camada de adaptação entre os protocolos MOT e AAT.	54
Tabela 13. URI para referência a elementos do multiplex do HD Radio.	56
Tabela 14. Presença dos elementos <i>&lt;region&gt;</i> , <i>&lt;regionBase&gt;</i> , <i>&lt;descriptor&gt;</i> , <i>&lt;descriptorBase&gt;</i> e <i>&lt;descriptorParam&gt;</i> nos perfis da NCL 3.1 e perfil DR.	62
Tabela 15. Presença de efeitos de transição nos perfis da NCL 3.1 e perfil DR.	63
Tabela 16. Presença dos elementos de importação nos perfis da NCL 3.1 e perfil DR.	64
Tabela 17. Existência dos elementos de estrutura e conteúdo nos diferentes perfis.	66
Tabela 18. Definição dos elementos de estrutura e conteúdo no perfil DR.	67
Tabela 19. Presença dos elementos de ligação nos diferentes perfis.	68
Tabela 20. Definição dos elementos de ligação no perfil DR.	68

Tabela 21. Presença dos conectores nos diferentes perfis.	73
Tabela 22. Elementos conectores, seus atributos e elementos filhos.	73
Tabela 23. Presença dos elementos de <i>Switch</i> e regras nos diferentes perfis.	76
Tabela 24. Elementos Switch e regras, seus atributos e filhos.	76
Tabela 25. Presença dos elementos de Interface nos diferentes perfis.	80
Tabela 26. Elementos de interface, seus atributos e conteúdo.	80
Tabela 27. Variáveis globais, seus grupos, semântica e possíveis valores.	89
Tabela 28. Presença dos elementos de metadados nos diferentes perfis.	90
Tabela 29. Elementos para especificação de metadados, seus atributos e conteúdo.	90
Tabela 30. Variáveis do grupo <i>location</i> que expõe informações de geolocalização do receptor.	99
Tabela 31. Sintaxe de envio de valores de tempo em dada base temporal.	101
Tabela 32. Entidade <i>TimeBaseEndpoint</i> .	102
Tabela 33. Estrutura de um <i>Editing Command event descriptor</i> .	105
Tabela 34. Comandos de edição suportados no perfil DR.	109
Tabela 35. Tipos de mídia a serem suportadas no perfil básico de receptores para rádio digital.	113

## Listagens

Listagem 1. Requisitos para um rádio digital interativo.	36
Listagem 2. Todos os elementos da linguagem NCL 3.1.	59
Listagem 3. Elementos de aparência e leiaute.	60
Listagem 4. Documento NCL que contém elementos <i>&lt;region&gt;</i> , <i>&lt;regionBase&gt;</i> , <i>&lt;descriptor&gt;</i> e <i>&lt;descriptorBase&gt;</i> .	61
Listagem 5. Documento NCL que faz o uso do elemento <i>&lt;property&gt;</i> .	61
Listagem 6. Elementos de efeitos de transição.	62
Listagem 7. Elementos de importação.	63
Listagem 8. Elementos de Estrutura e Conteúdo.	64
Listagem 9. Elementos de ligação.	67
Listagem 10. Exemplo de uso dos elementos de ligação.	68
Listagem 11. Elementos conectores.	69
Listagem 12. Código contendo um exemplo da definição de um conector.	72
Listagem 13. Elementos que permitem Switch e regras.	74
Listagem 14. Elementos de interface.	76
Listagem 15. Exemplo de uso do elemento <i>&lt;port&gt;</i> .	77
Listagem 16. Trecho de código NCL que exemplifica o uso dos elementos <i>&lt;area&gt;</i> e <i>&lt;property&gt;</i> .	80
Listagem 17. Elementos de metadados.	89
Listagem 18. Exemplo de uso de metadados.	90
Listagem 19. Trecho do arquivo de configuração do <i>drmcS</i> que contém as informações válidas para multiplexação de uma aplicação NCL.	123

*... é preciso transformar o rádio, convertê-lo de aparelho de distribuição em aparelho de comunicação. O rádio seria o mais fabuloso meio de comunicação imaginável na vida pública, um fantástico sistema de canalização.*

*Isto é, seria se não somente fosse capaz de emitir, como também de receber; portanto, se conseguisse não apenas se fazer escutar pelo ouvinte, mas também pôr-se em comunicação com ele. A radiodifusão deveria, conseqüentemente, afastar-se dos que a abastecem e constituir os radiouvintes como abastecedores. Portanto, todos os esforços da radiodifusão em realmente conferir, aos assuntos públicos, o caráter de coisa pública são totalmente positivos.*

Bertolt Brecht, *Teoria do Rádio*



# 1

## Introdução

### 1.1.Motivação

A interatividade na TV e no rádio, que se tornou definitivamente possível com a mudança do sistema analógico para digital, é, segundo Marshall (MARSHALL, 2004), a principal característica dessa mudança de tecnologia.

Para dar suporte à interatividade, NCL (*Nested Context Language*) é uma linguagem declarativa, criada no Brasil, para especificação de aplicações multimídia. A NCL foi adotada como padrão para aplicações multimídia do sistema ISDB-T que, em sua variante internacional, é uma evolução brasileira do ISDB-T utilizado no Japão (ver Figura 1), tendo sido adotado pela maioria dos países da América Latina e alguns países da África. NCL é também a linguagem para especificação multimídia de serviços de IPTV que seguem a Recomendação H.761 (ITU, 2014a) da União Internacional de Telecomunicações.

Um documento NCL é uma aplicação XML que descreve relacionamentos espaço-temporais entre objetos de mídia (texto, imagem, áudio, scripts NCLua, etc.) e interações do usuário. Os elementos XML, os atributos desses elementos e seus conteúdos que compõem a especificação NCL são agrupados por funcionalidade em diferentes módulos que, em conjunto, formam um perfil da linguagem.

A última versão da linguagem NCL é a 3.1. Ela é definida para serviços de TV Digital Terrestre e IPTV, e possui dois perfis (LIMA, 2013), o EDTV (*Enhanced Digital Television*), que é o perfil avançado, e o RawDTV (*Raw Digital Television*), que é um perfil composto por um subconjunto dos elementos presentes no EDTV. O perfil RawDTV foi concebido com o objetivo de simplificar a máquina de execução NCL, por meio da remoção de “acúcares sintáticos” do perfil principal. Todo documento RawDTV é também um documento EDTV válido, e todo documento EDTV pode ser convertido para um

documento RawDTV sem perda semântica. O ambiente de apresentação ou middleware especificado para reproduzir um documento NCL é denominado Ginga.

Em 2010, o Sistema Brasileiro de Rádio Digital (SBRD) foi instituído (COSTA, 2010). No entanto, um modelo de referência do sistema não foi estabelecido até o momento.

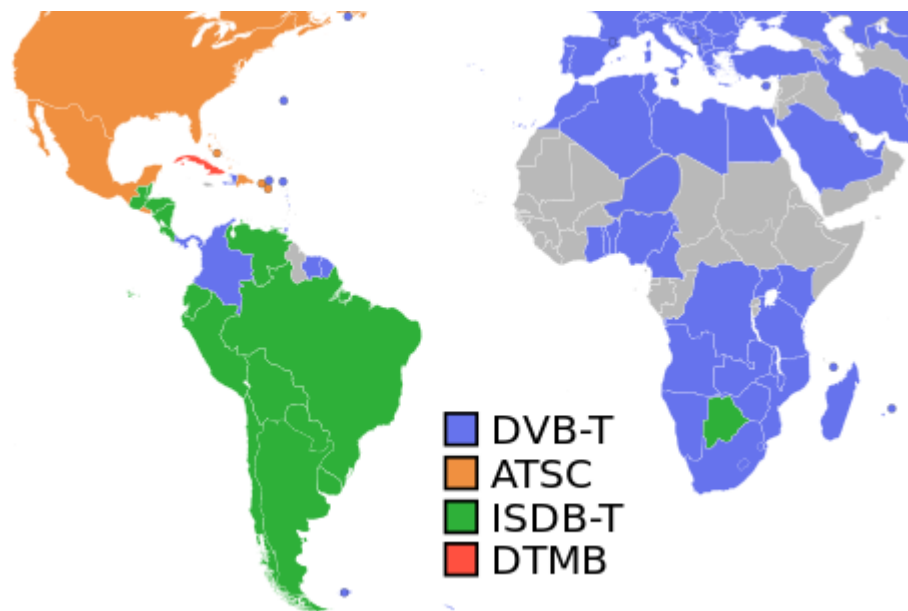


Figura 1. A cor verde identifica os países que utilizam o sistema ISDB-T variante internacional, cujo *middleware* é o Ginga.

O Rádio Digital, apesar de possuir muitas semelhanças com a TV Digital terrestre, como o fato de ser um sistema capaz de enviar arquivos e outras informações digitais por difusão (*broadcast*), apresenta algumas diferenças que justificam a criação de um novo perfil da linguagem NCL e uma nova especificação do *middleware* Ginga. São algumas dessas diferenças:

- Tipos de receptores diferentes, com suporte a bandas de frequência com distintas canalizações e características de propagação (OM, OT, OC e/ou VHF);
- Taxas de transmissão máximas distintas de acordo com a banda, e bem inferiores à TV Digital;
- Sistema de multiplexação bem simplificado, diferente da TV Digital, otimizado para ter menor desperdício de *bits*, e também menos meta-informações;
- Conteúdo diferente da TV, uma vez que o rádio é um meio focado no áudio, ao contrário do conteúdo audiovisual da TV. Como consequência, a

importância dos elementos visuais de um aplicativo NCL para o rádio é bastante diferente da TV;

- O número superior de emissoras de rádio comparado ao das emissoras de TV, o que permite que algumas emissoras de rádio tenham conteúdos focados em nichos de audiência, como emissoras focadas em informações de trânsito ou esportes;
- A grande audiência do rádio por ouvintes que estão em trânsito, seja caminhando ou dentro de automóvel, trem, barco ou ônibus;
- Grande penetração do rádio entre pessoas com restrições de visão.

Para os países que já usam o Ginga no sistema de TV, é muito importante a extensão dos recursos de interatividade também para o rádio, através da definição de um perfil da linguagem NCL e seu *middleware*.

Outro aspecto importante da definição do Ginga para o rádio digital são as vantagens provenientes da economia de escala na fabricação de receptores, incluindo o desenvolvimento de receptores que contemplem tanto o perfil para TV quanto para o rádio, sendo possível o uso de um mesmo *middleware* para aplicações provenientes do rádio ou TV.

O entendimento do ecossistema da radiodifusão é muito importante para a definição da NCL e do Ginga para o rádio digital. A especificação de um sistema para interatividade, que se configure como uma real ferramenta potencializadora da comunicação via rádio, é o cerne da motivação deste trabalho.

## 1.2.Objetivos

O objetivo principal deste trabalho é a concepção de um perfil da linguagem NCL para o rádio digital, denominado *Digital Radio* (DR), assim como a definição do *middleware* Ginga para o rádio digital, que irá reproduzir o novo perfil e incorporar as funcionalidades necessárias para operar no receptor de rádio digital.

O perfil para rádio da NCL e a implementação do Ginga deverá permitir um ambiente consistente e interoperável para a interatividade no rádio e TV, permitindo o reuso de aplicações feitas para a TV no rádio e vice-versa.

A autoria das aplicações no perfil para o rádio digital não deve ser dificultada, permitindo que autores já acostumados com o perfil para TV Digital possam se adaptar facilmente às modificações, ou possam utilizar conversores para a tradução de um perfil para outro.

As definições a serem adotadas para a NCL e para o Ginga devem ser embasadas em requisitos para a interatividade na radiodifusão digital, que são abordados no Capítulo 3.

Soluções para as questões do rádio digital, tais como a baixa taxa de transmissão, sistema de multiplexação simplificado, mobilidade e acessibilidade, devem ser definidas. Recursos avançados do Ginga, tais como sincronia fina com o fluxo principal, edição ao vivo da aplicação em tempo de execução e suporte a apresentação distribuída em múltiplos dispositivos devem ser mantidos.

Um objetivo específico se dá no âmbito da máquina de apresentação Ginga. É importante que exista uma implementação de referência funcional do Ginga para rádio digital.

As modificações na implementação de referência do Ginga devem permitir que o Ginga seja capaz de executar aplicações provenientes tanto do rádio como da TV, carregando na memória somente os módulos necessários, sempre que possível. Dessa forma, o consumo de CPU e memória do receptor poderá ser otimizado de acordo com o tipo do serviço.

Outro objetivo é o desenvolvimento de um sistema funcional que exercite toda a cadeia de transmissão e recepção de rádio digital com aplicações NCL. Para isso, todas as definições de sinalização e multiplexação de aplicações NCL pelo rádio digital são realizadas, assim como a implementação dessas definições.

Esse último objetivo é importante para permitir que as provas de campo que vem sendo realizadas no Brasil com rádio digital já possam lançar mão da interatividade através de aplicações NCL.

### **1.3.Organização da dissertação**

O restante do texto está organizado da seguinte forma: a Capítulo 2 discute alguns trabalhos relacionados. O Capítulo 3 apresenta um panorama da radiodifusão brasileira e extrai alguns requisitos para a interatividade no rádio digital. O Capítulo 4 discute e define a maneira de multiplexar aplicativos NCL na

transmissão de rádio digital. O Capítulo 5 contém a definição do perfil *Digital Radio* da NCL e discussões a respeito das opções feitas. O Capítulo 6 apresenta aspectos relacionados ao ambiente de apresentação e novos tipos de mídia. O Capítulo 7 discute as modificações na implementação de referência do Ginga, que proveem o suporte à NCL DR e aos outros recursos definidos para o rádio. O Capítulo 8 demonstra todas as etapas de transmissão, recepção e execução de uma aplicação NCL DR transportada pelo sistema Digital Radio Mondiale. Finalmente, o Capítulo 9 contém as conclusões e trabalhos futuros.

## 2

## Trabalhos relacionados

### 2.1. Transporte

Atualmente, quatro padrões de rádio digital são reconhecidos pela União Internacional de Telecomunicações (ITU) através de duas Recomendações (ITU, 2011) (ITU, 2014b): DAB, ISDB-Tsb, IBOC e Digital Radio Mondiale (DRM). Todos esses padrões especificam algum tipo funcionalidade para o envio de arquivos e comandos na forma de carrossel. O Brasil está considerando para adoção somente os sistemas de rádio digital de banda estreita, o IBOC e o DRM (DINIZ, 2013).

Os outros sistemas não considerados para adoção pelo país são o DAB, utilizado principalmente na Europa, e o ISDB-Tsb, utilizado no Japão. As aplicações definidas no DAB são semelhantes às definidas para o DRM e são discutidas a seguir. Para o ISDB-Tsb, o mesmo *middleware* e linguagem do ISDB-T, o BML (*Broadcast Markup Language*), são utilizados. O BML é baseado no padrão XHTML adicionado de algum suporte a folhas de estilo, possuindo recursos relativamente simples, tendo sido substituído pelo Ginga na variante do ISDB-T desenvolvida no Brasil.

O sistema IBOC (*In-Band On-Channel*) é o padrão de rádio digital adotado pelos Estados Unidos. Foi desenvolvido por empresas norte-americanas, tais como AT&T, Lucent Digital Radio, CBS, Gannett e Westinghouse durante a década de 90. Em 2000, os atores envolvidos na criação do padrão criaram a Ibiquity Digital Corporation, que ficou responsável pelo licenciamento do sistema e cobrança de royalties. Em 2002 foi adotado como único padrão estadunidense para radiodifusão sonora terrestre digital pela FCC (*Federal Communications Commission*).

O IBOC (NRSC, 2011a) recebeu da Ibiquity o nome comercial de HD Radio, nome pelo qual o sistema é comumente conhecido. As normas do sistema não especificam o codificador de áudio nem os protocolos que permitem a multiplexação de conteúdos multimídia. É o único sistema de rádio digital

reconhecido pela ITU considerado proprietário, devido aos segredos industriais que fazem parte do mesmo. O sistema opera nas faixas de Ondas Médias e VHF banda II (a faixa do FM). Além dos Estados Unidos, o HD Rádio também foi adotado no México, no entanto, somente para a faixa do VHF. O HD Radio não opera nas faixas de Ondas Tropicais e Ondas Curtas.

Devido ao fato do HD Radio utilizar protocolos fechados para multiplexação de aplicações interativas, é necessário que uma nova aplicação para o HD Radio acompanhe a definição do protocolo de multiplexação da mesma. No trabalho de Gorsak e Hendriks (GORSACK, 2010), é definido um protocolo de multiplexação simples para o HD Radio, capaz de transportar uma aplicação cujo conteúdo é semelhante ao de uma revista eletrônica.

O protocolo definido por Gorsak e Hendriks não contempla os requisitos necessários para a transmissão e controle de uma aplicação NCL, tais como abstração de arquivos e diretórios, primitivas para sincronização e possibilidade de transporte de comandos que modificam a exibição de uma aplicação.

Esta dissertação apresenta, portanto, uma abordagem mais completa para contemplar os recursos essenciais que o Ginga necessita, tais como sincronização fina através do uso de uma base temporal, suporte a abstração de arquivos de diretórios e o envio de comandos de controle e edição da aplicação.

O desenvolvimento do sistema DRM (ETSI, 2014) teve início em 1998, com a fundação do Consórcio DRM. O Consórcio DRM é uma organização sem fins lucrativos, com sede na Suíça, criado com o objetivo inicial de conceber um sistema de rádio digital para as faixas nas quais a modulação AM (Modulação em Amplitude) é utilizada. Em 2005, o sistema foi estendido para operar também em VHF, banda onde estão as emissoras FM (Modulação em Frequência).

O consórcio é formado por mais de 100 membros, dentre eles empresas de radiodifusão tais como BBC, Deutsche Welle, All India Radio, institutos de pesquisa e universidades, tais como o Fraunhofer Institute e a Universidade de Hannover, e empresas ligadas à indústria de transmissores e receptores. Foi adotado pela França e Alemanha para as faixas de OM e OC, e adotado como padrão nacional na Índia e Rússia. Vários radiodifusores que operam em OC já utilizam o sistema para transmissões internacionais, tais como Rádio Vaticano, Voice of Nigeria, Radio New Zealand International e BBC. O único país no

estágio de implementação avançada do sistema DRM é a Índia, onde grande parte do país já está coberto por sinais DRM.

O sistema DRM apresenta todos seus recursos definidos em normas internacionais públicas, sendo, portanto, um sistema considerado aberto. O DRM opera em todas as bandas definidas para radiodifusão sonora terrestre.

Para a multiplexação de aplicações, o DRM suporta o protocolo MOT (*Multimedia Object Transfer*) (ETSI, 2006b). O MOT permite a transmissão de arquivos individualmente ou de uma estrutura com arquivos e diretórios, existindo suporte para a transmissão comprimida do conteúdo. Também estão definidos campos para sinalização de horário de início e momentos de atualização de uma aplicação. No entanto, não existem mecanismos pré-definidos para o transporte de uma referência temporal independente do horário da transmissão, nem a possibilidade de comandos de edição da aplicação. Além desses fatores, o suporte ao controle do ciclo de vida da aplicação é limitado, sendo restrito ao controle do início da aplicação somente.

Pelo fato do protocolo MOT permitir a extensão de suas unidades sintáticas, o mesmo foi estendido para comportar recursos necessários ao Ginga tais como suporte ao envio de base temporal para sincronia fina de conteúdos, além de um mecanismo para envio de comandos de edição da aplicação. O protocolo MOT, com as devidas adaptações, foi o protocolo definido para o transporte de aplicações NCL, tanto para o sistema DRM, como para o sistema HD Radio.

Kurpiers (KURPIERS, 2003) mostrou a viabilidade da implementação, em software, da demodulação e decodificação de um sinal de rádio digital padrão Digital Radio Mondiale. A implementação, chamada Dream<sup>1</sup>, é software livre, e possui implementado o protocolo MOT de forma completa. O Dream é utilizado como base do receptor de rádio digital ao qual o Ginga desenvolvido nesta dissertação é embutido, no protótipo apresentado no Capítulo 8.

## 2.2.Aplicações

Para ambos os padrões passíveis de adoção pelo Brasil estão definidas algumas aplicações. Uma delas é o *Journaline*. O *Journaline* (ETSI, 2008) foi

---

<sup>1</sup> Receptor DRM Dream: <http://drm.sourceforge.net>



desenvolvido pelo Fraunhofer Institute, concebido como uma aplicação de revista eletrônica que necessita de uma baixa taxa de transmissão para transmissão.

O *Journaline* consiste de um serviço de dados semelhante a uma revista eletrônica. Sua unidade básica de transmissão é o objeto JML (*Journaline Markup Language*), que é um documento XML codificado de forma binária, que pode representar tanto uma página de menu, ou uma página com texto. Cada unidade JML é mapeada diretamente no fluxo do DRM, não sendo utilizado o protocolo MOT.

Uma vantagem do *Journaline* é o seu baixo consumo de bitrate, adequado ao contexto do rádio digital, graças ao uso de compressão e um protocolo de multiplexação específico para o serviço. No entanto, ele oferece um controle muito pequeno sobre como um conteúdo deve ser exibido, não oferece nenhum tipo de sincronização entre os conteúdos de mídia, nem a possibilidade de enviar mídias como imagens ou animações, se atendo basicamente ao texto. É um serviço bastante simples, porém, eficiente para o transporte de notícias. O *Journaline* pode ter sua reprodução embarcada em uma aplicação NCL através do elemento *<media>*, como será discutido no Capítulo 6.

A Figura 2 apresenta uma foto de uma aplicação *Journaline* sendo executada em um receptor de rádio digital.



Figura 2. Foto de aplicação *Journaline* sendo executada em receptor de rádio digital.

Outra aplicação definida para o sistema DRM é o do *Broadcast WebSite* (ETSI, 2006a), que é subdividido em dois perfis. O perfil básico consiste de

documentos HTML com conteúdo de texto e imagem. O perfil *TopNews* adiciona ao perfil básico o suporte a mídias de áudio MPEG-2 *Layer 2* (mp2) e *Layer 3* (mp3). Somente um conjunto restrito de elementos do padrão HTML 3.2 é permitido, não sendo suportado nenhum tipo de código imperativo como *ECMAScript*.

O *Broadcast WebSite* utiliza o protocolo MOT para o transporte dos arquivos da aplicação, assim como o Ginga proposto nesta dissertação. No entanto, não permite mídias visuais animadas, sincronia fina entre mídias e a edição de uma aplicação em tempo de apresentação. O Ginga, na versão para TV, já suporta o tipo de mídia HTML embarcada em uma aplicação NCL.

Outra aplicação presente no DRM é o *Slideshow* (ETSI, 2015a), que também utiliza o protocolo MOT. Essa aplicação consiste no envio de imagens e na apresentação das mesmas no receptor na forma de *Slideshow*. O *Slideshow* é uma aplicação simples e fácil de ser utilizada pelo radiodifusor. No entanto, consiste somente do envio de imagens, não permitindo o controle da exibição das mesmas ou a sincronia delas com outras mídias.

Muitas das normas para o DRM que definem aplicações como *Slideshow* (ETSI, 2015a) e guia de programação eletrônico (ETSI, 2015b) foram recentemente modificadas e tiveram seus títulos modificados. Onde se lia *Digital Radio* até 2014, em 2015 passou a ser *Hybrid Digital Radio*. Além da modificação dos títulos, definições para sinalização e aquisição de conteúdo via rede IP foram adicionadas às normas. Essa combinação de *broadcast* e IP (*Internet Protocol*), conhecida como *Hybrid Broadcast Broadband* ou *Integrated broadcast-broadband*, permite a convergência dos meios. O Ginga já na sua versão anterior suporta a transferência de dados via rede IP, sendo, portanto, um *middleware* que se enquadra no conceito *Integrated broadcast-broadband*.

Para o sistema HD Radio, nenhuma das aplicações existentes possuem documentação ou normas públicas, sendo, portanto, consideradas proprietárias.

No entanto, é possível identificar algumas aplicações do HD Radio por meio da análise de seus receptores. A aplicação *Artist Experience* consiste no envio de imagens, semelhante à aplicação *SlideShow* do DRM. A Figura 3 apresenta uma foto de um receptor automotivo HD Radio apresentando uma imagem.



Figura 3. Receptor HD Radio apresentando uma imagem.

Outra aplicação do HD Radio é conhecida como *iTunes Tagging Support*, recurso que também pode ser identificado na Figura 3 (palavra *Tag*). Quando o usuário seleciona a opção *Tag* no momento de reprodução de uma música, a mesma fica marcada para compra na *iTunes Store*, graças a um identificador que referencia a música na loja virtual. Posteriormente, o usuário deve completar a transação de compra através da aplicação da *iTunes Store*, da empresa Apple. Esse é um recurso específico, que pode ser implementado utilizando os elementos de programação da linguagem NCL, num sistema que utilize o Ginga como *middleware*.

Apesar do Brasil estar considerando somente os sistemas de rádio digital de banda estreita, o HD Radio e DRM, outros sistemas de radiodifusão como o DMB (*Digital Multimedia Broadcasting*), utilizado na Coreia do Sul, e o ATSC-M/H (*Advanced Television Systems Committee - Mobile/Handheld*), utilizado nos Estados Unidos, apesar de não serem considerados sistema de rádio digital, apresentam abordagens interessantes para aplicações.

O DMB utiliza o BIFS (*Binary Format for Scenes*), que consiste de um formato binário que transporta cenas 2D e 3D, trata da interação com usuário e modificações das cenas ao longo do tempo. BIFS é baseado no padrão VRML (*Virtual Reality Modeling Language*).

O ATSC-M/H adota um sistema baseado no LAsER (*Lightweight Application Scene Representation*), que é tido como mais leve computacionalmente que o BIFS (DUFOURD, 2008). O LAsER utiliza os conceitos de cena 2D do SVG (*Scalable Vector Graphics*), e permite, assim como o BIFS, o *streaming* do conteúdo interativo multimídia.

Tanto o BIFS quanto o LAsER são fortemente baseados nos conceitos do MPEG-4, que define unidades básicas para transporte e modificação de elementos de cena, e permitem a construção de aplicações multimídia com efeitos visuais e de áudio complexos. Esses padrões não utilizam um protocolo de multiplexação baseado no transporte de arquivos em carrossel, mas embutem o paradigma de carrossel para cada objeto de cena da apresentação.

Proporcionando uma experiência de apresentação multimídia com recursos de mídia complexos, esses sistemas focam na construção de aplicações com vários elementos (dinâmicos) de cena, realidade essa que não parece ser possível de ser concretizada em um sistema de largura de banda estreita, no qual a taxa de *bits* disponível para envio da aplicação é pequena. Além disso, poucas ferramentas de autoria estão disponíveis e a adoção de qualquer um desses dois sistemas iria exigir um grande esforço dos radiodifusores para compreender mais um sistema para interatividade, completamente diferente do sistema já utilizado na TV Digital.

É importante ressaltar que para nenhum dos dois sistemas em consideração para adoção no Brasil está definido um middleware para aplicações interativas com especificação aberta. Portanto, o Ginga será o primeiro middleware aberto para um sistema de rádio digital de banda estreita<sup>2</sup>.

### 2.3.Tecnologias utilizadas

A linguagem NCL, por ser uma linguagem de cola, permite que elementos de mídia diferentes sejam embutidos em uma aplicação.

Shiraishi (SHIRAISHI, 2006) propõe o uso da mídia SVG para transmissão de elementos visuais em sistemas com limitações de banda. Uma pesquisa da

---

<sup>2</sup> Sistemas de rádio digital de banda estreita utilizam poucos kHz de largura espectral (HOEG, 2009)

Mozilla Corporation (AAS, 2014) aponta que o formato de imagem baseado no H.265 apresenta maior compressão que todos os outros analisados. Essas análises são muito apropriadas para o contexto do rádio digital devido a grande restrição de banda.

A importância da geolocalização para aplicações interativas executadas em dispositivos móveis, incluindo rádio digital, é tratada por Vaughan-Nichols (VAUGHAN-NICHOLS, 2009). A interface para programação de aplicações *ECMAScript* para acesso às informações de geolocalização por aplicações *Web* é apresentada por Pejić (PEJIC, 2010). Uma interface de programação para acesso às informações de localização é essencial para os aplicativos NCL executados em receptores portáteis e móveis. No entanto, é hoje pobremente suportada pela implementação de referência do Ginga para TV digital.

Um *middleware* para aplicações interativas do rádio digital não deve deixar de lado recursos importantes como primitivas de sincronização fina entre mídias, apresentação distribuída em múltiplos dispositivos e edição ao vivo da aplicação permitidas pelo Ginga (SOARES, 2010) para TV Digital. O Ginga para o rádio digital conta com todos os recursos mencionados, com algumas adaptações.

### 3

## Requisitos para a interatividade na radiodifusão brasileira

Neste capítulo é apresentado um panorama da radiodifusão sonora terrestre brasileira, casos de uso da interatividade no rádio e alguns requisitos levantados para a interatividade no rádio digital que irão guiar algumas decisões apresentadas nos Capítulos 4, 5 e 6.

### 3.1. Quem são emissoras e os ouvintes de rádio no Brasil?

O rádio é o meio de comunicação de massa com maior alcance utilizando-se somente um transmissor, graças ao uso de baixas frequências do espectro eletromagnético (KNELLER, 2013). Nos países de tamanho continental como o Brasil, Rússia e Índia, e países subdesenvolvidos em geral, o rádio é o meio mais importante para transportar informação para as diversas comunidades, principalmente àquelas afastadas de grandes centros urbanos.

No Brasil, de acordo com os dados mais atualizados do Ministério das Comunicações<sup>3</sup>, de 01/10/2013, existem 9024 emissoras de rádio, sendo 4888 comunitárias, 3763 comerciais e 373 educativas.

Segundo Del Bianco (BIANCO, 2012), o Brasil é o segundo país em número de emissoras de rádio no mundo, atrás somente dos Estados Unidos.

Nas emissoras comerciais, a principal fonte de renda é a publicidade, respondendo por mais de 89% da receita total (FGV, 2008). Essas emissoras transmitem conteúdos variados: informações jornalísticas, incluindo informações de trânsito e cobertura de jogos esportivos; programas de entretenimento; proselitismo religioso; e reprodução de músicas mediante *jabá*<sup>4</sup>.

As grandes emissoras comerciais operam em rede, com conteúdo distribuído, normalmente, via satélite, e fazem parte de grandes conglomerados midiáticos (GORGEN, 2009).

---

<sup>3</sup> Site do Ministério das Comunicações: <<http://www.comunicacoes.gov.br/numero-de-emissoras-no-pais>>.

<sup>4</sup> Termo utilizado para se referir às músicas reproduzidas por emissoras mediante pagamento por parte de gravadoras.

Por outro lado, emissoras educativas, comunitárias ou de empresas públicas não têm fins lucrativos. Elas têm como foco a prestação ou divulgação de serviços públicos, educação, divulgação da cultura nacional e regional, além de servir como veículo de comunicação para pessoas que residem na área de cobertura da emissora.

Além dessas emissoras de rádio com outorga, rádios livres, comunitárias, religiosas e indígenas em pleno funcionamento sem concessão no Brasil (SILVA, 2013), apontam para a existência de mais de uma dezena de milhar de emissoras que não são contempladas com outorga especificada por lei.

Além da classificação oficial de emissoras entre comerciais, educativas e comunitárias, é importante ressaltar a frequência de operação das emissoras. No geral, a grande maioria das emissoras operam em Frequência Modulada<sup>5</sup> (FM, 87.4MHz - 108MHz) ou Ondas Médias (OM, 525kHz a 1705kHz), possuindo cobertura adjacente à antena de transmissão da rádio e alcance proporcional à potência de transmissão e à altura da antena (até 78km de raio de cobertura). Todas as emissoras comunitárias transmitem em FM, operam com baixa potência (25W) e possuem raio de cobertura de 1km.

Existem ainda 66 emissoras que operam na faixa de Ondas Tropicais (OT, faixas entre 2300kHz e 5060kHz) e 71 que operam na faixa de Ondas Curtas (OC, faixas entre 5900kHz e 26100kHz). Sinais transmitidos nessas faixas de frequência têm a propriedade especial de serem refletidos pelas camadas E e F da ionosfera terrestre, o que permite um alcance de centenas a milhares de quilômetros do ponto irradiante (DAVIES, 1990). Serviços de rádio recado, principalmente para a região amazônica, centro-oeste e semiárido nordestino, e serviços de alerta de emergência são de grande importância para essas emissoras e seus ouvintes. Muitas igrejas também emitem em Ondas Curtas com o objetivo de serem escutadas em regiões remotas. Exemplos de emissoras OC incluem a Rádio Nacional da Amazônia e a Rádio Verdes Florestas.

Na Tabela 1 estão dados sobre o número das emissoras brasileiras, de acordo com a classificação do Ministério das Comunicações.

---

<sup>5</sup> Algumas vezes os termos FM e AM são empregados para se referir às faixas de frequência nas quais essas modulações são utilizadas, ao invés dos termos VHF Banda II, para FM, e OM, OT ou OC, para AM.

Tipo de emissora	Número	Porcentagem
FM	1982	21,96%
OC	66	0,73%
OT	71	0,79%
OM	1644	18,22%
Comunitária	4888	54,17%
Educativa	373	4,13%
TOTAL	9024	100%

Tabela 1. Divisão por tipo das emissoras de rádio do Brasil.

Vale ressaltar que está em curso uma migração voluntária de emissoras OM para FM autorizada pelo Ministério das Comunicações, o que pode implicar na migração de mais de 80% das aproximadamente 1800 emissoras OM para o FM (SET, 2013). No entanto, emissoras OM públicas e outras emissoras que transmitem com alta potência não irão migrar e continuarão mantendo a grande cobertura propiciada por essa faixa de frequência, apesar da estreita largura de banda dos canais. Com a radiodifusão digital as emissoras OM irão contar com qualidade de áudio semelhante às atuais emissões em FM.

Na outra ponta da cadeia de radiodifusão, receptores de rádio estão presentes em 88% das residências, 80% dos automóveis em circulação e em 36% dos aparelhos celulares, e sua penetração ultrapassa 90% da população brasileira (IBGE, 2010).

Com relação aos tipos de receptor, podemos classificá-los como de recepção fixa, móvel e portátil.

Os rádios de recepção fixa são os “receptores de mesa”, tipicamente localizados nas residências e em locais de trabalho, fazendo parte desse grupo também receptores de rádio embutidos em televisores e centrais de mídia domésticas. Dentre esses receptores estão aqueles que possuem tela com boa definição e uma unidade de processamento de aplicação (*application processor*) como as TVs e centrais de mídia, e outros que são bastante simples e não possuem visor nem unidade de processamento de aplicações, tendo um sintonizador, demodulador e uma ou duas caixas de som.

Os de recepção móvel são tipicamente encontrados em automóveis. Dentre os autorrádios, os mais antigos são simples e não possuem processador de



aplicação nem uma tela com boa definição e tamanho. No entanto, os receptores de rádio para automóveis mais recentes vêm integrados em um sistema conhecido como *In-Vehicle Infotainment* (IVI) (MACARIO, 2009). Os sistemas IVI são compostos por um computador central com acesso a informações provenientes do sintonizador de rádio, GPS e Internet, além de uma tela com boa resolução, muitas vezes sensíveis ao toque. O suporte a tecnologias como WiFi e Bluetooth permite que uma aplicação possa ser executada de forma distribuída em múltiplos dispositivos de exibição, tais como celular e *tablet*, interconectados através do sistema IVI. A Figura 4 apresenta os elementos constituintes de um sistema IVI.

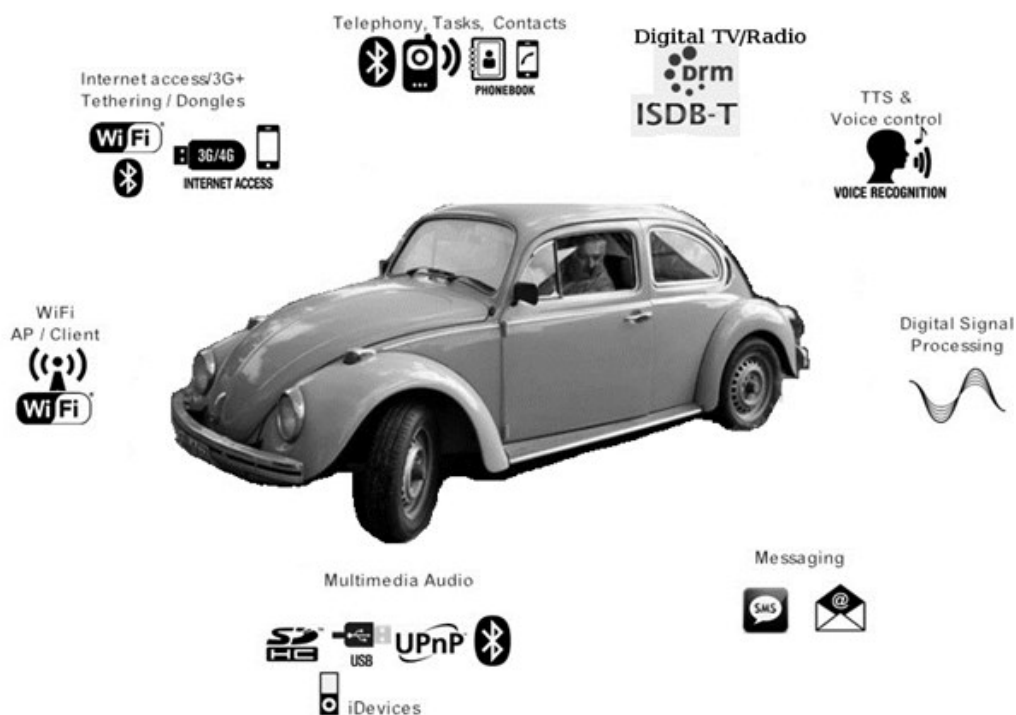


Figura 4. Tecnologias que fazem parte de um sistema In-Vehicle Infotainment (IVI). Fusca do ex-presidente do Uruguai Pepe Mujica meramente ilustrativo.

Receptores portáteis são os receptores de tamanho reduzido, alimentados por pilha, ou receptores de rádio embutidos em aparelhos celular e tocadores de mídia portáteis. Atualmente os “radinhos de pilha” são tipicamente simples e não contam com tela e recursos computacionais, enquanto os receptores de rádio embutidos em celulares e tocadores de mídia portáteis tem acesso a um processador de aplicação e tipicamente possuem telas com boa resolução.

Grande parte dos receptores automotivos, parte dos receptores de mesa, televisores com rádio embutido, centrais de mídia e receptores embutidos em celular possuem capacidade de processamento e memória suficiente para executar o GINGA, graças ao baixo consumo de recursos por parte do *middleware* brasileiro.

A interação do usuário com o receptor de rádio através do pressionamento de botões discretos ou de tela sensível ao toque deverá seguir o modelo hierárquico de controle do foco e eventos de entrada (MORENO, 2013b), assim como na TV Digital.

Os usuários desses receptores de rádio são marcados pela sua heterogeneidade, que vai desde a grande parte dos 33 milhões de analfabetos funcionais brasileiros até as classes mais altas da pirâmide social do país, que usualmente escutam rádio em automóveis com receptores de alta qualidade. Outros 16 milhões de brasileiros com deficiência visual parcial ou total (CASTRO, 2014) utilizam o rádio como um importante meio de informação e lazer.

### **3.2. Aplicativos vislumbrados para o rádio digital**

É possível vislumbrar alguns tipos de aplicativos interativos que serão interessantes para emissoras e ouvintes de rádio digital. Os aplicativos comumente estarão ligados a temas como publicidade, esportes, notícias, informações de trânsito, religião e ao conteúdo musical de uma emissora.

Nas rádios comerciais, aplicativos relacionados à publicidade deverão ter um papel fundamental. Em rádios comerciais de nicho como, por exemplo, as que transmitem informações de trânsito, a interatividade pode aumentar muito o poder de informação dessas emissoras através do envio de mapas e situação do tráfego.

Em emissoras públicas e comunitárias, aplicativos interativos poderão ser utilizados para educação a distância, distribuição de informações sobre serviços públicos e atividades na comunidade. Muitas novas possibilidades surgirão com a interatividade no rádio dada a grande capilaridade, estilo e número de emissoras.

A crescente penetração da Internet na sociedade torna possível aplicações híbridas, que permitam uma convergência entre as redes de dados ponto-a-ponto e a rede de radiodifusão. Muitos aparelhos de telefonia móvel com acesso à internet

terão receptor de rádio digital embarcado, além das centrais de mídia domésticas e automobilísticas que muitas vezes também contam, ou contarão, com acesso à Internet.

Uma característica importante no desenvolvimento de aplicativos para o rádio digital é a taxa de bits disponível para sua transmissão. A Tabela 2 apresenta as taxas de bits do fluxo transmitido através do Rádio Digital de acordo com o tipo de emissora, em comparação às taxas obtidas na TV Digital, tomando-se como referência configurações de robustez do sinal tipicamente utilizadas. Nas colunas da tabela, estão presentes os dois modos típicos do ISDB-T, *Full-Seg* (6MHz de banda, para recepção fixa) e *One-Seg* (430kHz de banda, para recepção móvel e portátil); e as faixas AM (canal de 10kHz) e FM (canal de 100kHz) do Rádio Digital.

TV Full-Seg	TV One-Seg	FM	AM
18 Mbit/s	420 kbit/s	100 kbit/s	24 kbit/s

Tabela 2. Taxas de bits típicas do broadcast digital.

Devido à estreita banda passante de um canal de rádio digital, o conteúdo interativo enviado terá limitações de tamanho importantes, que serão consideradas nas definições da NCL e do Ginga para o rádio digital.

Os aplicativos interativos para rádio digital devem estar aptos a serem reproduzidos em diferentes tamanhos de tela, dada à heterogeneidade de receptores.

### 3.3. Requisitos levantados

Levando-se em conta as considerações técnicas e o panorama da radiodifusão brasileira apresentado, é possível inferir alguns requisitos específicos para o rádio essenciais, que devem ser atendidos por uma linguagem e seu *middleware*, de forma a permitir que os aplicativos possam explorar ao máximo o potencial interativo do rádio. Esses requisitos estão expostos na Listagem 1.

- 1 **Otimizar a taxa de transmissão.** Muitas das bandas de rádio (OM, OT, OC) tem grande restrição na taxa de bits, mas todas são de grande relevância para a sociedade e a radiodifusão.
- 2 **Suporte a recursos multimídia.** Garantir que recursos visuais e aurais estejam disponíveis, assim como as primitivas para sincronização fina

entre as mídias, respeitando-se diferentes perfis de receptores, com telas, caixas de som e capacidade de processamento distintas.

- 3 **Interatividade por áudio facilitada.** Adicionar recursos que aprimorem a interatividade por áudio para os casos de uso por pessoas com restrições visuais e analfabetos funcionais.
- 4 **Suporte ao acesso de informações de geolocalização.** Prover meios que permitam a adaptação da aplicação de acordo com a geolocalização do receptor, dada a grande relevância dessa informação de contexto na recepção móvel do rádio.
- 5 **Menor complexidade da implementação do middleware.** Muitos dos receptores de rádio digital não têm grande capacidade computacional. É importante que o middleware, sempre que possível, economize esses recursos.
- 6 **Linguagem que dê suporte a radiodifusão híbrida.** É importante que a linguagem e o middleware permitam aplicações convergentes entre redes de rádio, TV, IP e de telefonia.

Listagem 1. Requisitos para um rádio digital interativo.

## 4

### Transporte de aplicações via rádio digital

Para o envio de uma aplicação e suas mídias através de uma transmissão de rádio digital, é necessário o uso de um protocolo que especifique a forma de multiplexar arquivos, diretórios e comandos de edição no fluxo de *bits* transmitidos pela emissora.

As definições contidas neste capítulo focam no atendimento do requisito 2, Listagem 1, principalmente no que tange o provimento dos mecanismos para a sincronização entre mídias, sinalização de diferentes perfis de receptores, além do suporte ao envio de arquivos e diretórios com pouco desperdício de bits.

Pelo fato do Brasil estar considerando para adoção<sup>6</sup> ou o sistema Digital Radio Mondiale (ETSI, 2014) ou o HD Radio (NRSC, 2011a), para ambos padrões serão definidos parâmetros e configurações de multiplexação.

Neste capítulo, portanto, são apresentados os protocolos e parâmetros de transmissão utilizados nesses dois sistemas. Também é apresentado o esquema da URI, para permitir que aplicações NCL possam referenciar os diferentes objetos e fluxos de mídia multiplexados no sinal.

#### 4.1.Digital Radio Mondiale

O sistema Digital Radio Mondiale é um sistema de radiodifusão sonora terrestre que funciona em todas as bandas definidas para radiodifusão, sendo definido por um conjunto de normas públicas. As camadas física e de multiplexação do DRM são apresentadas em sua norma de especificação de sistema (ETSI, 2014). Outros protocolos que definem a transmissão de arquivos e aplicações são definidos em outras normas, que serão discutidas nesta seção.

A modulação utilizada pelo DRM é a COFDM (*Coded Orthogonal Frequency-Division Multiplexing*) e vários parâmetros relacionados à robustez do sinal e à largura de banda estão disponíveis para o radiodifusor. Dois

---

<sup>6</sup> Em resposta ao chamamento público para testes com rádio digital de 13 de junho de 2011 publicado no Diário Oficial da União, dois sistemas se apresentaram, o DRM e o HD Radio.

codificadores de áudio podem ser utilizados, o HE-AACv2 (ISO, 2009) e o xHE-AAC (ISO, 2012), sendo que é permitido o uso da extensão MPEG *Surround* (ISO, 2007) para transmissão multicanal de forma retro compatível com decodificadores que não suportam tal recurso. O codificador xHE-AAC é uma introdução recente ao padrão, de 2014, e ainda não existe nenhum receptor no mercado compatível com o novo codificador.

Apesar de ambos DRM e ISDB-T utilizarem o codificador HE-AACv2, seus perfis são diferentes: no rádio é utilizado a variante do codificador que utiliza 960 amostras para a realização da etapa da Transformada Discreta de Cosseno, enquanto na TV Digital a variante com 1024 amostras é utilizada.

Apesar disso, um decodificador AAC que suporta ambas variantes utilizadas no ISDB-T e DRM é coberto pelos mesmos royalties de um decodificador que suporte somente uma variante. Dessa forma, é possível uma economia de gastos no receptor compatível com TV e rádio digitais, quando os sistemas compartilham o mesmo codificador, como no caso do ISDB-T e DRM.

Até 4 fluxos de informação ou *MSC Streams* podem ser transmitidos pelo DRM, sendo que cada fluxo pode conter áudio ou dados. Um fluxo de dados pode ser particionado em até 4 sub-fluxos em sua configuração conhecida como *Packet mode*, ou utilizado de forma completa. Esses fluxos são mapeados por serviços, sendo que até 4 serviços são permitidos. Cada serviço pode ser de áudio ou dados. Um serviço de áudio é associado a um fluxo de áudio e pode ser associado a até 4 (sub-)fluxos de dados, que recebem a denominação de *Program Associated Data* (PAD). Um serviço de dados é associado a um (sub-)fluxo de dados. As informações que compõem um aplicativo NCL são transmitidas por um sub-fluxo de dados no modo *Packet mode*. A Figura 5 apresenta um recorte da norma do DRM com um exemplo de mapeamento dos fluxos em serviços.

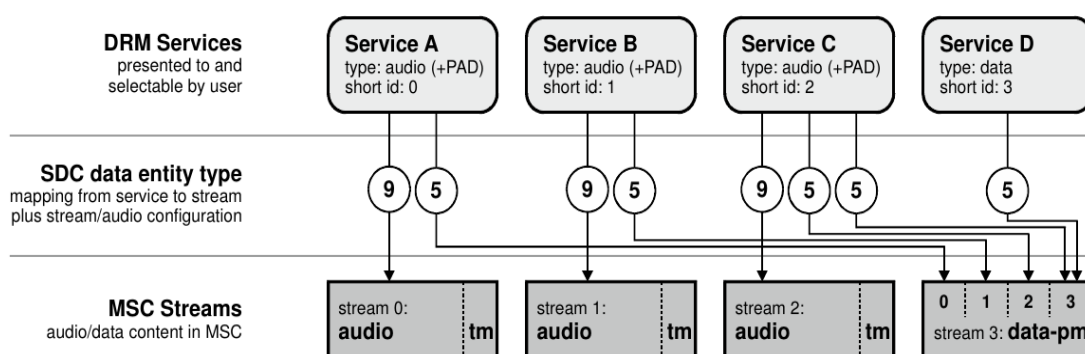


Figura 5. Exemplo de configuração do DRM mostrando um exemplo de mapeamento entre os serviços e os fluxos de transmissão.

Três canais lógicos são definidos no DRM: *Fast Access Channel* (FAC), *Service Description Channel* (SDC) e *Main Service Channel* (MSC).

O canal FAC contém informações que descrevem os parâmetros de configuração do canal necessárias para a demodulação completa do sinal, como ocupação espectral e tipo de modulação dos canais SDC e MSC. O FAC contém também informações a respeito do número de serviços presentes e carrega informações sobre esses serviços.

Os parâmetros do FAC para descrição dos serviços são os seguintes: *Service identifier*, *Short Id*, *Audio CA indication*, *Language*, *Audio/Data flag*, *Service descriptor* e *Data CA indication*. O *Service identifier* é um identificador único do serviço de 24bits com regras específicas de criação, incluindo, por exemplo, o código de país do serviço; *Short Id* é um identificador único de 2bits de um serviço dentro do multiplex; *Audio CA indication*, de 1 bit, indica se o serviço de áudio utiliza Acesso Condicionado ou não; *Language*, de 4 bits, indica a língua utilizada no serviço; *Audio/Data flag*, de 1 bit, indica se o serviço é de áudio ou dados, *Service descriptor*, de 5 bits, para o caso de um serviço de áudio, indica o tipo do programa, ou para o caso de serviços de dados, indica o tipo da aplicação; e *Data CA indication*, de 1 bit, indica a presença de serviço de dados com Acesso Condicionado.

Para o caso de um aplicativo NCL ser transmitido como um serviço de dados independente, o *Service descriptor*, que para serviços de dados representa o parâmetro *Application identifier*, deve assumir o valor 4, que indica que o serviço contém um aplicativo NCL, conforme a Tabela 3. O valor 4 é provisório, sendo o primeiro valor sem alocação do parâmetro, e poderá ser modificado até que haja uma alocação oficial para aplicação NCL no FAC.

<i>FAC Service Parameter</i>	Valor
<i>Service Descriptor</i> ( <i>Application identifier</i> )	4 (Provisório)

Tabela 3. Valor do *Application identifier* no FAC para sinalizar a presença de uma aplicação Ginga em um serviço de dados.

No processo de varredura do espectro em busca de emissoras e seus serviços por parte de um receptor DRM, as informações presentes na FAC são utilizadas para compor a lista de serviços disponíveis em dada região.

O canal SDC contém as informações necessárias para o mapeamento dos serviços em fluxos do canal MSC. O SDC contém também entidades que permitem a transmissão de outras informações como rótulos dos serviços e data atual. Existem entidades de transmissão obrigatórias e outras opcionais.

As entidades do SDC são: *Multiplex description*, *Label*, *Conditional Access Parameters*, *AFS - Multiple frequency network information*, *AFS - Schedule definition*, *Application information*, *Announcement support and switching*, *AFS - Region definition*, *Time and date information*, *Audio information*, *FAC channel parameters*, *AFS - Other services*, *Language and country*, *AFS - Region definition*, *Packet stream FEC parameters* e *Extension*.

Muitas dessas entidades estão relacionadas ao *Alternative Frequency checking and Switching* (AFS), recurso que permite a sinalização de frequências alternativas e comandos de chaveamento permitindo que um receptor de rádio possa chavear para uma frequência alternativa analógica ou digital para casos como troca de frequência da emissora, sinal com intensidade insuficiente ou situações de emergência. O recurso AFS é pouco relevante para a definição do transporte de aplicações NCL, pois trocando-se a frequência para um multiplex não síncrono, todo o processo de resintonização é feito, e para um multiplex síncrono, a transição de frequência é transparente para o middleware, sendo trocada somente a propriedade que expõe a frequência de sintonia da emissora para a aplicação.

Dentre as entidades do SDC não relacionadas ao AFS estão *Multiplex description*, que contém informações sobre os fluxos presentes no canal MSC e informações para demodulação dos fluxos do canal MSC; *Label* contém uma cadeia de caracteres que dá nome a um serviço referenciado pelo *Short Id*; *Conditional access* contém informações relacionadas a acesso condicionado; *Announcement support and switching*, que permite o envio de alertas, de clima e de emergência, por exemplo; *Time and date information* que envia a data, horas e minutos em UTC, além do fuso horário; *Language and country*, que permite a transmissão da língua e país em forma de caracteres nos padrões ISO 639-2 e ISO 3166, respectivamente; *Packet stream FEC parameters* define parâmetros para um código corretor de erros extra que pode ser aplicado a fluxos de dados no modo pacote e *Extension* permite o envio de entidades adicionais.



Duas entidades são de grande relevância, a *Application information* e a *Audio information*. A *Application information* é a entidade número 5 da SDC e faz o mapeamento entre fluxos de dados e serviços de áudio ou dados, ou seja, associa um serviço através de seu *Short Id* a um fluxo de dados do MSC através do *Stream Id* e um *Packet Id*, no caso de operação no modo de pacotes. A aplicação NCL deverá ser transmitida utilizando o campo *Packet mode indicator* com valor 1, que significa operação em modo pacotes, e *data unit indicator* também com valor 1, que permite a transmissão de unidades de dados com tamanho independente do tamanho do pacote utilizado no multiplex. Essa configuração é pré-requisito (ETSI, 2009) para o uso do protocolo *Multimedia Object Transfer* (ETSI, 2006b), que é o protocolo definido para o DRM para transmissão de arquivos em carrossel e, portanto, será adotado para transportar aplicações NCL.

Aplicações NCL normalmente serão transmitidas associadas a um serviço de áudio. No entanto, nada impede que um aplicativo NCL seja transmitido como um serviço de dados independente, com a possibilidade da reprodução de um fluxo de áudio disparado a partir de uma aplicação NCL.

Outros campos relevantes da entidade *Application information* são *application domain*, que deve ter o valor 0, indicando o sistema DRM e *application data*, que contém um identificador para o tipo de aplicação, além de informações definidas pela aplicação. O identificador presente em *application data* tem nome *user application identifier*, e até ser definido um valor oficial para o Ginga, o valor utilizado será 0x001, que é o primeiro identificador disponível para aplicações com especificação aberta na norma.

A Tabela 4 apresenta os campos da entidade *Application information* da SDC relacionados à transmissão de um aplicativo NCL.

<i>Application information parameters</i>	Valor
<i>Packet mode indicator</i>	1
<i>data unit indicator</i>	1
<i>application domain</i>	0
<i>user application identifier</i>	0x0001 (Provisório)

Tabela 4. Campos e valores da entidade *Application information* do canal SDC para transmissão de aplicativos NCL.

A *Audio information* é a entidade de número 9 da SDC e define o mapeamento entre um serviço de áudio, referenciado pelo *Short Id*, e o fluxo MSC de áudio, indicado no *Stream Id*. Também são transmitidos todos os parâmetros de codificação do áudio e um campo de nome *text flag* que indica a presença de mensagens de texto no fluxo de áudio. O recurso de mensagens de texto permite, por exemplo, o envio de manchetes e pequenas frases embarcadas no fluxo de áudio.

O protocolo *Multimedia Object Transfer* (ETSI, 2006b), foi inicialmente concebido para o sistema de rádio digital Digital Audio Broadcast (DAB), para permitir o envio de imagens e outras aplicações. Posteriormente foi mapeado (ETSI, 2009) para ser transportado pelo sistema de multiplexação do DRM, através de um sub-fluxo de dados em *Packet mode*.

O MOT provê dois modos de operação, o modo *MOT Header* e o modo *MOT Directory*.

O modo *Header* permite a transmissão e controle de somente um objeto por vez, enquanto o modo *Directory* permite o envio de um conjunto de objetos e o controle dos mesmos, como inclusão, remoção e atualização. Pelo fato de uma aplicação NCL ser composta ao menos de um documento NCL e mídias, o modo *Directory*, que implementa um tipo de transmissão em carrossel para múltiplos arquivos, deverá ser utilizado para transmissão de aplicativos NCL.

O protocolo MOT define duas estruturas no modo *Directory*, o *MOT Body* e o *MOT Directory*. O *MOT Body* contém o conteúdo dos objetos, no caso, arquivos, da aplicação, e carrega um identificador único, o *TransportId*. O *MOT Directory* contém informações sobre o carrossel como um todo, o número de objetos no carrossel e um conjunto de informações associadas a cada objeto, identificados pelo *TransportId*.

A estrutura do *MOT Directory* é apresentada na Figura 6, extraída da norma

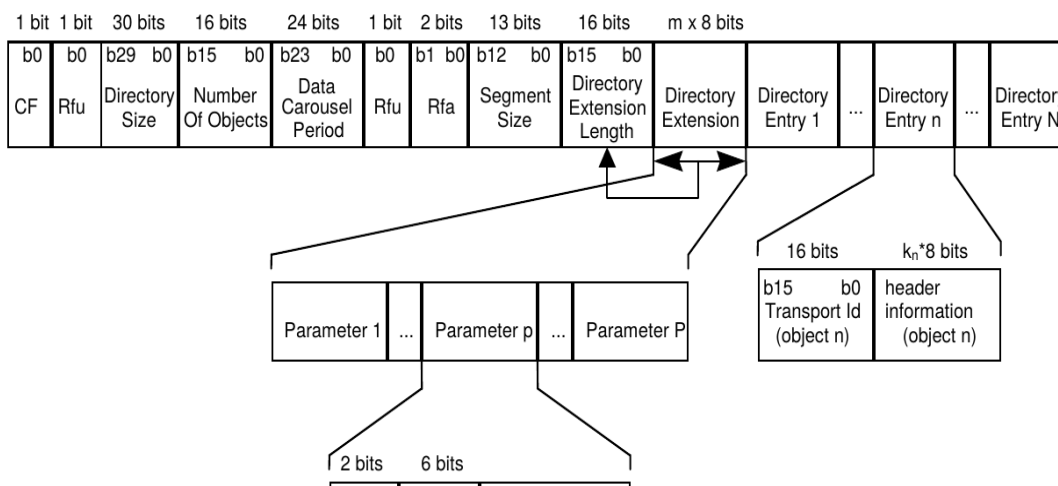


Figura 6. Estrutura do *MOT Directory*.

do MOT.

As informações presentes no MOT *Directory* referentes ao carrossel como um todo são: *DirectorySize*, *NumberOfObjects*, *DataCarouselPeriod*, *SegmentSize*, *DirectoryExtensionLength* e *DirectoryExtension*. *DirectorySize* indica o tamanho da estrutura do MOT *Directory*, em bytes; *NumberOfObjects* indica o número de objetos presentes no carrossel; *DataCarouselPeriod* indica o tempo máximo, em décimos de segundos, para a retransmissão de todos os objetos do carrossel. O valor 0 indica período indefinido; *SegmentSize* indica o tamanho, em bytes, que é utilizado para a segmentação dos objetos do carrossel. O valor 0 indica que os segmentos têm tamanhos diferentes; *DirectoryExtensionLength* indica o total em bytes do *DirectoryExtension*; e *DirectoryExtension* contém parâmetros que descrevem o carrossel.

A entidade *DirectoryExtension* contém parâmetros relacionados ao carrossel como um todo. A sintaxe do *DirectoryExtension* é composta por uma sequência de parâmetros, sendo que cada parâmetro utiliza um dentre os formatos apresentados na Figura 7.

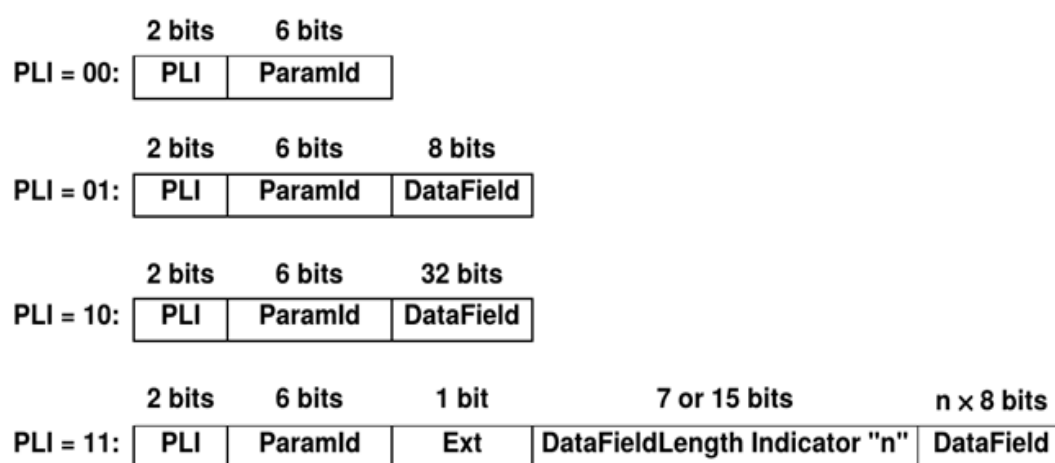


Figura 7. Sintaxe de um parâmetro do *DirectoryExtension* ou *HeaderExtension*.

Cada parâmetro tem seu tamanho e campos determinados pelo *Parameter Length Indicator* (PLI), assim como indicado na Figura 7. A lista dos identificadores dos parâmetros permitidos no *DirectoryExtension* é apresentada na Figura 8.

Parameter Id b <sub>5</sub> b <sub>0</sub>	Parameter	Definition	Possible occurrences	Usage mandatory for content provider	Support mandatory for MOT decoders
00 0000	SortedHeaderInformation	7.2.4.1	only once	no	no
00 0001	DefaultPermitOutdatedVersions	7.2.4.2	only once	no	no
00 0010	reserved for MOT protocol extensions				
...					
00 1000					
00 1001	DefaultExpiration	7.2.4.3	only once	no	yes, if receiver provides "MOT caching support"
00 1010	reserved for MOT protocol extensions				
...					
01 1111					
10 0000	reserved for user application specific parameters				
10 0001					
10 0010	DirectoryIndex (user application specific parameter)	see [6]	once or several times	see [6]	see [6]
10 0011	reserved for user application specific parameters				
...					
11 1111					

Figura 8. Parâmetros permitidos no *DirectoryExtension*.

O parâmetro *SortedHeaderInformation* indica que a lista dos objetos do MOT *Directory* está ordenada na ordem alfabética do parâmetro *ContentName*; *DefaultPermitOutdatedVersions* indica que em caso de atualização de algum objeto de carrossel, a versão antiga ainda pode ser exibida enquanto a nova versão do objeto é obtida pelo receptor; *DefaultExpiration* indica um tempo, que pode ser absoluto ou relativo, de quando o objeto irá expirar e se tornar inválido, ou ainda se o objeto nunca expira; *DirectoryIndex* indica o documento que deverá ser exibido inicialmente; e três novos parâmetros são definidos.

Os novos parâmetros são introduzidos para suprir requisitos do Ginga relacionados a edição e controle em tempo real da aplicação, e sincronização fina entre mídias e fluxos de áudio. O *EditingCommandEvent* dá suporte ao envio de comandos de edição, o *TimeBaseReference* carrega uma base temporal, e *TimeBaseEndpoint* provê suporte para iniciar e parar bases temporais.

A Tabela 5 indica os parâmetros do *DirectoryExtension* que devem ser suportados de forma mandatória.

Identificador	Parâmetro
0x22 (100010)	<i>DirectoryIndex</i>
0x23 (100011)	<i>EditingCommandEvent</i>
0x24 (100100)	<i>TimeBaseReference</i>
0x25 (100101)	<i>TimeBaseEndpoint</i>

Tabela 5. Parâmetros do *DirectoryExtension*.

A sintaxe do parâmetro *EditingCommandEvent* é definido na Tabela 33, *TimeBaseReference* na Tabela 31 e *TimeBaseEndpoint* na Tabela 32.

A sintaxe do parâmetro *DirectoryIndex* é apresentada na Tabela 6, sendo a mesma utilizada pela aplicação *Broadcast Website* (ETSI, 2006a).

Sintaxe	Tamanho	Tipo
<i>DirectoryIndex_parameter_data_field()</i> {		
<i>profile_id</i>	8 bits	Uimsbf <sup>7</sup>
<i>for (i=0;i&lt;N;i++)</i> {		
<i>index_name_byte</i>	8bits	Uimsbf
}		
}		

Tabela 6. Sintaxe do campo de dados do parâmetro *DirectoryIndex*.

O campo *profile\_id* contém o perfil do receptor. Até que outros perfis sejam propostos, *profile\_id* deve conter o valor 1, indicando perfil completo do Ginga para rádio, contemplando todos os recursos definidos neste texto. Outros valores ficam reservados para futura definição de perfis de receptores com diferentes recursos. Na TV Digital, por exemplo, são definidos 6 perfis de receptores, sendo eles OSA, OSB, OSC, FSA, FSB, FSC (ABNT, 2015), sendo os dois últimos os perfis mais interessantes, que permitem a reprodução de vídeo proveniente do carrossel e o armazenamento persistente de aplicações na memória do receptor para posterior execução por parte do usuário.

O parâmetro *DirectoryIndex* é associado a um perfil, permitido que o receptor execute a aplicação NCL apropriada para o seu perfil. Possíveis perfis de receptores poderão ser definidos, por exemplo, para contemplar recursos de acessibilidade, como suporte a reconhecimento de voz, para pessoas com restrição de movimento. Normalmente os perfis de receptores são definidos com a participação da indústria de receptores, portanto, não serão definidos vários perfis neste texto, além do perfil completo representado pelo valor 1 no *profile\_id*.

Já o parâmetro *EditingCommandEvent* carrega um comando de edição.

Mais de um parâmetro de cada tipo pode ser transportado por cada entidade *DirectoryExtension*, permitindo o envio de mais de um *DirectoryIndex* e *EditingCommandEvent*.

<sup>7</sup> Uimsbf: Unsigned Integer, transmitted Most Significant Bit First

O campo *index\_name\_byte* do parâmetro *DirectoryIndex* contém os bytes que compõem o nome do arquivo NCL que deverá ser o ponto de entrada da aplicação para o perfil de receptor indicado. Também é possível utilizar o argumento *InterfaceId* para indicar a porta da aplicação NCL a ser disparada no início da aplicação, recurso que pode ser útil para aplicações com suporte a múltiplos perfis de receptores.

A Tabela 7 apresenta as duas sintaxes válidas para o *index\_name* do parâmetro *DirectoryIndex*.

Sintaxes possíveis - <i>index_name</i>	Descrição
<i>string</i> (ex: nome_do_arquivo.ncl)	Informa ao Ginga o caminho da aplicação NCL a ser iniciada.
<i>string</i> [,idRef] (ex: nome_do_arquivo.ncl, porta1)	Dois argumentos separados por vírgula. O primeiro informa ao Ginga o caminho da aplicação NCL e o segundo argumento especifica a porta a ser disparada, e é opcional.

Tabela 7. Possíveis sintaxes para o *index\_name*.

Os parâmetros *SortedHeaderInformation*, *DefaultPermitOutdatedVersions* e *DefaultExpiration* são opcionais e podem ou não serem transmitidos pelas emissoras. O receptor deve seguir a norma do protocolo MOT e corretamente interpretar esses parâmetros e operar a base privada dos documentos de forma apropriada.

Além das entidades relacionadas ao carrossel como um todo, o MOT *Directory* é composto também pelas entidades que identificam os objetos individualmente presentes no carrossel. Cada objeto presente no carrossel é descrito pelo *DirectoryEntry*. O *DirectoryEntry* é composto pelo *TransportId* e pelo *Header information*, que por sua vez é composto por *Header core* e *Header extension*.

O *Header core* é composto por *BodySize* que indica o tamanho do objeto; *HeaderSize* que indica o tamanho das informações de cabeçalho; *ContentType* e *ContentSubType* que indicam o tipo do conteúdo.

Todos os arquivos transmitidos deverão conter em *ContentType* o valor 0, indicando general data, e em *ContentSubType* o valor 0, indicando *Object Transfer*. Os tipos e subtipos definidos na norma do MOT para indicar atualização das informações do carrossel são válidos e devem ser suportados. A Tabela 8

apresenta o valor que *ContentType* e *ContentSubType* devem assumir para a transmissão dos arquivos de uma aplicação NCL.

ContentType	ContentSubType	Interpretação
0	0	<i>File transfer</i>

Tabela 8. Valor de *ContentType* e *ContentSubType* possível para cada entrada do carrossel.

O *Header extension* é composto por uma sequência de parâmetros. Os parâmetros definidos em norma são mostrados na Figura 9, que contém uma tabela extraída da norma do MOT.

Parameter Id b <sub>5</sub> b <sub>0</sub>	Parameter	Definition	Possible occurrences	Usage mandatory for content provider	Support mandatory for MOT decoders
00 0000	<b>reserved for MOT protocol extensions</b>				
00 0001	PermitOutdatedVersions	6.2.3.1.2	only once	no	no
00 0010 00 0011 00 0100	<b>reserved for MOT protocol extensions</b>				
00 0101	TriggerTime (user application specific parameter)	see [5]	see [5]	see [5]	see [5]
00 0110	<b>reserved for MOT protocol extensions</b>				
00 0111	RetransmissionDistance	6.2.3.1.5	only once	no	no
00 1000	<b>reserved for MOT protocol extensions</b>				
00 1001	Expiration	6.2.3.1.1	only once	no	yes, if receiver provides "MOT caching support"
00 1010	Priority	6.2.3.1.4	only once	no	no
00 1011	Label (user application specific parameter)	see [6]	only once	no	no
00 1100	ContentName	6.2.2.1.1	only once	yes	yes
00 1101	UniqueBodyVersion	6.2.3.1.3	only once	no	no
00 1110 00 1111	<b>reserved for MOT protocol extensions</b>				
01 0000	MimeType	6.2.2.1.2	only once	user application specific	user application specific
01 0001	CompressionType	6.2.2.1.3	only once	yes (if body is compressed)	<b>yes</b> ; every receiver must check if an object is compressed
01 0010 ... 01 1111	<b>reserved for MOT protocol extensions</b>				
10 0000	AdditionalHeader (user application specific parameter)	see [6]	once or several times	see [6]	see [6]
10 0001	ProfileSubset	6.2.3.3.1	only once	no	no
10 0010	<b>reserved for MOT protocol extensions</b>				
10 0011	CAInfo	6.2.3.2.1	only once	yes (if CA is used)	<b>yes</b> ; every receiver must check if an object is scrambled
10 0100	CAReplacementObject	6.2.3.2.2	only once	no	no
10 0101 ... 11 1111	<b>reserved for user application specific parameters</b>				

Figura 9. Possíveis parâmetros do *Header extension* para objetos do carrossel.

Os parâmetros relevantes para o transporte das partes de uma aplicação são: *PermitOutdatedVersions* que define se após a sinalização de uma nova versão de um arquivo, o antigo ainda deve ser apresentado até a nova versão estar disponível; *Expiration* indica o momento que o arquivo expira, ou se o mesmo não expira; *ContentName* indica o nome do objeto; *TriggerTBV*, que é um novo parâmetro introduzido para aplicações NCL (não presente na Figura 9), e representa o momento no qual o objeto deve ser apresentado, e *CompressionType* o tipo de compressão utilizada pelo objeto. Vale ressaltar que o parâmetro *TriggerTBV* deve ser utilizado de forma mutuamente exclusiva com o comando de edição NCL *startDocument*, e o parâmetro *TriggerTime* não deve ser utilizado.

Os parâmetros *PermitOutdatedVersions* e *Expiration* são opcionais na transmissão, mas devem ser corretamente interpretados pelo receptor. A Tabela 9 apresenta os parâmetros mais importantes do *Header extension*.

Identificador	Parâmetro	Valor	Interpretação
0x0C (001100)	ContentName	Contém a codificação de caracteres e o nome do arquivo.	O nome do objeto, que indica um caminho relativo de um arquivo da aplicação. Exemplo: “img/img1.jpg”
0x11 (010001)	CompressionType	0x1 (gzip / DEFLATE)	Para ser utilizado somente quando o <i>MOT body</i> , ou seja, o conteúdo do arquivo for comprimido para transmissão.
0x25 (100101)	TriggerTBV	Utiliza a mesma sintaxe do parâmetro TimeBaseReference, especificado na Tabela 31.	Indica o momento na qual a aplicação deve ser iniciada. Pode também carregar um valor



			com significado “faça agora”. Uso aplicável para arquivos NCL indicados no <i>DirectoryIndex</i> .
--	--	--	--

Tabela 9. Parâmetros importantes do *Header extension* para os objetos do carrossel.

Os nomes dos arquivos não devem começar com “/”, e os diretórios devem ser indicados com uma “/” após seus nomes. Exemplo de nome de arquivo em *ContentName*: “images/game1/pic1.jpg”.

O *MOT Body* deve ser comprimido sempre que o *DirectoryEntry* que o identifica conter o parâmetro *CompressionType*, e o *MOT Directory* também pode ser comprimido, sendo que sua forma de compressão é apresentada na seção 7.2.8 da norma do protocolo MOT (ETSI, 2006b). A compressão é opcional na transmissão, porém deve ser suportada pelos receptores.

Elementos presentes no multiplex do DRM que não fazem parte do carrossel podem ser referenciados em um documento NCL através do atributo *src* do elemento *<media>*. A URI apresentada na Tabela 10, assim como definido em norma (ETSI, 2015c), deve ser utilizada para referenciar uma mídia que não faça parte do carrossel pelo qual a aplicação NCL é carregada.

Esquema	Parte específica do esquema	Interpretação
drm:	<sid>[.<appdomain>.<uatype>]	<i>sid</i> é o <i>Service Identifier</i> do serviço (6-char). <i>appdomain</i> é o <i>application domain</i> . Valores possíveis: 0 para indicar DRM e 1 para DAB (1-char). <i>uatype</i> indica o tipo da aplicação. (4-char).

Tabela 10. URI para referenciar serviços no multiplex DRM.

Para se referenciar um serviço de dados, *appdomain* e *uatype* podem ser omitidos, mas caso não sejam omitidos, o tipo do serviço deve estar de acordo com o *appdomain* e *uatype* especificados na URI.

Para se referenciar um serviço de áudio, *appdomain* e *uatype* devem ser obrigatoriamente omitidos, e para se fazer referência a um conteúdo de dados associado a um programa de áudio como PAD (*Program Associated Data*), deve-se utilizar obrigatoriamente os campos *appdomain* e *uatype* para a identificação do conteúdo desejado através do tipo da aplicação.

A URI especificada na Tabela 10 permite, por exemplo, que uma aplicação NCL possa controlar o volume do áudio do serviço ou reproduzir uma aplicação do tipo *MOT SlideShow* como um objeto de mídia.

## 4.2.HD Radio

O sistema HD Radio, conhecido nas normas como IBOC, é um sistema de rádio digital que foi concebido para operar nos canais adjacentes do sinal analógico AM ou FM nas bandas de Ondas Médias e VHF.

Diferente do sistema DRM, a especificação do HD Radio não é totalmente pública, sendo a definição das aplicações e o codificador de áudio proprietários (ANDERSON, 2011). Para transmitir um sinal HD Radio as emissoras necessitam de uma licença que deve ser obtida junto à empresa detentora dos direitos do sistema, a Iiquity Digital Corporation<sup>8</sup>.

As camadas física (transmissão em rádio-frequência) e de multiplexação do HD Radio são descritas pela sua norma de sistema (NRSC, 2011a).

Na norma do HD Radio o conteúdo a ser transmitido é classificado em serviços, sendo eles o MPS (*Main Program Service*), o SPS (*Supplemental Program Service*), o SIS (*Station Information Service*) e o ADS (*Advanced Data Services*). Todo o conteúdo desses serviços é particionado em PDUs (*Protocol Data Unit*), para então serem multiplexados e passados à camada de transmissão do HD Radio.

A Figura 10 contém uma representação em alto nível dos serviços do HD Radio.

Os serviços MPS são subdivididos em MPSA e MPSD, e o SPS em SPSA e SPSD, indicando os componentes de áudio e de dados desses serviços. A componente de áudio desses serviços representa a transmissão do áudio em si, que pode ser do áudio principal (MPSA) ou de áudio secundários (SPSA). A

---

<sup>8</sup> Iiquity licensing program: <http://ibiquity.com/broadcasters/licensing>

componente de dados desses serviços carrega informações como título e álbum da música, nome do artista e comentários.

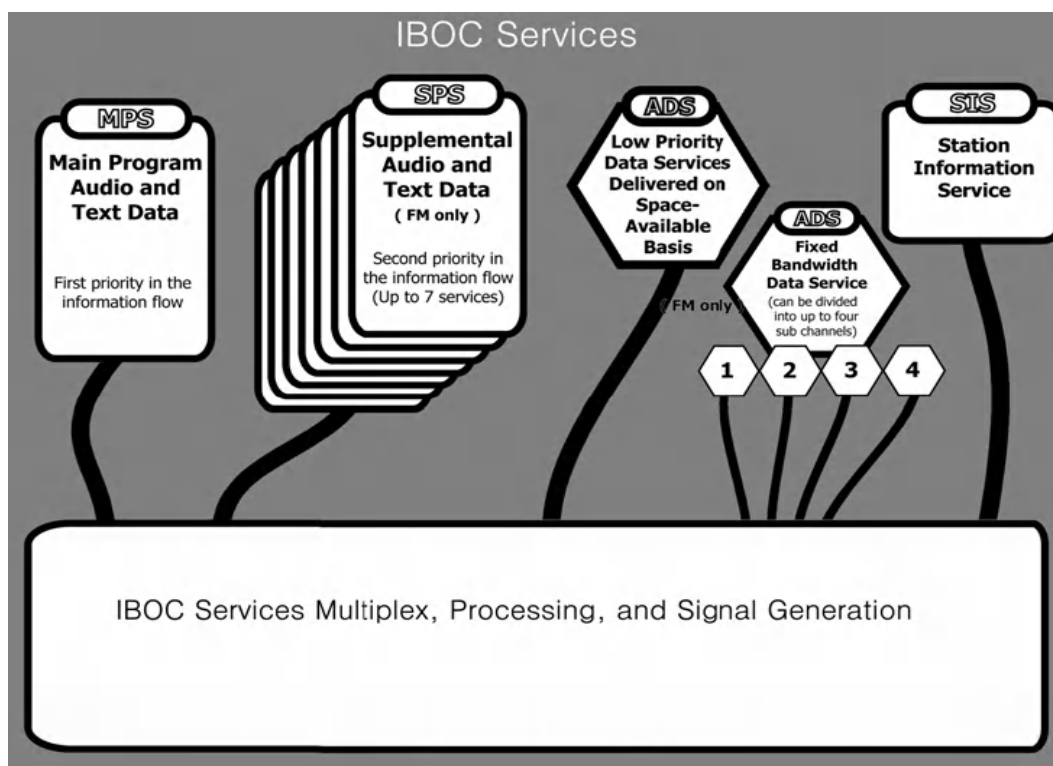


Figura 10. Serviços do HD Radio.

O serviço SIS transporta informações da emissora como nome, indicativo, localização e horário local.

O serviço ADS permite a transmissão de dados, que podem ou não estarem relacionados com os programas. Os dados de serviços ADS são transmitidos utilizando o protocolo *Advanced Application Services Transport* (AAT) (NRSC, 2011b). O AAT será o protocolo utilizado para o transporte dos componentes de uma aplicação NCL. A Figura 11 apresenta uma visão mais detalhada do HD Radio, onde pode-se identificar o AAT.

O protocolo AAT é implementado somente para a variante do HD Radio para FM, não sendo suportado pela variante AM do sistema (MAXSON, 2007), no entanto não existe qualquer limitação técnica para sua implementação na variante AM do HD Radio.

Os dados transmitidos pelo protocolo AAT são transportados por pacotes de nome *Data Transport Packets*, que são de tamanho variável, com tamanho entre 1 a 8192 de bytes úteis, dependendo da alocação do *bitrate* do serviço de dados pela emissora. Cada *Data Transport Packet* é encapsulado em um PDU.

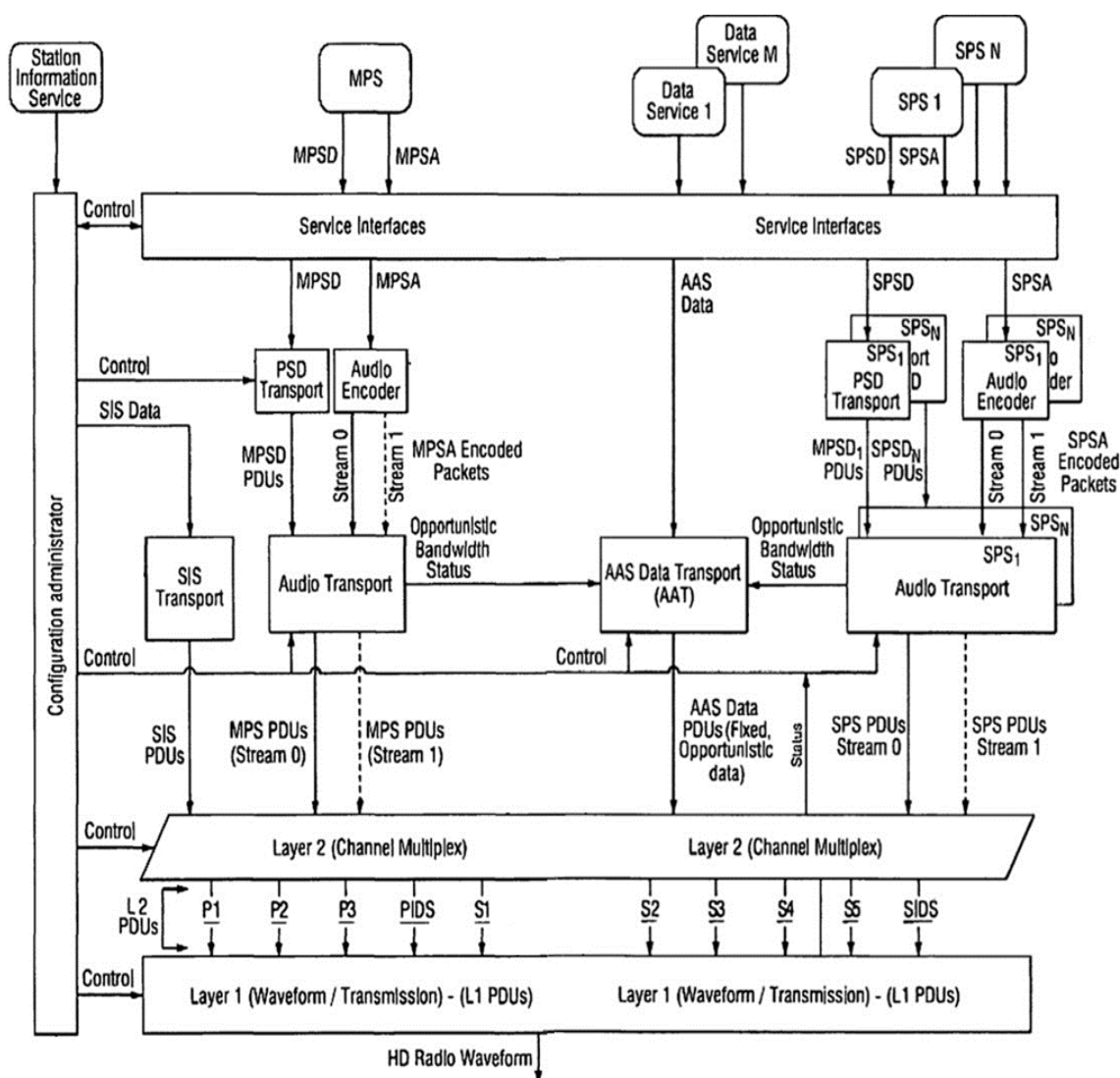


Figura 11. Diagrama de blocos do sistema HD Radio.

Os pacotes transmitidos pelo protocolo AAT possuem um identificador de 16 bits de nome *Port Number*. Essa identificação também é utilizada pelos outros serviços transmitidos no multiplex. A Tabela 11 apresenta a atribuição de portas.

Número da Porta	Status / Dados / Aplicação
0x0000 to 0x0400	Reservado para Uso do Sistema
0x0401 to 0x50FF	Livre para uso por Aplicações Atribuído pelo Administrador da Estação O Administrador da Estação pode selecionar portas com número até 0x50FF

0x5100	PSD para MPS Audio
0x5101 to 0x5200	Reservado para Uso do Sistema
0x5201 to 0x5207	PSD para SPS1 Audio (HD-2) até SPS7 Audio (HD-8)
0x5208 to 0x52FF	Reservado para Uso do Sistema
0x5300 to 0x7CFF	Reservado para Futuras Aplicações
0x7D00 to 0x7EFF	Inválido – Não deve ser utilizado
0x7F00 to 0xFEFF	Reservado para Futuras Aplicações
0xFF00 to 0xFFFF	Reservado para Uso do Sistema

Tabela 11. Atribuição do número das portas do HD Radio.

Para transmissão de aplicações NCL pelo HD Radio, até que seja adotada uma atribuição por norma, o número de porta utilizado será 0x50FF.

Pelo fato de não existir uma especificação aberta de um protocolo para transmissão de arquivos em carrossel que utilize o AAT, será utilizado o protocolo MOT, assim como definido na seção 4.2 deste texto, cujos elementos são transportados via pacotes AAT.

O protocolo MOT possui como unidade de dados elementar o *MSC data group*. A forma como as entidades *MOT Directory* e *MOT Body* são segmentadas no *MSC data group* é mostrada na Figura 12.

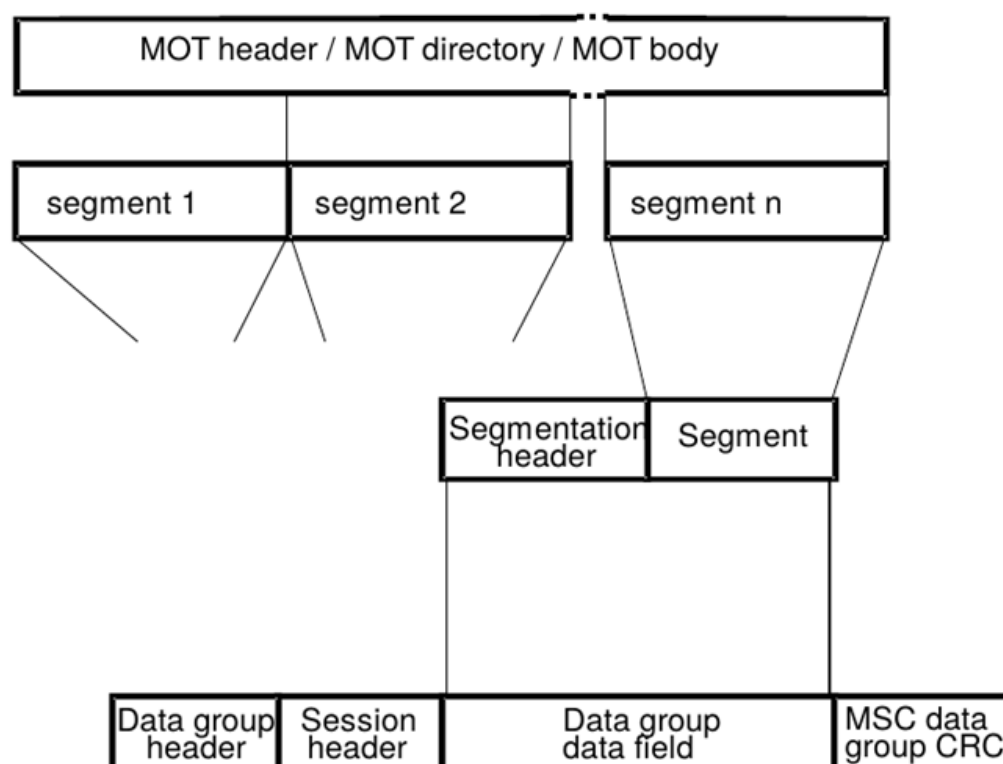


Figura 12. Etapas de segmentação dos objetos MOT em *MSC data groups*.

É interessante que o mapeamento do *MSC data group* nos *Data Transport Packets* do AAT não seja direta, e exista uma camada de adaptação entre os dois protocolos de forma que seja possível ajustar o tamanho dos pacotes AAT de forma independente do tamanho dos *MSC data group*, assim como é feito no sistema DRM através do *DRM Data Unit*. Outro exemplo de definição de um serviço de dados para o HD Radio, no caso, o Journaline, também sugere a criação de uma camada de adaptação (GORSACK, 2010).

Cada *MSC data group* deve então ser segmentado em pacotes que terão o tamanho exato do número de bytes úteis (incluso o cabeçalho) do *Data Transport Packets* do AAT, assim como apresentado pela Figura 13.

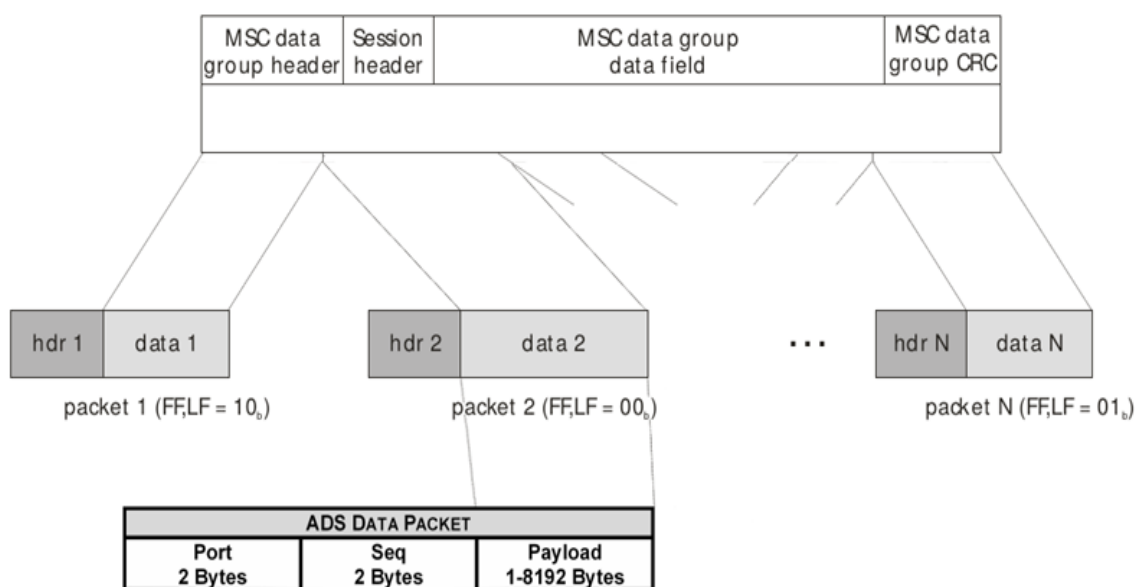


Figura 13. Segmentação do *MSC data group* do MOT em *Data Transport Packets* (identificados como *ADS Data Packet* na figura) do protocolo AAT do HD Radio.

O campo *hdr* dos pacotes da camada de adaptação é um cabeçalho de 8 bits que tem a sintaxe apresentada pela Tabela 12.

Sintaxe do cabeçalho ( <i>hdr</i> ) do pacote da camada de adaptação	
Campo	Tamanho
First Flag (FF)	1 bit
Last Flag (LF)	1 bit
Packet Id	2 bits
Padded Packet Indicator (PPI)	1 bit
Continuity Index (CI)	3 bits

Tabela 12. Campos do cabeçalho dos pacotes da camada de adaptação entre os protocolos MOT e AAT.

O campo *First Flag* (FF) indica o início do conteúdo da unidade de dados, no caso, o *MSC data group* do MOT, *Last Flag* (LF) indica o final do conteúdo. Pacotes intermediários devem receber 0 para ambos FF e LF, e para o caso do pacote conter uma unidade de dados completa, ambos os campos FF e LF devem conter o bit 1.

*Packet Id* deve receber o valor 0, ficando os outros valores reservados para futuras extensões. *Padded Packet Indicator* (PPI) indica, quando contém o valor 1, que o pacote pode conter *bits* de preenchimento (*padding bits*), e que os dois primeiros *bytes* do pacote indicam o tamanho em *bytes* do conteúdo útil do pacote. Finalmente, o campo *Continuity Index* (CI) deve ser incrementado de 1 módulo 8 a cada pacote gerado para dado *Packet Id*.

Caso o último pacote do MOT tenha o tamanho total do conteúdo útil do *Data Transport Packet*, menos 1 (tamanho do cabeçalho *hrd*), dois pacotes devem ser gerados com o PPI ajustado para 1, o penúltimo com ambos FF e LF setadas para 0, e o último com a FF 0 e LF 1, e carregando somente um *byte* útil.

Outra consideração a ser feita está relacionada a geração dos PDUs a partir dos pacotes do protocolo AAT com respeito à substituição dos caracteres reservados 0x7D e 0x7E, que tem função específica do protocolo HLDC (*High-Level Data Link Control*), seguindo-se as regras especificadas pela norma do HD Radio.

Utilizando as definições desta seção é possível utilizar as mesmas definições para a transmissão de uma aplicação Ginga através do protocolo MOT descritas para o DRM no sistema HD Radio.

Elementos presentes no multiplex do HD Radio podem ser referenciados em um documento NCL através da URI apresentada na Tabela 13, assim como definido em norma (ETSI, 2015c), adicionada de uma extensão.

Esquema	Parte específica do esquema	Interpretação
hd:	<cc>.<tx>[.<portnumber>]	cc Código do país (3-char). tx é indicativo da emissora, também conhecido como <i>Service Broadcast identifier</i> (5-char). portnumber é o número da

		porta do serviço HD Radio, é um parâmetro opcional.
--	--	---

Tabela 13. URI para referência a elementos do multiplex do HD Radio.

Além dos campos da URI definidos em norma, *cc* e *tx*, o campo *portnumber* foi adicionado de forma que os diferentes componentes de dados e áudio do multiplex do HD Radio possam ser referenciados de forma unívoca.

Uma consideração a ser feita é que existe um protocolo para transmissão de arquivos para o HD Radio conhecido como LOT (*Large Object Transfer*), no entanto, o LOT é um protocolo proprietário, não existindo norma ou documento que o defina, apenas menções no site da Ibiquity<sup>9</sup>.

<sup>9</sup> Site da Ibiquity. HD Radio data services:  
<[http://www.ibiquity.com/hd\\_radio/what\\_are\\_data\\_services](http://www.ibiquity.com/hd_radio/what_are_data_services)>. Acesso em 10/06/2015.



## 5

### O Perfil NCL Digital Radio

Este capítulo tem por objetivo a definição do perfil *Digital Radio* da NCL. São apresentados os elementos da linguagem NCL 3.1 e as escolhas a respeito da presença ou não de cada elemento no perfil DR, baseadas principalmente nos requisitos levantados no Capítulo 3 e também em análises sobre a relevância da presença de determinados elementos em aplicações NCL para o rádio digital.

A NCL 3.1 e seu middleware Ginga correspondente apresentam pequenos aperfeiçoamentos com relação a NCL 3.0, como novas classes de dispositivos para ambiente multi-dispositivos, e algumas mudanças, por exemplo, alguns elementos passaram para o status *deprecated*. Para o rádio digital não existe a necessidade de compatibilidade do perfil DR com alguma versão antiga da linguagem, pois será a primeira versão do perfil, portanto, conforme será discutido, o novo perfil se baseia inteiramente na NCL 3.1, sem qualquer compatibilidade com elementos marcados como *deprecated*.

A análise da linguagem NCL feita por Lima (LIMA, 2010) para a concepção do perfil *RawDTV* é utilizada para a definição de alguns aspectos do perfil Digital Radio, pois permite identificar “açúcares sintáticos” e recursos que propiciam mecanismos para reúso de código, que muitas vezes não tem importância no contexto da radiodifusão digital, onde a aplicação é normalmente enviada de forma completa pelo carrossel, não havendo possibilidade de reúso entre cada resintonização do receptor.

Parte da complexidade da apresentação definida por um documento NCL se dá devido ao distanciamento semântico entre um documento declarativo NCL e as interfaces de baixo nível do sistema operacional, que executam a apresentação sincronizada de mídias.

Tipicamente uma máquina de execução NCL converte o documento NCL em uma representação interna, para depois um *player* executar essa estrutura de dados da apresentação, assim como mostrado na Figura 14.

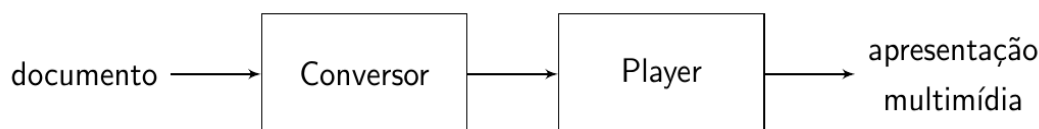


Figura 14. Arquitetura típica de uma máquina de apresentação NCL.

O perfil *Digital Radio* da linguagem NCL, com a retirada de alguns “açúcares sintáticos” da linguagem permite uma diminuição desse distanciamento entre a linguagem e sua representação interna. Uma possível consequência é a redução da complexidade do *middleware*, fato que normalmente vem acompanhado da redução na possibilidade de *bugs*.

É importante ressaltar que a maioria dos dispositivos que recebem rádio digital e tem a capacidade de apresentação de mídias possuem algum tipo de aceleração em hardware para a decodificação de imagens e sons<sup>10</sup>, portanto, grande parte do poder de processamento do Processador de Aplicação (*Application Processor*) é gasto pelo *middleware* no *parsing* do documento NCL, na conversão para o modelo de apresentação e no escalonador que faz a orquestração da apresentação hipermídia.

A Listagem 2 apresenta todos os elementos da NCL na organização adotada pelo *NCL Handbook* (SOARES, 2013).

```

Elementos básicos
Estrutura e Conteúdo:
    <ncl>
    <head>
    <body>
    <context>
    <media>
Interfaces:
    <area>
    <property>
    <port>
Ligação:
    <link>
    <linkParam>
    <bind>
    <bindParam>
Conectores:
    <connectorBase>
    <causalConnector>
    <compoundCondition>
  
```

<sup>10</sup> Um exemplo de API multiplataforma largamente utilizada em dispositivos embarcados para decodificação de mídias com assistência via hardware é a OpenMAX: <https://www.khronos.org/openmax/>

```

    <simpleCondition>
    <compoundAction>
    <simpleAction>
    <compoundStatement>
    <assessmentStatement>
    <attributeAssessment>
    <valueAssessment>
    <connectorParam>
Meta-data:
    <meta>
    <metadata>
Elementos Complementares
Switches e Regras:
    <switch>
    <switchPort>
    <mapping>
    <defaultComponent>
    <descriptorSwitch>
    <defaultDescriptor>
    <bindRule>
    <ruleBase>
    <compositeRule>
    <rule>
Aparência e Leiaute:
    <descriptorBase>
    <descriptor>
    <descriptorParam>
    <regionBase>
    <region>
Efeitos de Transição:
    <transitionBase>
    <transition>
Importação:
    <importBase>
    <importedDocumentBase>
    <importNCL>

```

Listagem 2. Todos os elementos da linguagem NCL 3.1.

A maioria dos elementos possuem o atributo *id*, que os identifica de forma única. Para evitar repetições, o atributo *id* não será discutido de forma recorrente na análise de cada elemento. Além dos elementos da linguagem e seus atributos, as variáveis globais, que são definidas como propriedades do elemento *<media>* do tipo “*application/x-ncl-settings*“, também são discutidas neste capítulo.

O conteúdo do capítulo está organizado na seguinte ordem: elementos de aparência e leiaute, elementos de efeitos de transição, elementos para importação de entidades externas, elementos de estrutura e conteúdo, elementos de ligação, elementos conectores, *Switches* e regras, elementos de interface, variáveis globais e elementos de meta-dados. Ao final é apresentado um resumo sobre as definições para novo perfil.

## 5.1.Elementos de aparência e leiaute

Os 5 elementos considerados para análise nesta seção são apresentados na Listagem 3:

```
<descriptorBase>
<descriptor>
<descriptorParam>
<regionBase>
<region>
```

Listagem 3. Elementos de aparência e leiaute.

Os elementos *<region>* e *<regionBase>* são utilizados para definir valores iniciais de localização dos objetos. A região retangular na tela que é utilizada para a apresentação de mídias é definida pelos atributos *left*, *right*, *top*, *bottom*, *height* e *width*, e a precedência das regiões entre si, indicando qual região é apresentada na frente e qual fica atrás, é feita através do atributo *zIndex*. Também é possível indicar o dispositivo no qual a região está, por exemplo, a tela de um segundo dispositivo, como um *smartphone*, através do atributo *device* do elemento *<regionBase>*.

Os elementos *<descriptorBase>*, *<descriptor>* e *<descriptorParam>* tem a função de definir algumas propriedades iniciais de mídia, por exemplo, a duração de apresentação da mídia e o destino do foco no caso de uma tecla direcional ser pressionada.

Como todas as informações especificadas por esses elementos também podem ser especificadas pelo elemento *<property>*, filho do um elemento *<media>*, esses elementos podem ser considerados “açúcares sintáticos”.

Um importante propósito desses elementos é permitir o reuso de regiões e descritores, no entanto, pelo fato de aplicações NCL para o rádio comportarem um pequeno número de mídias, devido ao baixo bitrate de transporte, a importância do reuso é minimizada.

Assim sendo, os elementos *<region>*, *<regionBase>*, *<descriptorBase>*, *<descriptor>* e *<descriptorParam>* não compõem o perfil DR. O código da Listagem 4 faz o uso dos elementos *<region>*, *<regionBase>*, *<descriptor>* e *<descriptorBase>*.

```

<ncl id="main">
  <head>
    <regionBase device="class(2)">
      <region id="mediaReg" width="100%" height="100%"
zIndex="1"/>
    </regionBase>
    <descriptorBase>
      <descriptor id="mediaDesc" region="mediaReg"
explicitDur="5s"/>
    </descriptorBase>
  </head>
  <body>
    <port id="entry" component="media"/>
    <media id="media" src="media.png" descriptor="mediaDesc"/>
  </body>
</ncl>

```

Listagem 4. Documento NCL que contém elementos *<region>*, *<regionBase>*, *<descriptor>* e *<descriptorBase>*.

O documento da Listagem 5 tem a mesma semântica do documento da Listagem 4 e faz o uso somente do elemento *<property>* para definir as mesmas características de apresentação:

```

<ncl id="main">
  <body>
    <port id="entry" component="media"/>
    <media id="media" src="media.png">
      <property name="device" value="class(2)">
      <property name="width" value="100%"/>
      <property name="height" value="100%"/>
      <property name="zIndex" value="1"/>
      <property name="explicitDur" value="5s"/>
    </media>
  </body>
</ncl>

```

Listagem 5. Documento NCL que faz o uso do elemento *<property>*.

Vale ponderar que para os casos em que for possível a utilização de um número alto de mídias em uma aplicação NCL DR, as facilidades de reuso que esses elementos proporcionam ao programador podem ser importantes. Para esses casos o desenvolvedor poderá lançar mão de um conversor que irá transformar as informações dos elementos de região e descritores em propriedades de mídia.

Pelo fato do perfil DR, nesse caso, ter as mesmas características do perfil RawDTV, os mesmos detalhes de conversão do perfil EDTV para RawDTV (LIMA, 2013) poderão ser utilizados na conversão para o perfil DR, com exceção das propriedades de transição, *transIn* e *transOut*, que não estão presentes no

perfil DR, conforme discutido na seção sobre os elementos de transição. Essa conversão pode ser feita com o software *dietncl*<sup>11</sup>.

Resumindo, o status de presença ou ausência dos elementos *<region>*, *<regionBase>*, *<descriptor>*, *<descriptorBase>* e *<descriptorParam>* nos dois perfis da NCL 3.1 e no perfil DR é exposto na Tabela 14.

Perfil	Presença
EDTV	SIM
RawDTV	NÃO
DR	NÃO

Tabela 14. Presença dos elementos *<region>*, *<regionBase>*, *<descriptor>*, *<descriptorBase>* e *<descriptorParam>* nos perfis da NCL 3.1 e perfil DR.

## 5.2.Elementos de efeitos de transição

Os 2 elementos considerados para análise neste subcapítulo são apresentados na Listagem 6.

```
<transitionBase>
<transition>
```

Listagem 6. Elementos de efeitos de transição.

Os elementos de efeitos de transição permitem a definição de um padrão de transição para mídias de áudio e mídias visuais. É possível ajustar o tempo de duração, tipo e subtipo da transição e outros parâmetros.

Pelo fato da taxa de transmissão no rádio digital ser baixa, poucos elementos visuais poderão existir em uma aplicação, portanto a importância da presença de efeitos de transição em imagens é pequena. No caso da transição de áudio, a única suportada pela NCL é o *fade* que, caso necessária, pode ser feita no pré-processamento do áudio do aplicativo antes da transmissão ou ainda através da manipulação da propriedade *soundLevel*. Portanto, por não ser um recurso relevante no rádio digital, os efeitos de transição não estarão no perfil DR.

A Tabela 15 compara a presença de efeitos de transição no perfil DR.

Perfil	Presença
EDTV	SIM
RawDTV	SIM

<sup>11</sup> DietNCL, uma ferramenta para remover açúcar sintático de documentos NCL: <https://github.com/gflima/dietncl>

DR	NÃO
----	-----

Tabela 15. Presença de efeitos de transição nos perfis da NCL 3.1 e perfil DR.

### 5.3.Elementos para importação de entidades externas

A Listagem 7 contém os 3 elementos considerados para análise neste subcapítulo são:

```
<importBase>
<importedDocumentBase>
<importNCL>
```

Listagem 7. Elementos de importação.

Esses elementos tem o propósito de importar entidades externas ao documento NCL sendo apresentado, a partir de outros documentos NCL.

O elemento *<importBase>* tem por finalidade incorporar de outro documento NCL as bases de elementos especificados no *<connectorBase>*, *<regionBase>*, *<descriptorBase>*, *<ruleBase>* e *<transitionBase>*, podendo ser especificada uma região pai, no caso de importação de regiões, através do atributo *region*. Outros dois atributos do elemento *<importBase>*, *documentURI* e *alias*, definem o caminho do documento a ser importado e um prefixo para ser utilizado na referência aos elementos importados, respectivamente.

Os elementos *<importedDocumentBase>* e *<importNCL>* são utilizados para importação de documentos NCL. Os atributos do elemento *<importNCL>* são o *documentURI*, que especifica o caminho do documento NCL, e *alias*, que especifica um rótulo para ser utilizado como prefixo nas referências às entidades importadas.

Um dos propósitos dos elementos de importação é o reuso de código, fator que no contexto do rádio digital não é muito relevante pois as aplicações tendem a ser pequenas e o reuso não acontece quando o usuário troca de canal.

Para o caso da autoria, esses elementos poder ser úteis na organização do código. Caso o desenvolvedor julgue os elementos de importação necessários, a ferramenta *dietncl* faz a conversão automática entre um documento que utiliza os elementos de importação em outro semanticamente equivalente, que não contém tais elementos.

A remoção dos elementos de importação não diminui a expressividade da linguagem (LIMA, 2011), sendo que o conversor opera da seguinte maneira:

“Os elementos das bases [do documento importado] são importados para as bases correspondentes [no documento de origem], e o <body> importado, bem como quaisquer de seus nós, pode ser reusado dentro do <body> do documento NCL que realizou a importação.”

Com os argumentos expostos e em consonância dos requisitos levantados para o rádio digital, os elementos de importação não são definidos para o perfil DR.

A Tabela 16 apresenta a presença ou ausência dos elementos de importação nos perfis EDTV, RawDTV e DR.

Perfil	Presença
EDTV	SIM
RawDTV	NÃO
DR	NÃO

Tabela 16. Presença dos elementos de importação nos perfis da NCL 3.1 e perfil DR.

#### 5.4.Elementos de Estrutura e Conteúdo

Nesta seção são parte da análise os seguintes 5 elementos presentes na Listagem 8.

```
<ncl>
<head>
<body>
<context>
<media>
```

Listagem 8. Elementos de Estrutura e Conteúdo.

Os elementos <ncl>, <head> e <body> são elementos de estruturação do documento. O elemento <ncl> representa a raiz do documento e contém todos os elementos filhos do documento. O elemento <ncl> possui um atributo *xmlns* que informa o esquema XML do documento, onde é definida a versão e perfil da NCL adotada. Já no elemento <body> é onde todos os objetos de mídia e os relacionamentos entre eles são declarados. O elemento <head> é pai de elementos para reuso, principalmente.

Esses três elementos não têm função na apresentação, mas servem para organizar o documento, facilitando sua legibilidade e identificação. Além disso, eles compõem a estrutura básica do documento NCL. Todos os perfis possuem



esses elementos básicos, e a remoção deles não traria qualquer benefício, portanto, `<ncl>`, `<head>` e `<body>` fazem parte do perfil DR.

O elemento `<context>` é um elemento de estrutura organizacional, que permite o agrupamento dos elementos NCL, objetos de mídia, elos e contextos, recursivamente. No entanto, o elemento não tem papel relevante na descrição da apresentação, apesar disso o elemento `<context>` permite a estruturação do documento e facilita a manutenção de sua consistência.

A linguagem NCL, Linguagem de Contexto Aninhado, perderia sua identidade sem a presença do elemento `<context>`. Portanto, assim como nos outros dois perfis da NCL, o elemento `<context>` está presente no perfil DR.

O elemento `<media>` define um objeto de mídia que pode ser uma imagem, texto, áudio, vídeo, scripts NCLua, aplicações HTML e aplicações NCL embutidas, dentre outros tipos. Não existe uma apresentação multimídia sem mídias, portanto, sua presença é essencial em qualquer perfil, inclusive no perfil DR.

O atributo `src` do elemento `<media>` contém um Uniform Resource Identifier (URI), que referencia um conteúdo de mídia através de uma *string*, em conformidade com algum dos esquemas suportados. No perfil DR, não são suportados os esquemas específicos para TV Digital *ts* e *dsm-cc*, mas esquemas específicos para rádio digital providos pelos sistemas Digital Radio Mondiale e HD Radio deverão ser suportados de acordo com o sistema de rádio digital utilizado. Os esquemas específicos de rádio digital são apresentados no Capítulo 4, que versa sobre o transporte de aplicações.

O esquema especial tipo *ncl-mirror* indica que a mídia que o utiliza no atributo `src` compartilha o mesmo conteúdo instantâneo da mídia referenciada, mas permitindo que a mídia referenciada e a que referencia tenham propriedades distintas, sendo útil para TV 3D. O esquema *file* faz referência a um arquivo local e os esquemas *http* e *https* referenciam arquivos remotos acessíveis através dos protocolos HTTP e HTTPS respectivamente. Os esquemas *file*, *http*, *https* são suportados no perfil DR.

Outro atributo existente é o *refer*, que permite a um objeto especificado por `<media>`, `<context>` ou `<switch>` ser reusado através da atribuição do *id* do objeto ao atributo *refer*. Todos os elementos filhos do objeto referido são herdados pelo elemento que faz referência. Para o caso de uso do *refer* em objetos

<media>, um outro atributo é permitido, o *instance*, que pode receber os valores *instSame*, que indica que o objeto referido e o objeto que faz a referência formam uma mesma apresentação, ou *new*, que indica que o objeto que faz referência é um objeto independente do objeto referenciado. Ambos atributos são suportados no perfil DR.

O atributo *descriptor* do elemento <media> é utilizado para indicar o descritor associado à mídia, no entanto, como os descritores não são suportados no perfil DR, o atributo *descriptor* não faz parte do perfil DR da NCL.

Algumas mídias especiais também são definidas na NCL, como a mídia de tipo *application/x-ncl-settings* e *application/x-ncl-time*, que definem variáveis globais e uma marcação para o tempo, respectivamente. Algumas mudanças nas variáveis globais são abordadas na Seção 5.9.

A Tabela 17 compara a existência dos elementos de estrutura e conteúdo nos diferentes perfis.

Perfil	Presença
EDTV	SIM
RawDTV	SIM
DR	SIM

Tabela 17. Existência dos elementos de estrutura e conteúdo nos diferentes perfis.

A Tabela 18 apresenta os elementos de estrutura e conteúdo, seus atributos e elementos filhos no perfil DR. Os itens riscados indicam os elementos do perfil EDTV que não estão presentes.

Elementos	Atributos	Conteúdo
<i>Ncl</i>	<i>id, xmlns</i>	<i>(head?, body?)</i>
<i>Head</i>		<i>(<del>importedDocumentBase?</del>, ruleBase?, transitionBase?, <del>regionBase*</del>, <del>descriptorBase?</del>, connectorBase?, meta*, metadata*)</i>
<i>Body</i>	<i>Id</i>	<i>(port   property  media   context   switch  link   meta   metadata)*</i>
<i>context</i>	<i>id, refer</i>	<i>(port   property   media   context   link   switch   meta   metadata)*</i>
<i>media</i>	<i>id, src, refer, instance,</i>	<i>(area   property)*</i>

	<i>type, <del>descriptor</del></i>	
--	------------------------------------	--

Tabela 18. Definição dos elementos de estrutura e conteúdo no perfil DR.

## 5.5.Elementos de ligação

Nesta seção são analisados 4 elementos, apresentados na Listagem 9.

```
<link>
<linkParam>
<bind>
<bindParam>
```

Listagem 9. Elementos de ligação.

Os elementos de ligação definem relacionamentos entre mídias e objetos compostos. O `<link>` tem um atributo *xconnector*, onde é indicado o identificador da relação, previamente definida em um elemento `<causalConnector>`. O elemento `<causalConnector>` somente define o tipo de relação e os papéis, ficando para o elemento `<link>` e seus filhos a tarefa de designar quem exerce os papéis.

O elemento `<bind>` é utilizado para associar uma interface de um objeto, através dos atributos *component* e *interface*, a um papel de um conector, através do atributo *role*. O atributo *descriptor* do elemento `<bind>` não é suportado no perfil DR devido à ausência desse recurso.

O elemento `<bindParam>` é filho do elemento `<bind>` e é utilizado para atribuir valor a uma interface de um objeto, através dos atributos *name* e *value*.

Já o elemento `<linkParam>` tem sintaxe semelhante ao `<bindParam>`, no entanto é filho do elemento `<link>` e tem a mesma finalidade do `<bindParam>`. O `<linkParam>` permite o reúso do valor de parâmetros, no entanto o mesmo efeito pode ser obtido apenas com o uso do `<bindParam>`. Portanto, de forma a evitar a presença de elementos redundantes sem relevante motivo, o elemento `<linkParam>` não faz parte do perfil Digital Radio da NCL.

A funcionalidade dos elos é fundamental em uma apresentação NCL, pois eles são necessários para o estabelecimento de relações de causalidade entre os diversos objetos da apresentação, inclusive a interação do usuário.

Na Listagem 10 está um fragmento de um documento NCL que exemplifica o uso dos elementos de ligação para iniciar uma mídia de áudio após 5s do início de uma mídia de vídeo.

```

<ncl id="main">
  <head>
    <connectorBase>
      <causalConnector id="onBeginStart_delay">
        ...
      </causalConnector>
    </connectorBase>
  </head>
  <body>
    <port id="entry" component="video"/>
    <media id="video" src="video.mp4">
      <property name="width" value="100%"/>
      <property name="height" value="100%"/>
    </media>
    <media id="audio" src="audio.mp4" />
    <link id="link" xconnector="onBeginStart_delay">
      <bind role="onBegin" component="video"/>
      <bind role="start" component="audio">
        <bindParam name="delay" value="5s"/>
      </bind>
    </link>
  </body>
</ncl>

```

Listagem 10. Exemplo de uso dos elementos de ligação.

Com exceção do *<linkParam>*, todos os elementos ligação estão presentes na NCL DR. A Tabela 19 compara a presença dos elementos de ligação nos diferentes perfis.

Perfil	Presença
EDTV	SIM
RawDTV	SIM
DR	SIM (com uma exceção)

Tabela 19. Presença dos elementos de ligação nos diferentes perfis.

A Tabela 20 apresenta a definição dos elementos de ligação, seus atributos e elementos filhos.

Elementos	Atributos	Conteúdo
<i>Bind</i>	<i>role, component, interface, descriptor</i>	<i>(bindParam)*</i>
<i>bindParam</i>	<i>name, value</i>	<i>Empty</i>
<i>Link</i>	<i>id, xconnector</i>	<i>(linkParam*, bind+)</i>

Tabela 20. Definição dos elementos de ligação no perfil DR.

## 5.6.Elementos conectores

Os 11 elementos considerados para análise nesta seção são apresentados na Listagem 11.

```
<connectorBase>
<causalConnector>
<connectorParam>
<simpleCondition>
<compoundCondition>
<simpleAction>
<compoundAction>
<assessmentStatement>
<attributeAssessment>
<valueAssessment>
<compoundStatement>
```

Listagem 11. Elementos conectores.

Como já mencionado, o conector é o responsável por definir uma relação, mas sem definir quais nós, como uma mídia ou uma interface dela, farão parte da relação. O elemento *<link>* é o responsável pela conexão dos nós e estabelecimento de um relacionamento.

O elemento *<connectorBase>* indica que seus elementos filhos farão parte da base de conectores da apresentação NCL. A única relação definida na NCL 3.1 é a relação de causalidade, representada pelo elemento *<causalConnector>*, que é pai de todos os outros elementos de conexão.

Um dos filhos do elemento *<causalConnector>* é *<connectorParam>*, que permite a parametrização do conector através dos atributos *name*, que é um identificador único do parâmetro, e *type*, que indica o tipo do parâmetro.

As condições de disparo do conector são definidas pelos elementos *<simpleCondition>*, *<compoundCondition>*, *<assessmentStatement>*, *<attributeAssessment>*, *<valueAssessment>* e *<compoundStatement>*.

As relações em NCL são baseadas em eventos. Eventos podem ser de três tipos: apresentação, seleção e atribuição.

O elemento *<simpleCondition>* é utilizado para definir uma condição para disparo do conector. Um de seus atributos é *role*, utilizado para definir um papel da condição. Os papéis dependem do tipo de evento e da transição na máquina de estado do evento, e podem ser indicados através de um valor de *role* pré-definido. Para evento de apresentação os valores pré-definidos são *onBegin*, *onEnd*, *onAbort*, *onPause* e *onResume*, e para evento de seleção estão pré-definidos

*onSelection*, *onBeginSelection* e *onEndSelection*, e para evento de atribuição *onBeginAttribution*, *onEndAttribution*, *onAbortAttribution*, *onPauseAttribution* e *onResumeAttribution*.

Opcionalmente ao uso do atributo *role* para identificar a condição, é possível indicar a condição através do tipo do evento, representado pelo atributo *eventType*, sendo os possíveis valores *presentation*, *selection* e *attribution*, e o tipo da transição da máquina de estados de um evento através do atributo *transition*, que pode assumir os valores *starts*, *stops*, *aborts*, *pauses* ou *resumes*. Para cada combinação dos atributos *eventType* e *attribution* existe um *role* associado. Portanto, para simplificar a linguagem em seu perfil DR através da remoção de “açúcares sintáticos”, os atributos *eventType* e *transition* do elemento *<simpleCondition>* não estão presentes no perfil DR.

É possível especificar a cardinalidade dos participantes de cada papel através dos atributos *min* e *max*. Também é permitido especificar um atraso para tornar verdadeira a condição do *<simpleCondition>* através do atributo *delay*. Outro atributo é o *qualifier*, cujos valores podem ser *or* ou *and*, e indicam que a condição é verdadeira caso alguma ou todas as condições associadas a cada papel ocorram, respectivamente. Finalmente o atributo *key* indica a tecla utilizada para a seleção, no caso de um evento de seleção.

Outro elemento que pode influenciar uma condição é o *<assessmentStatement>* e seus filhos *<attributeAssessment>* e *<valueAssessment>*, que permitem testar em qual estado um evento se encontra ou, no caso de evento de atribuição, é possível avaliar uma propriedade de um objeto. O atributo *comparator* de *<assessmentStatement>* define o tipo de comparação a ser realizada, sendo elas: igual a (*eq*), diferente de (*ne*), maior que (*gt*), menor que (*lt*), maior ou igual a (*gte*) e menor ou igual a (*lte*).

O elemento *<attributeAssessment>* possui os atributos *role*, *eventType*, *key*, *attributeType* e *offset*. O atributo *role*, que deve ser um identificador único, define um ponto de conexão do conector para associação a uma interface de um objeto através de um *<link>*. O atributo *eventType* identifica o tipo do evento e o atributo *key* a tecla que foi selecionada, para o caso de evento de seleção.

O atributo *attributeType* do elemento *<attributeAssessment>* pode receber os valores *state* ou *nodeProperty*, indicando o estado do evento ou uma propriedade do objeto, respectivamente, sendo que o tipo *nodeProperty* só pode

ser utilizado em eventos de atribuição. Finalmente, o atributo *offset* representa um valor que é somado a variável utilizada na avaliação da condição.

O elemento *<compoundStatement>* permite a criação de uma regra composta por um ou mais *<assessmentStatement>* e *<compoundStatement>*, e possui os atributos *operator*, que define a operação lógica da composição, *and* ou *or*, e *isNegated*, que nega logicamente a avaliação da regra composta.

Os elementos *<simpleCondition>*, *<assessmentStatement>* e *<compoundCondition>* podem fazer parte de uma condição composta definida por *<compoundCondition>*. O atributo *operator* do *<compoundCondition>* define o operador lógico a ser aplicado na avaliação das condições, podendo assumir os valores *and* ou *or*. Outro atributo é o *delay*, que indica que caso as condições representadas pelos elementos filhos de *<compoundCondition>* sejam verdadeiras, após o tempo indicado por *delay*, a condição composta se torna verdadeira.

Quando as condições de um conector são satisfeitas, as ações definidas por elementos *<simpleAction>* são disparadas. É possível definir papéis para ações associadas a eventos de apresentação e atribuição. Os papéis pré-definidos para eventos de apresentação possíveis para o atributo *role* são: *start*, *stop*, *abort*, *pause* e *resume*; e para eventos de atribuição são: *set*, *startAttribution*, *stopAttribution*, *abortAttribution*, *pauseAttribution* e *resumeAttribution*.

Como alternativa ao uso de um dos papéis pré-definidos para o atributo *role*, é possível indicar o tipo de ação através de *eventType* e *actionType*. O atributo *eventType* pode ser *presentation* ou *attribution*, e o atributo *actionType* pode ser *start*, *stop*, *abort*, *pause* e *resume*. Todas as possíveis ações que os atributos *eventType* e *actionType* definem podem ser expressas somente com o uso do atributo *role*, portanto, de forma semelhante ao elemento *<simpleCondition>*, os atributos *eventType* e *actionType* do elemento *<simpleAction>* não integram o perfil DR da NCL.

Outros atributos do *<simpleAction>* são *delay*, que indica uma espera antes do disparo das ações, *value*, que define o valor que será atribuído à propriedade associada à ação e *min* e *max* que definem a cardinalidade das conexões permitidas pelo conector.

Para o caso de eventos de atribuição outros dois atributos podem ser utilizados no *<simpleAction>*, *duration* e *by*, que possibilitam que uma atribuição

seja realizada por um período explícito finito de tempo, indicado em *duration*, além de ser possível indicar a cadência com que a mudança de valor é realizada através do atributo *by*, que pode assumir um valor numérico ou *indefinite*, indicando que a cadência da transição de valor da propriedade deve ser a menor possível, idealmente linear. Esses dois atributos permitem a definição de animações de forma declarativa em NCL.

Elementos `<simpleAction>` podem ser disparados em conjunto, um após o outro na ordem em que foram definidos, através do elemento `<compoundAction>`. O atributo *operator* do `<compoundAction>` tem o status *deprecated* na NCL 3.1, e não é suportado no perfil DR pois não existe o motivo de se manter compatibilidade com a NCL 3.0. O conector utilizado da Listagem 10, que exemplifica o uso de elementos de ligação, é apresentado na Listagem 12.

```
<ncl id="main">
  <head>
    <connectorBase>
      <causalConnector id="onBeginStart_delay">
        <connectorParam name="delay"/>
        <simpleCondition role="onBegin"/>
        <simpleAction role="start" delay="$delay"
max="unbounded"/>
      </causalConnector>
    </connectorBase>
  </head>
  ...
</ncl>
```

Listagem 12. Código contendo um exemplo da definição de um conector.

Os conectores são uma funcionalidade essencial para a criação de elos entre os objetos da apresentação e são, portanto, uma funcionalidade essencial da NCL, estando presente em todos os perfis.

Os elementos conectores presentes no perfil EDTV e não presentes no perfil NCL DR são o atributo *operator* do elemento `<compoundAction>`, os atributos *eventType* e *transition* do elemento `<simpleCondition>` e os atributos *eventType* e *actionType* do elemento `<simpleAction>`, considerados açúcares sintáticos. A Tabela 21 compara a presença dos conectores nos diferentes perfis.

Perfil	Presença
EDTV	SIM
RawDTV	SIM
DR	SIM (com exceções)



Tabela 21. Presença dos conectores nos diferentes perfis.

A Tabela 22 apresenta os elementos conectores, seus filhos e atributos.

Elementos	Atributos	Conteúdo
<i>connectorBase</i>	<i>Id</i>	<i>causalConnector*</i>
<i>causalConnector</i>	<i>Id</i>	( <i>connectorParam*</i> , ( <i>simpleCondition</i>   <i>compoundCondition</i> ), ( <i>simpleAction</i>   <i>compoundAction</i> ))
<i>connectorParam</i>	<i>name, type</i>	<i>Empty</i>
<i>simpleCondition</i>	<i>role, delay, eventType,</i> <i>key, <del>transition</del>, min,</i> <i>max, qualifier</i>	<i>Empty</i>
<i>compoundCondition</i>	<i>operator, delay</i>	(( <i>simpleCondition</i>   <i>compoundCondition</i> ) +, ( <i>assessmentStatement</i>   <i>compoundStatement</i> )*)
<i>simpleAction</i>	<i>role, delay, eventType,</i> <i><del>actionType</del>, value, min,</i> <i>max, duration, by</i>	<i>Empty</i>
<i>compoundAction</i>	<i>delay, <del>operator</del></i>	( <i>simpleAction</i>   <i>compoundAction</i> ) +
<i>assessmentStatement</i>	<i>Comparator</i>	( <i>attributeAssessment</i> , ( <i>attributeAssessment</i>   <i>valueAssessment</i> ))
<i>attributeAssessment</i>	<i>role, eventType, key,</i> <i>attributeType, offset</i>	<i>Empty</i>
<i>valueAssessment</i>	<i>Value</i>	<i>Empty</i>
<i>compoundStatement</i>	<i>operator, isNegated</i>	( <i>assessmentStatement</i>   <i>compoundStatement</i> ) +

Tabela 22. Elementos conectores, seus atributos e elementos filhos.

## 5.7.Switches e regras

Os 10 elementos considerados para análise são apresentados na Listagem 13.

```
<switch>
<switchPort>
<mapping>
<defaultComponent>
<descriptorSwitch>
<defaultDescriptor>
<bindRule>
<ruleBase>
<compositeRule>
<rule>
```

Listagem 13. Elementos que permitem Switch e regras.

Os elementos *<descriptorSwitch>* e *<defaultDescriptor>* são elementos que permitem a seleção de um descritor dentre uma lista de descritores a partir da avaliação de uma ou mais regras. Como os descritores não são suportados no perfil NCL DR, esses dois elementos não serão discutidos pois não são suportados.

Regras podem ser definidas através dos elementos *<ruleBase>*, *<compositeRule>* e *<rule>*. O elemento *<ruleBase>* é filho do elemento *<head>* e define uma base de regras. Seus elementos filhos são *<compositeRule>* e *<rule>*. O *<compositeRule>* permite o agrupamento de regras que são avaliadas de acordo com o atributo *operator*, que pode assumir os valores *and* ou *or*, já o elemento *<rule>* é o responsável pela definição de uma regra.

Cada regra tem um atributo *var*, que deve ter o mesmo nome de alguma variável global, representada como propriedade da mídia do tipo *application/x-ncl-settings* na NCL. Outro atributo de *<rule>* é *comparator*, que pode assumir os valores *eq*, *ne*, *gt*, *lt*, *gte* e *lte*, indicando o tipo de comparação a ser realizada. Finalmente o atributo *value* indica o valor a ser utilizado na avaliação da comparação com a variável global.

O elemento *<switch>* é uma composição de objetos (*media*, *context* ou *switch*) alternativos, dentre os quais somente um é selecionado após a avaliação de regras. O atributo *refer* do elemento *<switch>* permite o reuso de um *switch* e deve conter o identificador do elemento que se quer referenciar. Os elementos *<defaultComponent>*, *<switchPort>* e *<bindRule>* são filhos do elemento *<switch>*.

O elemento *<bindRule>* permite a seleção dos objetos através dos atributos *rule*, que contém o nome da regra, e *constituent*, que contém o nome do objeto a ser selecionado caso a regra seja verdadeira. Caso nenhuma regra seja avaliada verdadeira, o objeto indicado em *<defaultComponent>* no atributo *component* é selecionado.

Outros dois elementos, *<switchPort>* e *<mapping>* são utilizados para a criação de interfaces no *<switch>*. Uma interface definida em *<switchPort>* permite que sejam selecionadas interfaces internas dos objetos do *<switch>*, mapeados através de elementos *<mapping>*. Os atributos do elemento *<mapping>* são *component* e *interface*. *Component* indica o nome do objeto, e *interface*, a interface interna do objeto que deve ser utilizada no *switch*.

Para o caso do *switch* ser acessado através de sua âncora padrão, que representa todo seu conteúdo (*wholeContentAnchor*), não existe a necessidade do uso do *<switchPort>* e *<mapping>*. Neste caso todos os objetos do *switch* são mapeados através da âncora padrão.

As regras e Switches permitem ao autor de uma aplicação NCL a seleção de mídias baseadas em valores de variáveis globais. Isso é importante pois permite a seleção de mídias apropriadas ao contexto do usuário, de acordo, por exemplo, com a localização do receptor, presença de conectividade ou língua do utilizador.

Outro aspecto é que a construção sintática de switches e regras é mais compacta que a de conectores e elos, aumentando a legibilidade do código e diminuindo o tamanho do documento NCL.

Conforme discutido por Lima et al (LIMA, 2013), as regras e Switches são elementos cuja semântica pode ser obtida através de elos e conectores especialmente construídos. No entanto, o processo de conversão proposto não é trivial.

A presença dos elementos de *switch* e regras favorece o uso do perfil DR diretamente para autoria, visto que o uso de elementos conectores e elos para funcionalidades de seleção baseada em regras não são triviais para um programador iniciante.

Os elementos de *Switch* e regras integram o perfil Digital Radio, com exceção dos elementos *<descriptorSwitch>* e *<defaultDescriptor>*, e devido ao fato do reuso não ter grande relevância no rádio digital, o atributo *refer* do elemento *<switch>* também não integra o perfil DR.

A Tabela 23 compara a presença dos elementos de Switch e regras nos diferentes perfis.

Perfil	Presença
EDTV	SIM
RawDTV	NÃO
DR	SIM (com exceções)

Tabela 23. Presença dos elementos de *Switch* e regras nos diferentes perfis.

A Tabela 24 mostra os elementos, filhos e atributos discutidos nesta seção e suportados pelo perfil DR.

Elementos	Atributos	Conteúdo
<i>Switch</i>	<i>id</i> , <del><i>refer</i></del>	<i>(defaultComponent?,(switchPort bindRule media  context  switch)*)</i>
<i>defaultComponent</i>	<i>Componente</i>	<i>Empty</i>
<i>bindRule</i>	<i>constituent</i> , <i>rule</i>	<i>Empty</i>
<i>switchPort</i>	<i>Id</i>	<i>mapping+</i>
<i>Mapping</i>	<i>component</i> , <i>interface</i>	<i>Empty</i>
<i>ruleBase</i>	<i>Id</i>	<i>(rule compositeRule)+</i>
<i>Rule</i>	<i>id</i> , <i>var</i> , <i>comparator</i> , <i>value</i>	<i>Empty</i>
<i>compositeRule</i>	<i>id</i> , <i>operator</i>	<i>(rule   compositeRule)+</i>
<del><i>descriptorSwitch</i></del>	<i>Id</i>	<del><i>(defaultDescriptor?,(bindRule descriptor)*)</i></del>
<del><i>defaultDescriptor</i></del>	<del><i>Descriptor</i></del>	<i>Empty</i>

Tabela 24. Elementos Switch e regras, seus atributos e filhos.

## 5.8.Elementos de interface

Os três elementos avaliados estão listados na Listagem 14.

<pre>&lt;area&gt; &lt;property&gt; &lt;port&gt;</pre>
---

Listagem 14. Elementos de interface.

Os elementos de interface permitem a criação de interfaces (propriedades ou âncoras) em objetos de mídia ou objetos compostos.

O elemento `<port>` permite a criação de interfaces em objetos compostos (`<body>` e `<context>`), sem as quais não é possível um elo externo ao contexto ser ligado a um objeto interno. O `<port>` possui os atributos *component*, que indica para qual objeto de mídia ou contexto a porta mapeia, e opcionalmente o atributo *interface*, que recebe o nome do ponto de interface de destino, como uma propriedade ou âncora de um objeto. Ao iniciar a execução de um documento NCL, caso nenhuma interface seja especificada, a máquina de apresentação inicia todas as portas do contexto principal representado pelo elemento `<body>`. O mesmo acontece se um elemento `<context>` é iniciado sem especificar uma porta. A Listagem 15 apresenta um exemplo de uso de `<port>`.

```
<ncl id="main">
  <body>
    <port id="main" component="menu" interface="img" />
    <context id="menu">
      <port id="img" component="img1" />
      <media id="img1" src="img.png" />
    </context>
  </body>
</ncl>
```

Listagem 15. Exemplo de uso do elemento `<port>`.

O elemento `<area>` é utilizado para definir uma âncora de conteúdo em uma mídia, que pode ser, por exemplo, um intervalo de tempo ou uma região de uma mídia.

Âncoras de tempo podem ser definidas através dos atributos *begin* e *end*, que assumem valores em segundos, adicionando-se um “s” após o valor, ou no formato “*Hora:Minuto:Segundo.Fração*”. Os atributos *first* e *last* são utilizados para definir uma âncora de início ou final baseada em determinada amostra da mídia ou valor de tempo relativo à base temporal em uso. No caso do rádio digital, a base temporal é a RTB (*Radio Time Base*), definida na seção 6.4.

A sintaxe dos atributos *first* e *last* para o caso de uso de valores relativos a uma base de tempo será um número inteiro com o sufixo “tbv” (*time base value*). Por exemplo, “35000tbv”, indica que quando o valor da base atingir 35000, uma transição de evento acontecerá na interface especificado no elemento `<area>`.

O atributo *coords* permite a definição de uma âncora espacial para mídias visuais. Os valores no formato “X1, Y1, X2, Y2, X3, X3, ...” identificam as coordenadas em pixels do polígono que determina a âncora.

Para mídias de texto os atributos *beginText*, *beginPosition*, *endText* e *endPosition* permitem a seleção um trecho do texto. Os atributos *beginText* e *endText* definem as cadeias de caracteres delimitadora do texto e *beginPosition* e *endPosition* definem a posição de cada cadeia de caracteres no texto.

Outros dois atributos fazem parte do elemento *<area>*: *label* e *clip*. O atributo *label* estabelece uma interface para uma âncora de conteúdo interna de um objeto, tal que o exibidor para o tipo de objeto saberá identificar. Por exemplo, para objetos de mídia do tipo “*application/x-ginga-NCLua*” o atributo *label* corresponde a uma interface que pode ser receber ações como *start* e *stop* disparada via script Lua. O atributo *clip* identifica o trecho de uma cadeia temporal de um objeto hipermídia declarativo, e apresenta o formato “(*chainId*, *beginOffset*, *endOffset*)”, onde *chainId* identifica uma das cadeias do objeto declarativo hipermídia, e *beginOffset* e *endOffset* definem o tempo de início e final da âncora assim como os atributos *begin* e *end*.

O atributo *clip* é raramente utilizado e nenhum *player* citado no texto suporta tal âncora. O atributo *coords*, que permite a definição de âncoras em objetos visuais, tem pouca relevância no contexto de aplicações de rádio digital, no qual as poucas mídias possivelmente serão apresentadas por completo. Além disso, a funcionalidade do atributo *coords* é pequena, devido a impossibilidade da definição de um índice para navegação através do *focusIndex*, de uso exclusivo da mídia como um todo.

O perfil DR, portanto, não contem os atributos *clip* e *coords*, o que implica que os *players* de elementos visuais do Ginga terão a complexidade levemente reduzida. Todos os outros atributos estão presentes no perfil DR.

O elemento *<property>* tem o propósito de definir propriedades de um objeto de mídia, contexto ou *switch*. Três atributos estão definidos, *name*, *value* e *externable*. O atributo *name* identifica o nome da propriedade, e o atributo *value* define o valor inicial de uma propriedade. Não é permitido mais de uma propriedade com mesmo nome por cada elemento pai (*<media>*, *<context>* ou *<switch>*). O atributo *externable* deve receber um valor booleano que indica se a

propriedade poderá ou não ser utilizada em um relacionamento (elemento *<link>*).

Algumas propriedades são pré-definidas, dependendo do tipo de mídia. Por exemplo, para mídias visuais, propriedades relativas a apresentação da mídia tais como *top*, *left*, *bottom*, *right*, *width* e *height* são inerentes à apresentação da mídia e definidas de forma implícita, caso não declaradas explicitamente. Outro exemplo é a propriedade implícita *timeBaseId*, que foi definida por Moreno e Soares (MORENO, 2014), e possui inicialmente um valor vazio, mas para o caso de uma mídia que é um fluxo de áudio nativo do sistema de rádio digital, essa propriedade recebe o identificador da base temporal que referencia o fluxo, quando ela existe.

As propriedades reservadas das mídias são: *top*, *left*, *bottom*, *right*, *width*, *height*, *location*, *size*, *bounds*, *baseDeviceRegion*, *device*, *explicitDur*, *background*, *transparency*, *rgbChromakey*, *visible*, *fit*, *scroll*, *style*, *soundLevel*, *trebleLevel*, *bassLevel*, *balanceLevel*, *zIndex*, *fontColor*, *fontAlign*, *fontFamily*, *fontStyle*, *fontSize*, *fontVariant*, *fontWeight*, *player*, *reusePlayer*, *playerLife*, *moveLeft*, *moveRight*, *moveUp*, *moveDown*, *focusIndex*, *focusBorderColor*, *focusBorderWidth*, *focusBorderTransparency*, *focusSrc*, *focusSelSrc*, *selBorderColor*, *freeze* e *timeBaseId*. A propriedade *timeBaseId* foi definida por Moreno (MORENO, 2014).

Todas elas são suportadas no perfil DR, com exceção das propriedades: *transIn* e *transOut*, que são utilizadas para definir transições, recurso não suportado no perfil DR, e *plane*, que define o plano gráfico a ser utilizado na hierarquia de apresentação dos elementos gráficos da TV Digital, não fazendo sentido no caso do rádio digital, que só utiliza um plano gráfico. Um exemplo de uso dos elementos *<area>* e *<property>* é apresentado na Listagem 16.

```

...
<media id="video" src="video.mp4">
  <property name="left" value="5%"/>
  <property name="top" value="10%"/>
  <property name="width" value="50%"/>
  <property name="height" value="50%"/>
  <area id="trecho1" begin="0s" end="10s"/>
  <area id="trecho2" begin="10s" end="20s"/>
</media>
...

```

Listagem 16. Trecho de código NCL que exemplifica o uso dos elementos `<area>` e `<property>`.

As fontes de caracteres suportadas pelo sistema de rádio digital devem estar disponíveis para todos os objetos de mídia que contém texto, como texto puro (*txt*) e objetos declarativos como SVG e HTML.

A Tabela 25 exibe um comparativo a respeito da presença de elementos de interface nos diferentes perfis da NCL 3.1 e no perfil DR.

Perfil	Presença
EDTV	SIM
RawDTV	NÃO
DR	SIM (com exceções)

Tabela 25. Presença dos elementos de Interface nos diferentes perfis.

A Tabela 26 apresenta os elementos discutidos nesta seção, seus atributos e conteúdo.

Elementos	Atributos	Conteúdo
<i>Área</i>	<i>id, coords, begin, end, beginText, beginPosition, endText, endPosition first, last, label, elip</i>	<i>empty</i>
<i>property</i>	<i>name, value, externable</i>	<i>empty</i>
<i>Port</i>	<i>id, component, interface</i>	<i>empty</i>

Tabela 26. Elementos de interface, seus atributos e conteúdo.

## 5.9.Variáveis globais

As variáveis globais de NCL são definidas como propriedades da mídia especial do tipo “application/x-ncl-settings”. Essas propriedades podem ser testadas em regras e elos, e possuem diferentes permissões de leitura e escrita.



As variáveis globais são divididas em grupos, sendo eles *system*, *user*, *si*, *metadata*, *default*, *service*, *channel* e *shared*.

O grupo *system* contém variáveis controladas pelo sistema embarcado do receptor e somente podem ser lidas. As variáveis definidas pela NCL 3.1 são: *language*, *caption*, *subtitle*, *returnBitRate(i)*, *screenSize*, *screenOrientation*, *screenVideoSize*, *screenBackgroundSize*, *screenGraphicSize*, *audioType*, *screenSize(i)*, *screenGraphicSize(i)*, *audioType(i)*, *classDevNumber(i)*, *classType(i)*, *classDevMin(i)*, *classDevMax(i)*, *classInfo(i)*, *classNumber(i)*, *CPU*, *memory*, *operatingSystem*, *luaVersion*, *luaSupportedEventClasses*, *nclVersion*, *nclProfiles* e *gingaNclVersion*.

As variáveis do grupo *system* suportadas no perfil DR são: *language*, *caption*, *subtitle*, *returnBitRate(i)*, *screenSize*, *screenOrientation*, *audioType*, *screenSize(i)*, *audioType(i)*, *classDevNumber(i)*, *classType(i)*, *classDevMin(i)*, *classDevMax(i)*, *classInfo(i)*, *classNumber(i)*, *CPU*, *memory*, *operatingSystem*, *luaVersion*, *luaSupportedEventClasses*, *nclVersion*, *nclProfiles* e *gingaNclVersion*, e possuem a mesma sintaxe da NCL 3.1. Não são suportadas as variáveis relacionadas a planos gráficos, *screenVideoSize*, *screenBackgroundSize*, *screenGraphicSize* e *screenGraphicSize(i)*, que são um recurso exclusivo para TV. A variável *classInfo(i)* somente pode assumir os valores *active* ou *onDemandMedia*, que são as classes para multidispositivos suportadas no perfil DR, conforme discutido na seção 6.6.

Outro grupo de variáveis é o *user*, cujas variáveis são gerenciadas pelo *software* embarcado do receptor e não podem ser modificadas pela aplicação NCL, sendo somente para leitura. Pertencem ao grupo *user* as variáveis *age*, *location*, *genre* e *language*. Todas as variáveis são suportadas no perfil DR.

O terceiro grupo de variáveis é o *si*, que expõe informações de sistema do canal sintonizado. As variáveis desse grupo são controladas pelo middleware e somente podem ser lidas pela aplicação NCL.

As variáveis do grupo *si* são adaptadas ao rádio digital, e diferem da TV Digital, sendo elas *stationLabel*, *numberOfServices*, *channelFrequency*, *signalQuality* e *serviceDecoding*, que indicam respectivamente o rótulo do emissora sintonizada, o número de serviços presentes no sinal multiplexado, a frequência central da emissora, em kHz, *signalQuality* indica a relação de erro de modulação (*Modulation Error Ratio*), em db (somente valores positivos), sendo

que o valor 0 indica ausência de sinal, e finalmente *serviceDecoding* indica se o serviço está sendo decodificado corretamente, devendo conter o valor “true” quando o BER (Bitrate Error Rate) for igual ou menor que  $10^{-4}$  (HOFMANN, 2003), limiar que indica um serviço *quasi error free* (QEF). Quando o BER for maior que  $10^{-4}$ , *serviceDecoding* deve assumir o valor “false”.

A variável *serviceDecoding* pode ser utilizada em conjunto com a variável *signalQuality* para a determinar, por exemplo, o limiar de recepção em db da estação. Um uso da variável *serviceDecoding*, por exemplo, poderá ser utilizada para chavear a recepção do áudio da emissora para um fluxo proveniente da internet, até que a recepção pelo ar seja restabelecida.

Um grupo para serviços de IPTV é o *metadata*, cujas variáveis são definidas pela Recomendação ITU-T H.750 da União Internacional de Telecomunicações. Esse grupo não é suportado no perfil DR pois tal recomendação não aborda o rádio digital.

Outro grupo é o *default*, que define valores padrões para alguns elementos do middleware, sendo que as variáveis desse grupo podem ser lidas e alteradas na aplicação NCL. As variáveis que pertencem ao grupo default são: *focusBorderColor*, *selBorderColor*, *focusBorderWidth* e *focusBorderTransparency*. Todas são suportadas no perfil DR.

O grupo *service* contém variáveis que podem ser lidas e modificadas. Fazem parte desse grupo as variáveis *currentFocus* e *currentKeyMaster*. Ambas propriedades são suportadas no perfil DR assim como definido na norma do Ginga.

Já o grupo *channel* agrupa variáveis relacionadas à captura de caracteres e teclado virtual. As variáveis desse grupo são: *keyCapture*, *virtualKeyboard* e *keyboardBounds*. Essas variáveis estão presentes no perfil DR.

Por fim, o grupo *shared* permite a definição de novas variáveis com permissão de leitura e escrita que persistem por todo o ciclo de vida do receptor. Esse grupo está presente no perfil DR.

Os grupos e variáveis cujas semânticas não foram discutidas nesta seção seguem as mesmas definições contidas na norma do Ginga na UIT (ITU, 2014a).

Um novo grupo de variáveis de nome *position* é definido na seção 6.3. Esse grupo define variáveis que indicam a localização do usuário num ambiente com mobilidade, diferentemente da variável *location* do grupo *user*, que somente

indica o endereço postal do usuário, funcionalidade útil para recepção fixa. Esse recurso visa atender o requisito 4 da seção 3.3, que versa sobre a importância da recepção em ambiente móvel no rádio digital.

A Tabela 27 apresenta as variáveis globais, seus grupos, semântica e valores permitidos. Destacadas em negrito estão as novas variáveis do perfil DR.

Grupo	Variável	Semântica	Possíveis valores
<b>system</b>	system.language	Idioma do áudio.	"ISO 639-1 code   ISO 639-2 code"
	system.caption	Idioma do <i>caption</i> .	"ISO 639-1 code   ISO 639-2 code"
	system.subtitle	Idioma de legenda.	"ISO 639-1 code   ISO 639-2 code"
	system.screenSize	Tamanho da tela do dispositivo em (linhas, pixels/linha), quando a classe não está definida.	"positive integer, positive integer"
	system.screenOrientation	Orientação da tela do dispositivo.	"portrait"   "landscape"
	system.returnBitRate(i)	Taxa de bits do canal de interatividade (i) em Kbps.	"positive real"
	system.audioType	Tipo de áudio do dispositivo, quando a classe não está definida.	"mono"   "stereo"   "5.1"
	system.screenSize(i)	Tamanho da tela da classe(i) de dispositivos.	"positive integer, positive integer"
	system.audioType(i)	Tipo de áudio da classe(i) de dispositivos.	"mono"   "stereo"   "5.1"
	system.classDevNumber(i)	Número de	"positive integer"

	)	dispositivos registrados na classe (i).	
	system.classType(i)	Tipo da classe(i).	"passive"   "active"   "mediaCapture"   "onDemandMedia" "
	system.classDevMin(i)	Número mínimo de dispositivos na classe(i).	"positive integer"
	system.classDevMax(i)	Número máximo de dispositivos na classe(i).	"positive integer"   "unbounded"
	system.classInfo(i)	Lista dos exibidores de mídia da classe(i).	"string"
	system.classNumber	Número de classes que foram definidas.	"positive integer"
	system.CPU	Performance da CPU em MIPS, com relação a sua capacidade de executar aplicações.	"positive real"
	system.memory	Tamanho mínimo de memória em Mbytes provido para aplicações.	"positive integer"
	system.operatingSystem	Tipo do Sistema Operacional.	"string"
	system.luaVersion	Versão do motor Lua suportado pelo receptor.	"string"
	system.luaSupportedEven	Lista de classes de	"string"

	tClasses	eventos suportados pelo NCLua, separados por ','.	
	system.nclVersion	Versão da linguagem NCL.	"string"
	system.nclProfiles	Perfis da linguagem suportados pelo receptor, separados por ','.	"string"
	system.gingaNclVersion	Versão do ambiente Ginga-NCL.	"string"
	system.*	Qualquer variável com o prefixo "system." não listada nesta tabela deve ser reservada para futuro uso.	
<b>user</b>	user.age	Idade do usuário.	"positive integer"
	user.location	A localização do usuário deve ser o código do país concatenado com o código postal. O código do país deve seguir o formato especificado pela ISO 3166-1 alpha 3.	"string"
	user.genre	Sexo do usuário.	"m"   "f"
	user.language	Língua do usuário.	"ISO 639-1 code   ISO 639-2 code"
	user.*	Qualquer variável com o prefixo "user." não listada nesta	

		tabela deve ser reservada para futuro uso.	
<b>si</b>	si.numberOfServices	Número de serviços disponíveis no canal sintonizado.	"positive integer"
	<b>si.stationLabel</b>	Nome da estação sintonizada.	"string"
	<b>si.channelFrequency</b>	Frequência central do canal sintonizado, em kHz.	"positive integer"
	<b>si.signalQuality</b>	O <i>Modulation Error Ratio</i> , em db. O valor 0 significa “sem sinal”.	“positive real”
	<b>si.serviceDecoding</b>	Informação sobre o estado de decodificação do sinal. “true” significa que o sinal está sendo recebido <i>quasi error free</i> , “false” deve ser utilizado no caso contrário.	“true”   “false”
	si.*	Qualquer variável com o prefixo “si.” não listada nesta tabela deve seguir as regras especificadas para o grupo.	
<b>default</b>	default.focusBorderColor	Cor padrão aplicada à margem de um elemento em foco.	"white"   "black"   "silver"   "gray"   "red"   "maroon"

			"fuchsia"   "purple"   "lime"   "green"   "yellow"   "olive"   "blue"   "navy"   "aqua"   "teal"
	default.selBorderColor	Cor padrão aplicada à margem de um elemento em foco quando ativado.	"white"   "black"   "silver"   "gray"   "red"   "maroon"   "fuchsia"   "purple"   "lime"   "green"   "yellow"   "olive"   "blue"   "navy"   "aqua"   "teal"
	default.focusBorderWidth	Largura padrão (em pixels) aplicada à margem de um elemento em foco.	"integer"
	default.focusBorderTransparency	Transparência padrão aplicada à borda de um elemento em foco.	"Valor real entre 0 e 1"   "Valor real na faixa [0,100] terminado com o caractere '%" (ex. 30%)" NOTA: "1" ou "100%" significa transparência completa e "0" ou "0%" significa sem transparência.
	default.*	Qualquer variável com o prefixo	

		"default." não listada nesta tabela deve ser reservada para futuro uso.	
<b>service</b>	service.currentFocus	Valor do atributo focusIndex do elemento <media> em foco.	<i>"positive integer"</i>
	service.currentKeyMaster	Identificador (id) do elemento <media> que detém o controle das chaves de navegação; se o elemento não estiver sendo apresentado ou não estiver pausado, o controle é do formatador.	<i>"string"</i>
	service.*	Qualquer variável com o prefixo "service." não listada nesta tabela deve seguir as regras especificadas para o grupo.	
<b>channel</b>	channel.keyCapture	Requisição de teclas alfanuméricas para aplicações NCL.	<i>"string"</i>
	channel.virtualKeyboard	Requisição de teclado virtual para aplicações NCL.	<i>"true"   "false"</i>
	channel.keyboardBounds	Região do teclado virtual (left, top,	<i>"positive integer, positive integer,</i>



		width, height).	<i>positive integer, positive integer"</i>
	channel.*	Qualquer variável com o prefixo “channel.” não listada nesta tabela deve seguir as regras especificadas para o grupo.	
<b>shared</b>	shared.*	Qualquer variável com o prefixo “shared.” não listada nesta tabela deve seguir as regras especificadas para o grupo.	

Tabela 27. Variáveis globais, seus grupos, semântica e possíveis valores.

## 5.10.Elementos de metadados

Os dois elementos considerados estão apresentados na Listagem 17.

```
<meta>
<metadata>
```

Listagem 17. Elementos de metadados.

Os elementos de metadados contém informações relacionadas ao conteúdo do documento. As informações expressas pelos dois tipos de elementos de metadados, *<meta>* e *<metadata>*, são processadas para, por exemplo, facilitar a catalogação e buscas de aplicações com determinadas características.

O elemento *<meta>* possui dois atributos, name e content, que definem o nome do metadado e seu conteúdo. O elemento *<metadata>* contém como filha uma estrutura RDF (*Resource Description Framework*), que permite uma estrutura mais complexa para a organização dos metadados. A Listagem 18 apresenta um exemplo de uso dos metadados.

```
<meta name="author" content="Rafael Diniz" />
<meta name="data" content="2015-01-10" />
<meta name="license" content="gpl 3.0" />
```

Listagem 18. Exemplo de uso de metadados.

Pelo fato dos elementos de metadados muitas vezes serem ignorados pelo exibidor NCL, sendo úteis para outras ferramentas de organização, não existe um motivo para a remoção desses elementos do perfil DR, portanto, eles fazem parte do perfil da NCL para rádio digital. A Tabela 28 compara a presença dos elementos de metadados nos diferentes perfis.

Perfil	Presença
EDTV	SIM
RawDTV	NÃO
DR	SIM

Tabela 28. Presença dos elementos de metadados nos diferentes perfis.

A Tabela 29 contém os elementos para especificação de metadados, seus atributos e conteúdo.

Elementos	Atributos	Conteúdo
<i>Meta</i>	<i>name, content</i>	<i>empty</i>
<i>Metadata</i>	<i>Empty</i>	<i>RDF tree</i>

Tabela 29. Elementos para especificação de metadados, seus atributos e conteúdo.

### 5.11. Resumo das características do perfil DR da NCL

Considerando os módulos definidos na NCL 3.1 em seu perfil EDTV, as principais características do perfil DR são a remoção de algumas entidades da linguagem e a alteração de outras, conforme listado a seguir.

- Elementos *<descriptorBase>*, *<descriptor>*, *<descriptorParam>*, pertencentes ao módulo *Descriptor*, e *<regionBase>*, *<region>*, pertencentes ao módulo *Layout* não estão definidos no perfil DR, no entanto funcionalidade equivalente é obtida através do uso do elemento *<property>*;
- Elementos *<descriptorSwitch>* e *<defaultDescriptor>*, do módulo *DescriptorControl*, não fazem parte do perfil DR;

- Elementos *<transition>* e *<transitionBase>*, dos módulos *Transition* e *TransitionBase* não são definidos, e também não é permitida a definição de transições via elemento *<property>*;
- Elementos *<importBase>*, *<importedDocumentBase>* e *<importNCL>*, pertencentes do módulo *Import* não fazem parte do perfil DR;
- Elemento *<linkParam>*, módulo *Linking*, não é definido;
- Atributo *operator* do elemento *<compoundAction>*, atributos *eventType* e *transition* do elemento *<simpleCondition>* e atributos *eventType* e *actionType* do elemento *<simpleAction>*, módulo *CausalConnector*, não são definidos;
- Atributos *clip* e *coords* do elemento *<area>*, módulo *MediaContentAnchor*, não compõem a NCL DR;
- Para o elemento *<property>*, módulo *PropertyAnchor*, as propriedades pré-definidas *transIn*, *transOut* e *plane* não são definidas;
- Variáveis globais são definidas como propriedades da mídia especial do tipo *application/x-ncl-settings* e divididas em grupos. Do grupo *system* não são definidas: *screenVideoSize*, *screenBackgroundSize*, *screenGraphicSize* e *screenGraphicSize(i)*. Um novo grupo de variáveis para indicar a posição do receptor com precisão é adicionado (seção 6.3). As propriedades do grupo *si* apresenta modificações, sendo elas: *stationLabel*, *numberOfServices*, *channelFrequency*, *signalQuality* e *serviceDecoding*. O grupo *metadata* não é definido.
- No atributo *src* de um elemento *<media>* os esquemas específicos para TV, “*ts:*” e “*dsm-cc:*”, não são suportados, sendo suportado, de acordo com o sistema de rádio digital em uso, o esquema “*drm:*”, referente ao sistema Digital Radio Mondiale, ou “*hd:*”, referente ao sistema HD Radio;
- As características da NCL no perfil DR não cobertas neste capítulo são iguais às definições contidas na norma ITU H.761, que define a NCL 3.1.

## 6

### Ambiente de apresentação Ginga e novas mídias

Neste capítulo são apresentados e definidos alguns aspectos do ambiente de apresentação, o middleware Ginga, para uso em rádio digital, como o suporte a tipos de mídia e variáveis globais que possuem relevância na radiodifusão digital e diferem da atual especificação do Ginga existente para TV Digital. Também serão abordadas outras funcionalidades da NCL: os comandos de edição, que permitem a edição ao vivo de uma aplicação, o suporte a múltiplos dispositivos, que permite a execução de partes de uma aplicação em dispositivos secundários conectados à mesma rede do receptor, e o suporte a objetos imperativos NCLua.

Com relação aos tipos de mídia suportados pelos diferentes perfis de terminais para o Sistema Brasileiro de TV Digital (ABNT, 2015), dois novos tipos são introduzidos, um para permitir a apresentação de gráficos vetoriais e outro para sintetização de voz. Monomídias visuais e aurais que otimizam o uso da banda do rádio digital são também discutidas.

Um recurso importante para o rádio digital é o acesso pela aplicação NCL a informações de geolocalização. Novas propriedades globais são apresentadas de forma a permitir que uma apresentação NCL acesse os dados de localização instantânea do receptor, normalmente provenientes de sinais GPS. Uma base temporal para a sincronia fina de mídias no rádio digital também é apresentada.

#### 6.1.Monomídias

Esta seção introduz monomídias cujo suporte nos receptores de rádio digital é muito relevante, de forma que sejam atacados os requisitos expostos no Capítulo 3, relacionados à otimização no uso da banda da transmissão (requisito 1), garantia que recursos visuais e de áudio ricos estejam disponíveis (req. 2) e que se aprimore a interatividade por áudio (req. 3).

### 6.1.1. Gráficos Vetoriais

O suporte a gráficos vetoriais permite que uma apresentação NCL utilize objetos visuais sem preocupação com o tamanho de tela do receptor, visto que as mídias vetoriais são aumentadas ou diminuídas sem perda de qualidade. Além disso as mídias vetoriais ocupam um menor tamanho em bits do que mídias baseadas em mapa de pixels (*bitmap*), o que permite uma redução no tamanho total de uma aplicação NCL, fator importante para as transmissões de baixo bitrate características do rádio digital.

O Scalable Vector Graphics (SVG) é utilizado em outros sistemas de radiodifusão como o ATSC M/H (ATSC, 2009), sendo também o padrão W3C para gráficos vetoriais na Web. O SVG foi o formato vetorial selecionado para ter seu reprodutor adicionado ao Ginga para rádio digital.

O padrão SVG é uma linguagem declarativa XML para descrever gráficos bidimensionais. SVG permite três tipos de objetos: entidades vetoriais, como linhas e curvas, objetos multimídia como áudio e imagens *bitmap*, e texto. Os objetos declarados podem ser agrupados, transformados e formar objetos compostos. A linguagem é modular e permite a definição de perfis com diferentes requisitos de recursos de CPU e memória. O tipo MIME do SVG é *image/svg+xml*. Duas extensões devem ser suportadas pelo Ginga, o “.svg” e o “svgz”, que é o SVG comprimido com GZIP (IETF, 1996). O player SVG implementado obedece a API de player do Ginga, conforme definido em norma (ITU, 2014a).

A norma mais recente recomendada para dispositivos móveis é a SVG Tiny 1.2 (W3C, 2008). O nível de recursos suportado pelo reprodutor SVG do Ginga para rádio será o SVG-animated<sup>12</sup>, que inclui todos os recursos estáticos do padrão adicionado do módulo *TimedAnimation*, que permite a descrição de animações de forma declarativa.

Uma característica importante do SVG é o suporte oferecido por ferramentas de autoria como o Adobe Illustrator e Corel Draw, além de ferramentas em código fonte aberto, como o Inkscape e o SVG-edit.

Como exemplo, a Figura 15 ilustra a tela de um aplicativo simples contendo informações sobre uma emissora e sua programação. Em um cenário de pior caso

---

<sup>12</sup> SVG-animated features: <http://www.w3.org/Graphics/SVG/feature/1.2/#SVG-animated>

em termos de bitrate, em uma emissora AM, por exemplo, a versão do aplicativo utilizando o logotipo da emissora em SVG ficou com o tamanho de 3824 bytes e a versão utilizando PNG 11172 bytes. Essa diferença reduz de 22,3s (versão PNG) o tempo de envio da aplicação a 4000bit/s para 7,6s (versão SVG), uma diferença

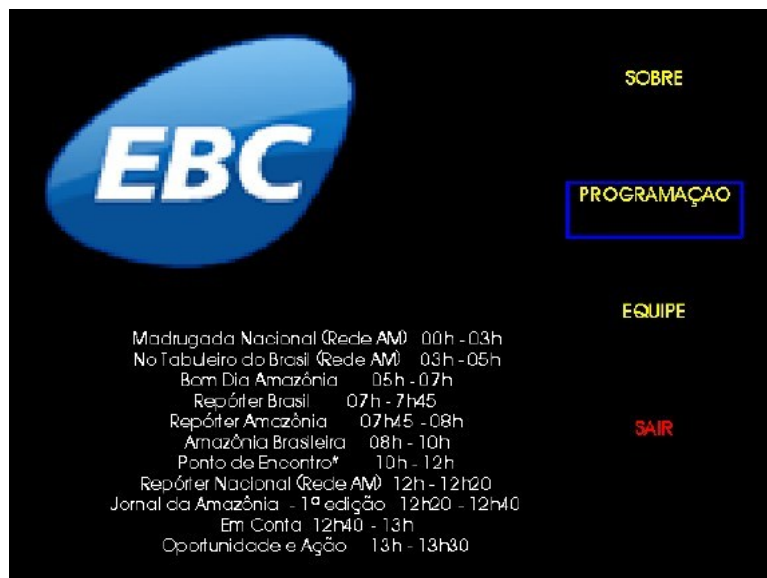


Figura 15. Aplicativo Ginga para rádio digital.

considerável.

Um uso importante que é feito do SVG e que poderá ser utilizado no rádio digital é a visualização de dados. Por exemplo, informações relacionadas a condições do trânsito, profundidade de rios navegáveis ou locais de incidência de doenças são perfeitamente modeláveis em SVG com um gasto de bytes baixo. Até mesmo a geração dinâmica de SVG a partir de um script NCLua é possível.

### 6.1.2.Sintetização de voz

As mídias declarativas para sintetização de voz são utilizadas para definir um texto que será lido e propriedades associadas a cada parte do texto, como a velocidade da fala, presença de ênfases e pausas.

Ao Ginga para rádio digital deve ser acrescentado o suporte ao tipo de mídia XML Speech Synthesis Markup Language 1.1 (W3C, 2010), que é recomendação W3C para sintetização de voz. Ao menos uma voz para português brasileiro deverá ser suportada. O tipo MIME do SSML é *application/ssml+xml*. O player SSML desenvolvido obedece a API de player do Ginga, conforme definição em norma (ITU, 2014a).

Âncoras poderão ser definidas num documento NCL via elemento `<area>`, através do atributo `label`. O valor de `label` deve conter um identificador especificado via atributo `name` do elemento `<mark>` do SSML, que corresponde a uma marcação no documento. Por exemplo, é possível usar uma âncora para disparar uma ação no momento em que a leitura do texto ultrapassa uma marcação, ou ainda utilizar a âncora para determinar o ponto inicial ou final de um trecho de texto.

Mídias SSML permitem uma grande redução do tamanho de uma aplicação que utilize falas. Por exemplo, uma pequena fala que ocupa 481 bytes em um arquivo SSML, ocupa 13342 bytes quando armazenada num arquivo de áudio com ~8s de duração no formato AAC a uma taxa de 12kbit/s. O arquivo SSML tem um tamanho 25 vezes menor que o áudio codificado. Além disso, a sintetização de voz permite a adaptação de aplicações NCL para analfabetos e pessoas com restrições de visão.

### 6.1.3. Áudio e imagem

Como por diversas vezes mencionado, um problema do rádio digital é sua limitada largura de banda para o envio de mídias. Para mitigar esse problema é interessante que os tipos de mídia suportadas pelo middleware consigam o máximo de compressão possível. Já foi visto que as mídias SVG e SSML consomem um baixo bitrate para a transmissão de imagens vetoriais e para síntese de voz. No entanto, é importante que existam opções de baixo bitrate também para imagens baseadas em mapa de bits e áudio.

Um novo formato para imagens, que utiliza a mesma codificação espacial dos quadros do padrão de codificação de vídeo H.265 (ITU-T, 2014c), apresenta maior compressão<sup>13</sup> que os padrões JPEG, JPEG2000, JPEG XR, WebP, de acordo com pesquisa da Mozilla Corporation (AAS, 2014). Esse novo formato chama-se Better Portable Graphics (BELLARD, 2015) e deve ser suportado pelo Ginga para rádio digital.

Para áudio existem dois codificadores que permitem uma grande compressão tanto para conteúdo com fala, como para música, e suportam desde

<sup>13</sup> Comparativo entre BPG, JPEG, JPEG2000, JPEG XR e WebP: <http://xooyoozoo.github.io/yolo-octo-bugfixes/>

conteúdo monoaural com pequena largura de banda de áudio até conteúdo surround multi-canal com banda de áudio completa. São eles o Opus (IETF, 2012), e o xHE-AAC, que é um dos perfis de codificação definido por sua norma (ISO, 2012). Até o momento ainda não existem implementações abertas ou gratuitas do xHE-AAC, razão pela qual o codificador Opus foi escolhido para ter suporte no Ginga.

Além dessas mídias, pode ser interessante que o middleware Ginga suporte tipos de mídia nativas do sistema de rádio digital em uso. Por exemplo, no caso do sistema Digital Radio Mondiale, as mídias *Animated PNG* (ETSI, 2015) e *Journaline* (ETSI, 2008), poderão ser suportadas.

Mais uma vez, cabe lembrar que todos os novos players de mídia implementados obedecem a API de player do Ginga, conforme definido em norma (ITU, 2014a).

## 6.2.Objetos imperativos NCLua

A linguagem adotada pelo Ginga para implementar objetos imperativos em documentos NCL é Lua. Além da biblioteca padrão de Lua, foram adicionados alguns módulos extras e o ciclo de vida de um objeto Lua segue o modelo de eventos da NCL. Objetos Lua devem ser referenciados pelo atributo *src* de um objeto *<media>* e ter o tipo *application/x-ginga-NCLua*. Objetos de mídia imperativos Lua escritos para serem embarcados em documentos NCL são conhecidos como NCLua.

Todas as bibliotecas padrão do Lua são suportadas, com exceção da função *loadlib* da biblioteca *package*, e das funções *clock*, *execute*, *exit*, *getenv*, *remove*, *rename*, *tmpname* e *setlocale* da biblioteca *os*, assim como especificado na norma H.761.

Além das bibliotecas padrão Lua, NCLua define 5 novos módulos: *canvas*, *event*, *settings*, *persistent* e *security*. O módulo *canvas* permite realizar operações gráficas e textuais na região atribuídas ao objeto NCLua. O módulo *event* expõe funções para manipulação de eventos, que são divididos em classes.

As classes de eventos definidas são: *key*, *pointer*, *ncl*, *edit*, *udp*, *tcp*, *http*, *sms*, *si*, *metadata* e *user*. O módulo *settings* exporta uma tabela global que contém as variáveis globais, definidas pelo tipo de mídia *application/x-ncl-settings*, e que



são somente para leitura, enquanto o módulo *persistent* exporta uma tabela sem valores pré-definidos que pode ser escrita e lida sem restrições por qualquer objeto imperativo da apresentação. Finalmente o módulo *security* expõe uma API com funções para assinatura digital, *hash* e criptografia, representadas nas classes *signature*, *digest* e *cipher*.

No módulo *event*, os tipos de eventos da classe *ncl* podem ser *presentation*, *selection* ou *attribution* e os tipos de eventos definidos para a classe *si* são *services*, *epg* e *time*.

Dentre as classes definidas no módulo *event* do NCLua, todas são suportadas no perfil DR, inclusive a classe *si*, que define tipos de eventos que permitem o acesso a informações que chegam embarcadas no sinal de rádio digital.

A classe *metadata* não está presente no perfil DR pois a especificação que a define, ITU-T H.750, não contempla rádio digital, a exemplo da ausência do grupo de variáveis *metadata* no conjunto das variáveis globais da NCL DR.

O módulo *security* implementa funções de assinatura, hash e criptografia, mas não implementa o suporte a nenhum protocolo de comunicação segura, como o protocolo TLS (*Transport Layer Security*). Devido à presença da classe *http*, que já suporta o estabelecimento de conexão segura via *https* (HTTP over TLS), o aumento de complexidade e tamanho do *player* NCLua para a implementação do módulo *security* não é justificável. Por esse motivo, a classe *security* não é suportada no perfil DR da NCL.

### 6.3. Acesso a informações de geolocalização do receptor

Na NCL 3.1 o único recurso disponível para indicar a localização do usuário é a variável *user.location*, que contém o código postal do endereço do utilizador. Essa informação atende ao contexto de receptores fixos, mas não ao contexto geral de receptores de rádio, que conta com um grande número de receptores portáteis e automotivos.

O acesso às informações de geolocalização permitem que uma aplicação NCL se adapte de acordo com a localização do receptor. O suporte mais adequado para o acesso às informações de geolocalização, devido a sua natureza unívoca, se dá através de um novo grupo de variáveis definidas como propriedades mídia

especial do tipo “application/x-ncl-settings”, do escopo *position*. As variáveis pertencentes ao grupo *position* são listadas na Tabela 30.

Variável	Valor	Descrição
latitude	Número real, faixa de [-90.0, 90.0], unidade em graus.	Latitude do receptor. (Somente leitura)
longitude	Número real. Faixa de [-180.0, 180.0], unidade em graus.	Longitude do receptor. (Somente leitura)
altitude	Número real, unidade em metros.	Altitude do receptor. (Somente leitura)
accuracy	Número real positivo, unidade em metros.	Precisão das coordenadas. (Somente leitura)
hasGeolocation	“true” ou “false”.	Indica se o receptor possui sistema de geolocalização. (Somente leitura)
locationAvailable	“true” ou “false”	Indica se as informações disponíveis estão atualizadas e corretas. Normalmente os receptores de GPS tem um tempo de <i>warm up</i> . (Somente leitura)
geolocationActive	“true” ou “false”. Default é “false”.	Indica se o sistema de geolocalização está ativado. (Leitura e Escrita)
geolocationMinTime	Seconds”s”, onde Seconds é um número real positivo. Default é 1s	Define o tempo mínimo de espera para a atualização dos valores de latitude e longitude e altitude, em segundos. (Leitura e Escrita)
geolocationMinDistance	Número real positivo, em metros. Default é 0.	Define a menor mudança de localização que ativa a atualização das coordenadas,

		em metros. O valor “0” significa que a distância percorrida não influencia na atualização das coordenadas. (Leitura e Escrita)
--	--	--

Tabela 30. Variáveis do grupo *location* que expõe informações de geolocalização do receptor.

Antes de utilizar os valores de latitude e longitude, o aplicativo deve testar a variável *hasGeolocation*, para verificar se o receptor possui algum sistema de geolocalização. Caso o receptor possua suporte, essa variável recebe o valor *true*. Exemplos de sistemas de geolocalização que utilizam dados de satélites são GPS, GLONASS e Galileo, aqueles baseados em triangulação de rádio utilizam-se de redes de telefonia móvel ou WiFi, e existem ainda os sistemas baseados na localização do endereço IP, como o GeoIP.

No caso do receptor possuir suporte a geolocalização, a variável *geolocationActive* poderá ter o valor *true* atribuído a ela pela aplicação NCL, o que deverá ativar o sistema de geolocalização no receptor. Após a ativação do sistema de geolocalização, o aplicativo deve esperar a variável *locationAvailable* passar para o valor *true*. Quando *locationAvailable* assumir o valor *true*, as propriedades contendo a latitude, longitude, altitude e precisão (*accuracy*) estarão corretas e atualizadas.

Vale ressaltar que para se minimizar o uso da bateria em um receptor portátil, deve-se procurar ajustar as variáveis *geolocationMinTime* e *geolocationMinDistance* para o maior valor possível, evitando atualizações desnecessárias por parte da aplicação. Essas duas propriedades são inspiradas no método *requestLocationUpdates()* da API de localização do sistema operacional Android.

Dentre os casos de uso, um muito relevante é o de uma aplicação que apresenta informações de trânsito de uma cidade, de acordo com a localização do receptor. Nesse caso, é possível exibir notícias e mapas, por exemplo, pertinentes à localização do ouvinte. No caso de alertas de emergência esse recurso pode ser muito importante para guiar cidadãos em perigo.

## 6.4.Base temporal

A marcação de tempo definida para a TV Digital brasileira e utilizada no Ginga é a *Normal Play Time* (ISO, 1998). As bases temporais utilizadas na TV Digital possuem características fortemente acopladas ao sistema de multiplexação e às convenções utilizadas nesse sistema, como o MPEG 2 Transport Stream (ISO, 2013) e o MPEG 2 DSM-CC (ISO, 1998). Uma nova base temporal, mais simples, mas baseada na NPT, é de grande relevância para permitir a sincronização do fluxo de áudio principal a eventos de apresentação de uma aplicação.

Devido os sistemas de rádio digital não terem o requisito de sincronizar duas ou mais mídias, os mesmos não transportam nenhum tipo de base temporal. Por esse motivo, essa seção introduz o suporte a bases temporais no Ginga para rádio digital, o que permite que seja possível a sincronia fina do fluxo de áudio da emissora com mídias de uma aplicação interativa.

No caso do sistema Digital Radio Mondiale, por exemplo, em sua única estrutura relacionada ao tempo, o *Service Description Channel*, na entidade *Time and date information data entity*, a data é transmitida com o uso dos campos *Modified Julian Date* (MJD), UTC (*hours and minutes*) e outros dois parâmetros opcionais que indicam o fuso-horário da região alvo de recepção. As informações transmitidas pelo DRM permitem a sincronização do horário entre a emissora e um receptor a cada minuto, que é uma granularidade pobre para uma aplicação multimídia. Além disso, o uso de UTC como base de tempo impede que uma aplicação possa ser reutilizada sem adaptações, e vincula a autoria da aplicação ao horário de apresentação da mesma.

A solução proposta para o rádio digital permite o uso de mais de um tipo de base temporal. Um identificador do tipo da base temporal, de nome *timeBaseType*, é utilizado para sinalizar o tipo da base em uso e o valor da marcação de tempo é associado ao parâmetro *timeBaseValue*.

O campo *timeBaseType* poderá assumir o valor 0, que é reservado e não indica um tipo de base temporal, mas uma ação do tipo “faça agora” (*do it now*), útil por exemplo, para comandos de edição, e tem tamanho de 7 bits. Quando *timeBaseType* for 0, o valor em *timeBaseValue* deve ser ignorado.

Para indicar que uma base temporal é tipo *Radio Time Base* (RTB), definida a seguir, *timeBaseType* deve conter o valor 1 e *timeBaseValue* deve conter um valor RTB com tamanho de 33 bits. Outros valores para o campo *timeBaseType* são reservados para futuras definições de outras bases temporais.

A entidade utilizada para transportar um valor de base temporal no rádio digital é apresentada na Tabela 31.

Sintaxe	Número de bits
<i>TimeBaseReference</i> () {	
<i>timeBaseId</i>	8
<i>timeBaseType</i>	7
<i>timeBaseValue</i>	33
}	

Tabela 31. Sintaxe de envio de valores de tempo em dada base temporal.

O *Radio Time Base* consiste de uma base de tempo composta de um valor inteiro positivo, cujo *System Clock Frequency* equivale ao número máximo de amostras de áudio que podem ser transmitidas por segundo, por exemplo, 48.000 para o sistema DRM. Nesse caso, a cadência do incremento do RTB é 48.000 unidades por segundo.

Possui tamanho de 33 bits, o que permite mais de 2 dias para que a base atinja seu valor máximo e volte ao inicial na cadência de 48.000Hz, conforme apresentado a seguir:

$$\text{Período RTB} = (((2^{33}) / 48000 \text{ Hz}) / 3600 \text{ seg/hora}) / 24 \text{ hora/dia} = 2 \text{ dias, } 1 \text{ hora e } 42 \text{ minutos}$$

Essa característica garante uma boa faixa de tempo para a definição de âncoras temporais na autoria de aplicações que utilizem referências de tempo baseadas na *Radio Time Base*.

Além das marcações de tempo, o fluxo que transporta os valores da base temporal deve também comportar entidades que indicam o início e final da base de tempo, semelhante ao *NPT Endpoint Descriptor* do protocolo DSM-CC.

A conversão de valores RTB em segundos e milisegundos deverá ser feita da seguinte forma:

```

RTB_seconds = RTB / System_Clock_Frequency
RTB_milliseconds = ( ( RTB * 1000 ) / System_Clock_Frequency )
- ( RTB_seconds * 1000 )

```

Onde *System\_Clock\_Frequency* tipicamente é 48.000, como no caso do sistema DRM.

A entidade que indica o início e o final de uma base temporal tem sua sintaxe definida na Tabela 32, pela entidade genérica *TimeBaseEndpoint*.

Sintaxe	Tamanho em bits
<i>TimeBaseEndpoint</i> () {	
<i>timeBaseId</i>	8
<i>timeBaseType</i>	7
<i>startRTB</i>	33
<i>Reserved</i>	7
<i>stopRTB</i>	33
}	

Tabela 32. Entidade *TimeBaseEndpoint*.

Uma base temporal comumente estará associada a um serviço de áudio via sinalização dos descritores de serviços do sistema, permitindo a sincronia fina entre o serviço de áudio principal e eventos da aplicação NCL.

Assim que uma base temporal *timeBaseId* é recebida, a sua cadência deve ser mantida pelo middleware. Essa base é assumida como estando no estado ocorrendo. Assim que a entidade *TimeBaseEndpoint* chega ao receptor, o middleware sabe o instante exato de início da base temporal pelo parâmetro *startRTB*, e o instante do final da base temporal pelo parâmetro *stopRTB*. Quando o middleware começa a receber o valor de tempo de uma base temporal no estado ocorrendo e essa base é diferente de uma outra que ainda não terminou, essa última deve ter sua cadência pausada. A base é dita estar no estado pausado. Mesmo uma base temporal pausada deve continuar a receber valores de RTB (sempre valores idênticos), pois se estrutura *TimeBaseReference* deixar de ser recebida por mais de 5 minutos o receptor deverá assumir que a base terminou.

A cadência do passo da base temporal deve ser a cadência da quantidade de amostras de áudio recebidas (ou que deveriam ter sido recebidas, no caso de falhas<sup>14</sup>), de forma que a sincronia não seja perdida, mesmo que a taxa de envio da base temporal não seja alta, economizando bits na transmissão. No caso da TV Digital, o intervalo máximo de envio entre dois valores de tempo é de 1s, mas no caso do rádio digital esse valor pode ser mais alto, adaptado à necessidade de sincronia do emissor e à disponibilidade de bits no multiplex.

Para interpolar o instante exato do tempo RTB entre dois valores recebidos pela estrutura *TimeBaseReference*, não temos, como na TV Digital, um valor STC (*System Time Clock*) de referência no qual a base temporal estaria atrelada. Para resolver o problema, o valor de pulsação (*tick*) do relógio RTB deve tomar como referência a pulsação da geração das amostras de áudio. Mais precisamente, deve tomar como referência a primeira amostra de áudio contida nos quadros, que são enviados a uma taxa constante, com relação ao relógio do transmissor. Por exemplo, no caso do sistema DRM, essa unidade é o *super-frame* de transmissão, que, dependendo do modo de transmissão, pode ter duração de 400ms, nos modos para frequências abaixo de 30MHz (OT, OC e OM) ou 200ms, para a banda do VHF.

## 6.5. Comandos de edição

Comandos de edição permitem diversas ações em uma aplicação NCL, como iniciar a aplicação, parar, salvar ou modificar a mesma. Esses comandos podem ser disparados externamente à aplicação, através do envio do comando via sinal de rádio, ou internamente, disparado através de um objeto imperativo NCLua.

Um comando de edição NCL é transmitido embutido em uma estrutura chamada *Editing Command event descriptor*. Os campos dessa estrutura são o *eventId*, que é um identificador único do comando de edição, *eventNPT*, que é um selo de tempo que indica o momento de ativação do comando, *commandTag*, que

---

<sup>14</sup> Devido ao quadro ter um tamanho de tempo definido e um número de amostras definido, é trivial ao receptor estimar o número de amostras em caso de falha na recepção de alguns quadros.

é um identificador do tipo de comando, e *privateDataPayload*, que é um campo privado que contém os parâmetros do comando de edição.

Os possíveis tipos de comando que podem ser indicados no *commandTag* são: *openBase*, *activateBase*, *deactivateBase*, *saveBase*, *closeBase*, *addDocument*, *removeDocument*, *startDocument*, *stopDocument*, *pauseDocument*, *resumeDocument*, *saveDocument*, *addRegion*, *removeRegion*, *addRegionBase*, *removeRegionBase*, *addRule*, *removeRule*, *addRuleBase*, *removeRuleBase*, *addConnector*, *removeConnector*, *addDescriptor*, *removeDescriptor*, *addDescriptorSwitch*, *removeDescriptorSwitch*, *addDescriptorBase*, *removeDescriptorBase*, *addTransition*, *removeTransition*, *addTransitionBase*, *removeTransitionBase*, *addImportBase*, *removeImportBase*, *addImportedDocumentBase*, *removeImportedDocumentBase*, *addImportNCL*, *removeImportNCL*, *addNode*, *removeNode*, *addInterface*, *removeInterface*, *addLink* e *removeLink*.

Conforme já discutido, o perfil DR não tem regiões, descritores, *switch* de descritores, transições e os elementos de importação, portanto, os seguintes comandos de edição não são suportados no perfil DR: *addRegionBase*, *removeRegionBase*, *addDescriptor*, *removeDescriptor*, *addDescriptorSwitch*, *removeDescriptorSwitch*, *addDescriptorBase*, *removeDescriptorBase*, *addTransition*, *removeTransition*, *addTransitionBase*, *removeTransitionBase*, *addImportBase*, *removeImportBase*, *addImportedDocumentBase*, *removeImportedDocumentBase*, *addImportNCL* e *removeImportNCL*. Todos os outros comandos são suportados, e o comportamento relativo às bases privadas, que controla o acesso e permissões das aplicações NCL no receptor, é o mesmo da NCL 3.1.

No perfil DR o campo *eventTBV* assume o lugar do campo *eventNPT* definido na NCL 3.1. A Tabela 33 apresenta a estrutura do *EditingCommandEvent* para o perfil DR, na qual é utilizada a marcação de tempo definida na seção 6.4.

Sintaxe	Número de bits
<i>EditingCommandEvent</i> ( ) {	
<i>eventId</i>	16
<i>timeBaseType</i>	7
<i>eventTBV</i>	33



<i>privateDataLength</i>	16
<i>commandTag</i>	8
<i>privateDataPayload</i>	N
}	

Tabela 33. Estrutura do *EditingCommandEvent*.

O tamanho N do *privateDataPayload* depende do tamanho dos parâmetros do comando de edição, cujo tamanho máximo é de 65535 bytes. Alguns campos presentes no *Editing Command event descriptor* para TV, como o *sequenceNumber*, *finalFlag* e *FCS* não são necessários e portanto não são utilizados, pois os dois primeiros são utilizados para segmentação da estrutura e o terceiro para verificação da integridade dos dados, recursos que deverão ser providos na camada de transporte dos sistemas de rádio digital.

O parâmetro *nptTrigger*, utilizado pelo comando *startDocument*, também tem seu nome mudado para *tbvTrigger*, e deve conter o mesmo tipo da base temporal indicada na *timeBaseType* do *Editing Command Event Descriptor*.

O parâmetro *baseId* deve conter um valor unívoco relacionado ao serviço e à emissora que carrega o aplicativo Ginga. Por exemplo, no caso do DRM, esse valor tem 24 bits.

O transporte dos comandos de edição assim como o esquema da URI de objetos são dependentes do padrão de rádio digital, e são discutidos no Capítulo 6.

A Tabela 34 apresenta os comandos de edição suportados no perfil DR.

Comando	commandTag	Descrição
openBase ( <i>baseId</i> , location, meta)	0x00	Idem ITU-T H.761.
activateBase ( <i>baseId</i> )	0x01	Idem ITU-T H.761.
deactivateBase ( <i>baseId</i> )	0x02	Idem ITU-T H.761.
saveBase ( <i>baseId</i> , location)	0x03	Idem ITU-T H.761.
closeBase ( <i>baseId</i> )	0x04	Idem ITU-T H.761.

addDocument (baseId, {uri, id}+, meta)	0x05	Idem ITU-T H.761.
removeDocument (baseId, documentId)	0x06	Idem ITU-T H.761.
startDocument (baseId, documentId, interfaceId, offset, timeBaseId, tbvTrigger)  NOTE. The offset parameter is a time value.	0x07	<p>Inicia a reprodução de um documento NCL em uma base privada ativa, iniciando a apresentação a partir de uma interface específica do documento.</p> <p>A referência de tempo provida pelo campo <i>tbvTrigger</i> determina a posição inicial de tempo com relação a base temporal identificada pelo campo <i>timeBaseId</i>.</p> <p>Três casos podem ocorrer:</p> <ul style="list-style-type: none"> <li>i) Se <i>tbvTrigger</i> é válido e maior ou igual ao valor de tempo corrente da base temporal identificada por <i>timeBaseId</i>, a apresentação do documento deve esperar até que o tempo corrente contenha o valor especificado em <i>tbvTrigger</i> para ser iniciada a partir seu momento de início+ <i>offset</i>.</li> <li>ii) Se <i>tbvTrigger</i> é válido e menor que o tempo corrente da base temporal identificada por <i>timeBaseId</i>, a aplicação deve ser iniciada imediatamente a partir do seu momento de início +<i>offset</i>+(tempo corrente – <i>timeTrigger</i>) segundos.</li> </ul>

		<p>NOTA. Somente neste caso o valor do parâmetro <i>offset</i> pode ser um tempo negativo, pois <math>offset + (\text{tempo corrente} - \text{timeTrigger})_{\text{seconds}}</math> pode ser um valor de tempo positivo.</p> <p>iii) Se <i>tbvTrigger</i> não é válido (Time base type = 0), a aplicação deve iniciar sua apresentação imediatamente de seu momento de início+<i>offset</i></p> <p>NOTA. Se o parâmetro <i>interfaceId</i> for especificado como "null", todos os elementos &lt;port&gt; do elemento &lt;body&gt; devem ser disparados (iniciados).</p> <p>Se o parâmetro <i>offset</i> for especificado como "null", ele deve assumir o valor "0".</p> <p>Se a aplicação não está na base privada, ou se a aplicação já está executando, ignorar o comando.</p>
stopDocument (baseId, documentId)	0x08	Idem ITU-T H.761.
pauseDocument (baseId, documentId)	0x09	Idem ITU-T H.761.
resumeDocument (baseId, documentId)	0x0A	Idem ITU-T H.761.
saveDocument (baseId, documentId, location)	0x2E	Idem ITU-T H.761.
addRule (baseId, documentId, xmlRule)	0x0F	Idem ITU-T H.761.

removeRule (baseId, documentId, ruleId)	0x10	Idem ITU-T H.761.
addRuleBase (baseId, documentId, xmlRuleBase)	0x11	Idem ITU-T H.761.
removeRuleBase (baseId, documentId, ruleBaseId)	0x12	Idem ITU-T H.761.
addConnector (baseId, documentId, xmlConnector)	0x13	Idem ITU-T H.761.
removeConnector (baseId, documentId, connectorId)	0x14	Idem ITU-T H.761.
addConnectorBase (baseId, documentId, xmlConnectorBase )	0x15	Idem ITU-T H.761.
removeConnector Base (baseId, documentId, connectorBaseId)	0x16	Idem ITU-T H.761.
addNode (baseId, documentId, compositeId, {uri, id}+, meta)	0x27	Idem ITU-T H.761.

removeNode(baseId, documentId, compositeId, nodeId)	0x28	Idem ITU-T H.761.
addInterface (baseId, documentId, nodeId, xmlInterface)	0x29	Idem ITU-T H.761.
removeInterface (baseId, documentId, nodeId, interfaceId)	0x2A	Idem ITU-T H.761.
addLink (baseId, documentId, compositeId, xmlLink)	0x2B	Idem ITU-T H.761.
removeLink (baseId, documentId, compositeId, linkId)	0x2C	Idem ITU-T H.761.

Tabela 34. Comandos de edição suportados no perfil DR.

## 6.6. Suporte a múltiplos dispositivos

O NCL permite a execução da aplicação de forma distribuída em múltiplos dispositivos (SOARES, 2009) que estejam acessíveis via rede. Exemplos de dispositivos incluem aparelhos de telefonia móvel, tablets, computador pessoal, central de mídia automotiva ou residencial, televisores conectados ou qualquer outro dispositivo que consiga estabelecer uma conexão com o receptor de rádio.

São definidas quatro classes de dispositivos: *active*, *passive*, *mediaCapture* e *onDemandMedia*, sendo que os dispositivos podem se cadastrar em uma ou mais classes. Na classe *active* os dispositivos devem suportar os reprodutores de mídia do Ginga e na passiva os dispositivos recebem somente imagens e áudio já decodificados. A classe *mediaCapture* define uma classe de dispositivos capazes de capturar conteúdo de mídia e enviar ao dispositivo pai, e a classe *onDemandMedia* permite que dispositivos capazes de detectar oferecimento de mídia na rede possam reproduzir conteúdos definidos nessa classe sob demanda. As duas últimas classes são novidades da NCL 3.1.

O suporte a múltiplos dispositivos é importante no rádio digital e permite novos cenários de interação para o rádio. Por exemplo, uma família que esteja viajando de carro e ouvindo um jogo de futebol pelo autorrádio. Nesse caso, todas as pessoas dentro do veículo, com exceção do motorista (por questão de segurança), poderão acessar informações relacionadas ao jogo de futebol em um celular, tablet ou segundas telas presentes no interior do veículo (atrás do encosto de cabeça dos bancos dianteiros, por exemplo). O aplicativo NCL pode especificar que os dispositivos acessem informações como melhores momentos, estatísticas do jogo e classificação do campeonato.

Em outro cenário, a emissora de rádio pode enviar um jogo de computador que pode ser jogado entre os passageiros do veículo. Em um cenário da zona rural, uma família que escuta o rádio em uma região sem conexão à internet poderá usufruir de diferentes tipos de mídia disponíveis para acesso sob demanda por aparelhos celular ou *tablet* que estejam pareados com o receptor de rádio via *bluetooth* ou qualquer outro tipo de rede sem fio.

É relevante notar que os receptores de rádio de ondas curtas e tropicais normalmente possuem uma antena telescópica (ver Figura 16) ou são utilizados com antena externa, assim como em automóveis, para melhor recepção. Essa característica torna o suporte a múltiplos dispositivos importante para permitir que equipamentos portáteis como um celular ou *tablet* possam ter acesso a informações provenientes do rádio. A utilização do Ginga no rádio digital poderá tornar o rádio uma excelente plataforma de difusão de conteúdo multimídia.

Devido à importância do suporte a múltiplos dispositivos no rádio digital, esse suporte está presente no perfil Digital Radio da NCL. No entanto, pelo fato do receptor de rádio comumente ter pouco poder de processamento, a classe

*passive* não faz sentido ser suportada no rádio digital, pois normalmente os dispositivos secundários tem igual ou maior processamento que o receptor. A classe *mediaCapture* também não tem um cenário de uso claro para o rádio e, portanto, também não é suportada.

Portanto, as classes *active* e *onDemandMedia* são suportadas no Ginga para rádio digital.



Figura 16. Exemplo de receptor de rádio com antena telescópica em região rural.

### 6.7. Resumo das características do middleware Ginga para rádio

As principais adições e diferenças do Ginga para rádio com relação ao middleware Ginga definido para TV são:

- Novo grupo de variáveis globais de nome *position*, que contém informações sobre a geolocalização do receptor. As variáveis desse novo grupo são: *latitude*, *longitude*, *altitude*, *accuracy*, *hasGeolocation*, *locationAvailable*, *geolocationActive*, *geolocationMinTime* e *geolocationMinDistance*;

- Não são suportados os seguintes comandos de edição: *addRegionBase*, *removeRegionBase*, *addDescriptor*, *removeDescriptor*, *addDescriptorSwitch*, *removeDescriptorSwitch*, *addDescriptorBase*, *removeDescriptorBase*, *addTransition*, *removeTransition*, *addTransitionBase*, *removeTransitionBase*, *addImportBase*, *removeImportBase*, *addImportedDocumentBase*, *removeImportedDocumentBase*, *addImportNCL* e *removeImportNCL*;
- Uma forma genérica para sinalização de bases temporais é utilizada, e uma nova base temporal de nome *Radio Time Base* (RTB) é definida;
- Introduzida uma nova entidade para carregar comando de edição, de nome *EditingCommandEvent*, é apresentada na Tabela 33, e possui uma sintaxe um pouco diferente do elemento com função semelhante na TV Digital;
- Tipos de mídia com maior compressão para imagem, BPG, e áudio, Opus, são suportadas;
- Novos tipos de mídia suportados para representação de gráfico vetorial e para sintetização de voz. Os padrões escolhidos foram o SVG, para gráfico vetorial, e SSML para sintetização de voz. Âncoras podem ser definidas através do uso do atributo *label* do elemento *<area>*. A Tabela 35 apresenta os tipos de mídia cujo suporte é requerido no perfil básico de receptor.

Categoria	Tipo de mídia	MIME type	Extensão
Imagem	PNG	image/png	png
	JPEG	image/jpeg	jpg, jpeg
	BPG	image/x-bpg	bpg
Áudio	MPEG-4 AAC HEv2	audio/mp4	mp4, mpg4
	Opus	audio/ogg	opus
Gráficos vetoriais	SVG Tiny 1.2	image/svg+xml	svg, svgz
Sintetização de voz	SSML 1.1	application/ssml+xml	ssml
Texto	Texto puro	text/plain	txt
Aplicação	ginga-NCL	application/x-ginga-NCL	ncl



	ginga-NCLua	application/x-ginga-NCLua	lua
	ncl-settings	application/x-ncl-settings	-
	ncl-time	applications/x-ncl-time	-

Tabela 35. Tipos de mídia a serem suportadas no perfil básico de receptores para rádio digital.

Além dos tipos especificados na Tabela 35, o tipo HTML poderá ser suportado, dependendo dos recursos computacionais que o receptor possuir. O tipo de mídia Journaline poderá ser suportado dependendo do sistema de rádio digital em uso, e o suporte a mídias de vídeo também poderá ser adicionado caso o receptor suporte. Um perfil com suporte a recursos de acessibilidade, como reconhecimento de voz, também poderá ser adicionado. Essas futuras definições serão importantes na composição dos perfis de receptores do Sistema Brasileiro de Rádio Digital, que dependem de vários fatores externos a este trabalho.

## 7 Implementação de referência do Ginga

Neste capítulo é apresentada a arquitetura e a estrutura da atual implementação de referência do Ginga, desenvolvida pela PUC-Rio e as modificações realizadas para o suporte ao perfil DR da NCL e aos novos recursos para rádio digital.

O Ginga é um middleware para aplicações interativas. Um middleware tem a função de prover suporte à execução de aplicações de forma independente de plataforma.

A Figura 17 apresenta a arquitetura do Ginga.

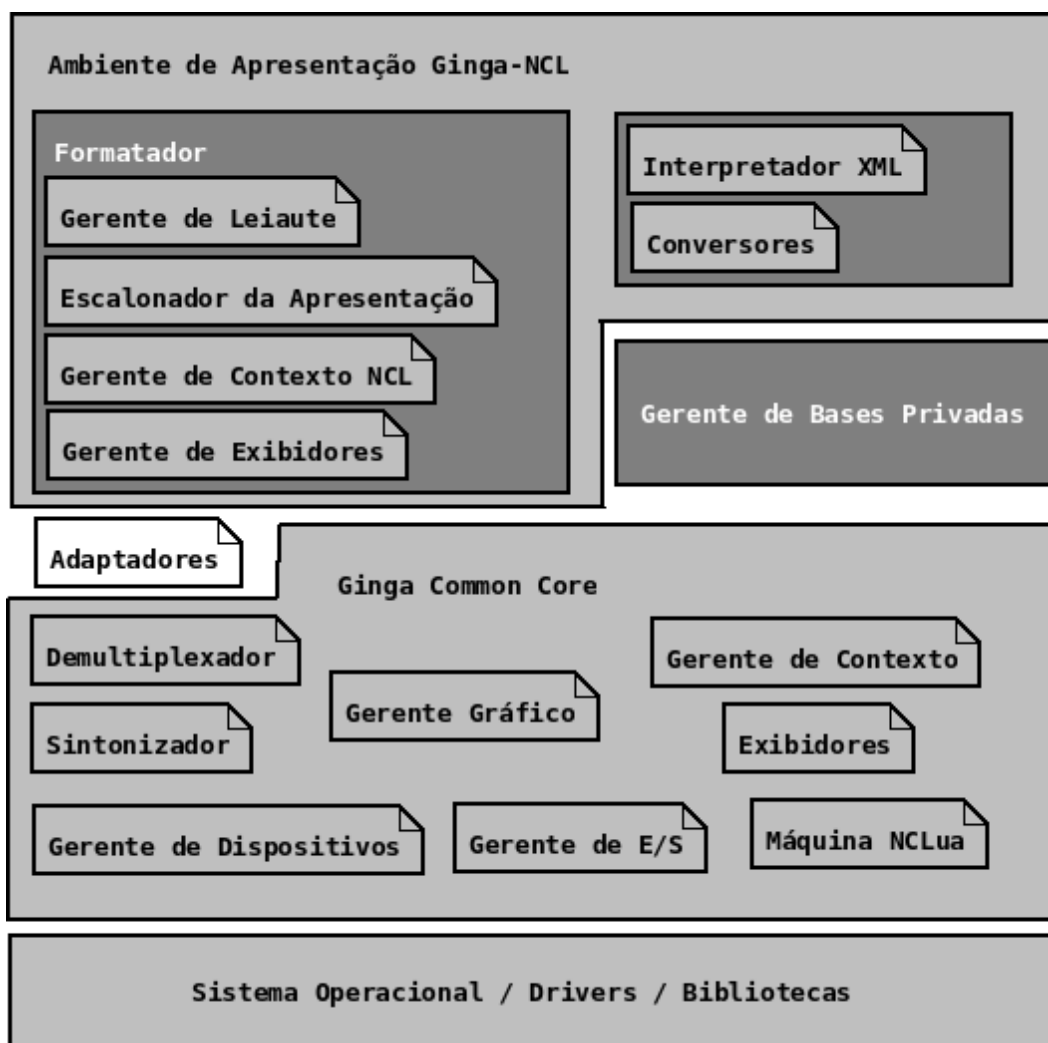


Figura 17. Componentes do Ginga.

A arquitetura do Ginga (MORENO, 2013a) é subdividida em três subsistemas, o Ginga Common Core (Ginga-CC), o Ginga-NCL e o Gerente de Bases Privadas.

O Ginga-CC é responsável pela interação com o hardware e bibliotecas de sistema do receptor. Serviços como decodificação de mídias, sintonização de canal, processamento dos dados provenientes de carrossel de dados, E/S e gerenciamento de dispositivos de vídeo são providos por esse subsistema.

Já o Ginga-NCL é o subsistema responsável por todo o ciclo de vida de uma aplicação NCL. A interpretação do código NCL e gerenciamento da apresentação definida pela aplicação são controlados por esse subsistema, que depende exclusivamente dos serviços providos pelo Ginga-CC.

Aplicações NCL são armazenadas em uma estrutura conhecida como base privada. O Gerente de Bases Privadas é responsável pela manutenção das Bases Privadas, sendo que ao menos uma base privada é aberta para cada emissora sintonizada. O Gerente de Bases privadas também tem a função de receber e executar ações provenientes de comandos de edição, além de permitir que aplicações em uma base privada possam ser iniciadas, pausadas, resumidas ou paradas.

A Figura 18 apresenta a arquitetura do Ginga com destaques em colorido para os módulos que necessitam modificações para prover suporte ao Rádio Digital.

Os componentes do subsistema de apresentação Ginga-NCL são o Formatador, Interpretador XML e Conversores.

O Interpretador XML tem a tarefa de analisar sintaticamente documentos NCL no perfil DR, verificar a conformidade do documento e montar uma representação interna da apresentação. Posteriormente um Conversor transforma essa representação interna, baseada no modelo NCM, em estruturas de dados apropriadas para a execução da apresentação pelo Formatador. Para o perfil DR tanto o interpretador XML quanto os conversores foram simplificados, diminuindo a chance da existência de bugs, em decorrência da menor quantidade de elementos no perfil da NCL para rádio digital.

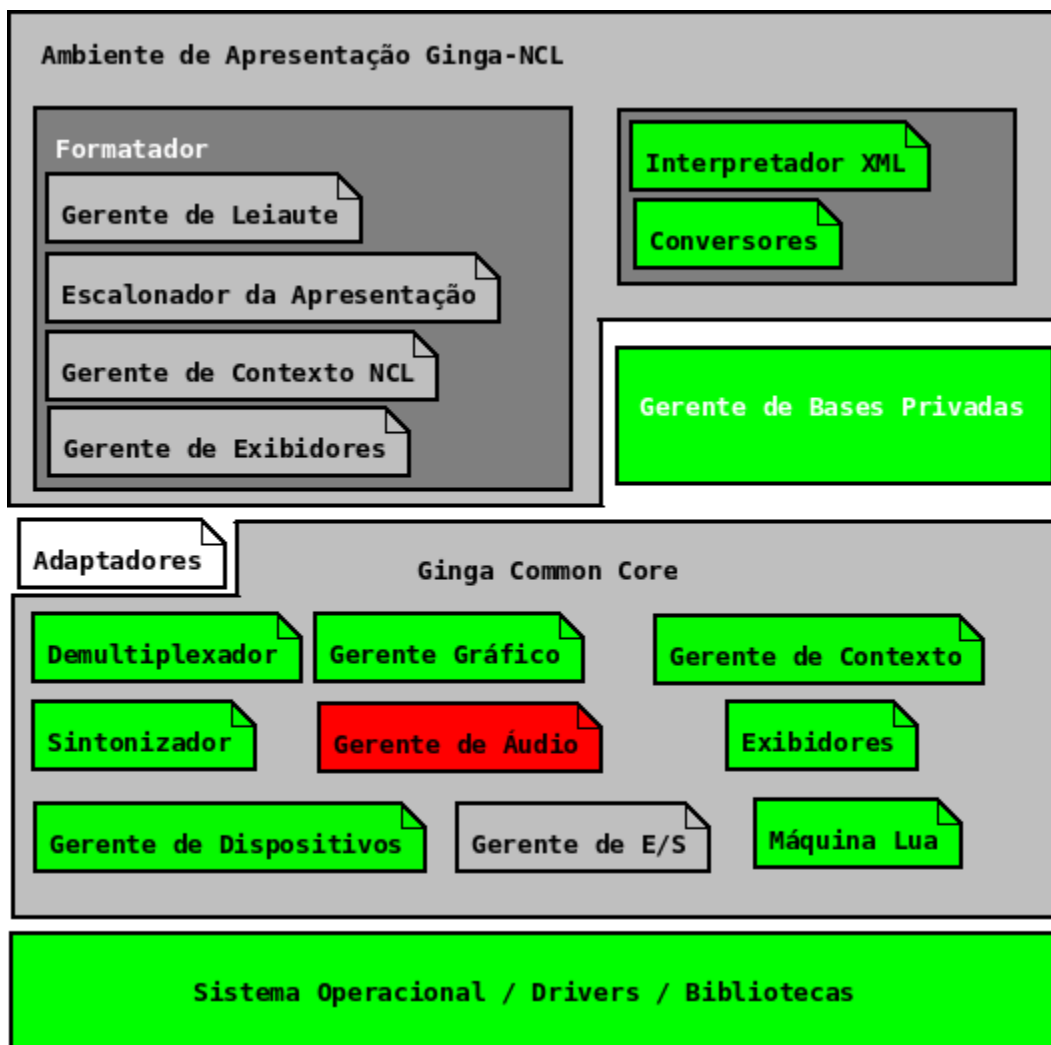


Figura 18. Arquitetura do Ginga com destaque aos módulos adaptados para o rádio digital em verde, e módulo novo em vermelho.

O Formatador é responsável por executar uma aplicação em uma base privada ativa a partir de uma estrutura de dados gerada por um Conversor. O Formatador é composto pelo Gerente de Leiaute, Escalonador de Apresentação, Gerente de Contexto NCL e Gerente de Exibidores. O Formatador NCL não necessita ser modificado para o contexto do rádio digital.

Pelo fato do perfil DR ter menos elementos que o EDTV, seria possível simplificar o modelo utilizado internamente pelo Ginga para a apresentação, no entanto, tal reescrita iria requerer uma completa refatoração da implementação de referência, fugindo do escopo do trabalho presente.

Aplicações armazenadas de forma persistente pelo Gerenciador de Bases Privadas podem ser iniciadas e pausadas pelo ouvinte através do Catálogo de Aplicações. É importante que o Gerente de Bases privadas sirva de forma harmônica tanto ao rádio digital como à TV digital em um receptor que suporte

ambos os sistemas, de forma que o Catálogo de Aplicações exiba aplicações provenientes de ambos meios.

Pelo fato do sistema de multiplexação do rádio digital ser diferente da TV Digital, tanto a sinalização do carrossel quanto a forma de envio de comandos de edição e a base temporal, o Gerente de Bases Privadas requereu adaptações para o rádio digital.

O Ginga Common Core é formado pelos seguintes módulos: Gerente de Dispositivos, Gerente de E/S, Sintonizador, Demultiplexador, Gerente Gráfico, Gerente de Áudio, Gerente de Contexto, Exibidores e Máquina NCLua.

O Gerente de Dispositivos é responsável pelo registro e comunicação de outros dispositivos com o receptor e permite a execução sincronizada de uma aplicação NCL em múltiplos dispositivos. Somente os módulos do middleware que provêm suporte às classes de dispositivos suportadas no perfil Digital Radio devem ser carregadas, sendo elas: *active* e *onDemandMedia*.

O Gerenciador de E/S é responsável pelo armazenamento e recuperação de aplicações e mídias da memória de armazenamento. Esse módulo não necessita modificação para o rádio digital.

Já o Sintonizador recebeu suporte para recepção de sinal no padrão Digital Radio Mondiale.

Trabalhando em conjunto com o Sintonizador, o Demultiplexador é o responsável por receber o sinal do Sintonizador e demultiplexá-lo em seus vários elementos, como os fluxos de áudio principais e os dados multimídia que compõem a aplicação NCL e suas mídias.

Foi adicionado à implementação de referência do Ginga o protocolo MOT (Multimedia Object Transfer) da forma como especificado na Seção 4.1, que descreve o transporte de aplicações NCL através do sistema Digital Radio Mondiale.

Outro módulo do Ginga-CC é o Gerente Gráfico, que para o rádio digital foi simplificado e apresenta somente um plano gráfico de apresentação, sendo que na TV digital mais de um plano gráfico é definido. O Gerente Gráfico é responsável pela renderização e composição dos elementos visuais de uma apresentação.

De forma análoga ao Gerente Gráfico, o Gerente de Áudio foi introduzido na arquitetura do Ginga para o rádio digital para prover um conjunto de rotinas

uniformes para o processamento de áudio no middleware. Na implementação de referência do Ginga os exibidores de áudio e vídeo interagem entre si de forma direta para compor o áudio da apresentação, sendo que para o rádio digital, um controle maior para o fluxo de áudio principal e para a mídia do tipo TTS (Text-To-Speech) tornaram a presença de Gerente de Áudio relevante.

O Gerente de Contexto é responsável pela manutenção das informações relacionadas ao usuário, ao sistema receptor e classes de dispositivos, permitindo adaptações na apresentação de acordo com o contexto. O Gerente de Contexto para o rádio digital passou a controlar também as variáveis que expõem a localização geográfica do receptor.

Para o rádio digital novos exibidores foram introduzidos, notadamente o exibidor do tipo TTS (Text-To-Speech), mídia visual vetorial e mídias para áudio e vídeo que apresentam grande otimização em ocupação de bits.

Para o caso da máquina NCLua, pequenas modificações foram feitas de forma a adequar a API, conforme definido na Seção 6.2.

Ao sistema operacional que executa o Ginga, foi adicionado drivers que suportam recepção de rádio digital.

A implementação de referência em C++ do Laboratório TeleMídia da PUC-Rio<sup>15</sup>, feita inicialmente para o perfil EDTV, foi modificada para suportar também os recursos do rádio digital. Uma descrição detalhada da implementação original é encontrada no Capítulo 4 da dissertação de mestrado de Moreno (MORENO, 2006).

Apesar da implementação ter evoluído desde então, e muitas bibliotecas terem sido substituídas por outras consideradas melhores, notadamente as bibliotecas que interfaceiam com dispositivos de vídeo e áudio e os decodificadores de mídia, pouco mudou na estrutura e organização do código.

Algumas opções de implementação para suporte ao perfil DR da NCL no Ginga são listadas a seguir:

- O middleware verifica em tempo de execução a origem do documento NCL. Se o mesmo chegou via TV, o perfil EDTV da NCL é assumido, se chegou via rádio, o perfil DR é assumido. Essa informação fica

---

<sup>15</sup> Código fonte do Ginga-NCL: <http://git.telemidia.puc-rio.br>

armazenada internamente para permitir ao Ginga prover o suporte adequado para cada meio;

- Os novos exibidores descritos no Capítulo 6 foram adicionados seguindo a arquitetura do Ginga, mas podem ser removidos em tempo de compilação.
- O Gerenciador de Contexto foi modificado para suportar as novas variáveis de ambiente e as variáveis relacionadas à geolocalização.
- O exibidor NCLua foi adequado às modificações descritas na Seção 6.2.
- Sintonizador foi modificado para suportar o sistema DRM.

O Exibidor SSML adicionado ao Ginga utiliza a biblioteca provida pelo projeto Espeak<sup>16</sup>, que é multiplataforma e de software livre. O suporte ao padrão SSML não é completo, mas para os testes realizados ele se mostrou completo o suficiente para todos os tipos de diálogo pensados nas aplicações de teste<sup>17</sup>.

Já o Exibidor SVG utiliza a implementação SVG provida pela biblioteca LibRsvg<sup>18</sup>. Essa biblioteca tem suporte completo aos elementos visuais estáticos da linguagem, sendo sua única deficiência a ausência do suporte ao elemento de animação da linguagem SVG.

O suporte ao tipo de imagem BPG (Better Portable Graphics) foi adicionado através do uso da biblioteca libbpg<sup>19</sup>.

Já o suporte ao formato de áudio Opus foi introduzido de forma trivial visto que a ffmpeg<sup>20</sup>, utilizada pelo *player* de áudio do Ginga, já possui suporte ao formato.

<sup>16</sup> eSpeak: Speech Synthesizer. Website: <http://espeak.sourceforge.net>

<sup>17</sup> Aplicações de teste: <http://www.telemidia.puc-rio.br/~rafaeldiniz/radio/>

<sup>18</sup> LibRsvg: A high performance SVG rendering library.  
<https://wiki.gnome.org/action/show/Projects/LibRsvg>

<sup>19</sup> Site do projeto: <http://bellard.org/bpg/>

Código fonte utilizado, com pequenas modificações: <https://github.com/rafael2k/libbpg>

<sup>20</sup> FFmpeg multimedia framework: <http://www.ffmpeg.org>





## 8

## Demonstração de transmissão, recepção e execução de aplicação NCL

Com o objetivo de validar o uso do Ginga no contexto da radiodifusão digital, foram realizadas provas de conceito envolvendo a transmissão, recepção e execução de aplicações NCL via rádio digital.

O sistema de rádio digital escolhido para o desenvolvimento das provas de conceito foi o sistema Digital Radio Mondiale, devido a suas normas serem abertas e ao fato de existirem implementações em software livre de grande parte do padrão.

Com o objetivo de documentar e incentivar o uso do Ginga no rádio digital, uma página web<sup>21</sup> foi criada com explicações de como transmitir e receber um sinal DRM com uma aplicação NCL embutida.

A cadeia de transmissão de rádio digital é composta elementos apresentados na Figura 19.

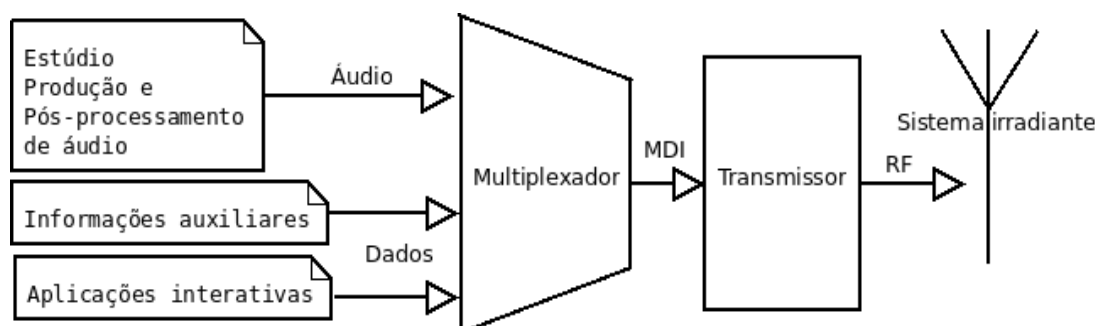


Figura 19. Elementos que compõem a transmissão de rádio digital.

À esquerda, na Figura 19, temos a caixa que representa o estúdio de uma emissora de rádio, que é composta por equipamentos de reprodução, captura e mixagem de áudio, além de equipamentos de pós-processamento como compressor e limitador. Os sinais de áudio, normalmente estéreo, que podem ser mais de um no caso de uma emissora operando com multiprogramação, são então

<sup>21</sup> Informações sobre o Ginga no DRM: <http://www.telemidia.puc-rio.br/~rafaeldiniz/radio/>

enviados ao multiplexador através de AoIP (Audio over IP), cabos XLR balanceados ou, em emissoras mais simples, cabos RCA não balanceados.

Logo abaixo temos uma caixa que representa informações auxiliares que são inseridas no sinal, tais como nome e indicativo da emissora, nome, língua e tipo do programa e também o horário da região da emissora.

A terceira caixa de cima para baixo, à esquerda, representa o repositório com aplicações interativas NCL.

O áudio, informações auxiliares e aplicações interativas são então multiplexados por um elemento conhecido no ecossistema do DRM como *Content Server*, que gera um fluxo de nome MDI (Multiplex Distribution Interface). Esse fluxo MDI é então enviado, normalmente via rede IP, até o transmissor.

O transmissor tem a função receber o fluxo MDI e a partir das informações desse fluxo ajustar os parâmetros de transmissão e gerar o sinal OFDM a partir das informações que devem ser transmitidas. Posteriormente, esse sinal é modulado no canal da emissora, amplificado e enviado para o sistema irradiante, através de uma linha de transmissão de RF (Rádio Frequência).

Finalmente, o sistema irradiante, composto por uma ou mais antenas, emite o sinal.

Para a transmissão de uma aplicação NCL, portanto, o multiplexador deve ser configurado de acordo com as definições apresentadas na Seção 4.1.

O software com o papel de *Content Server* utilizado nas provas de conceito foi o *drmcs*<sup>22</sup>, desenvolvido pela BBC (British Broadcasting Corporation). O código fonte do *drmcs* foi liberado como software livre após ser requerido para esse projeto, e teve algumas modificações adicionadas diretamente ao repositório oficial da aplicação.

Para a recepção e execução de uma aplicação NCL embutida em um sinal DRM o receptor deve saber identificar a aplicação e executá-la corretamente, de acordo com a Seção 4.1. O receptor escolhido foi o Dream (KURPIERS, 2003), que foi modificado para executar aplicações NCL. As modificações já foram

---

<sup>22</sup>Free software BBC DRM Content Server:  
<http://sourceforge.net/p/drm/code/HEAD/tree/drmcs/>

contribuídas para o projeto<sup>23</sup> e a implementação de referência do Ginga da PUC-Rio é utilizada para executar a aplicação.

Com esses dois softwares, o *drmcs* e o Dream, foi possível fazer o teste necessário para a validação do Ginga no sistema DRM. O primeiro gera o fluxo MDI, e o segundo, que suporta ler e decodificar o fluxo MDI, reproduz o conteúdo. A Listagem 19 apresenta um trecho do arquivo de configuração do *mdigen*, parte do *drmcs*, com configurações válidas para multiplexação de uma aplicação NCL.

```
<component id="ginga" xsi:type="data_packet" implementor="MOT">
  <private>
    <directory_mode>1</directory_mode>
    <always_send_mime_type>0</always_send_mime_type>
    <use_crc>1</use_crc>
    <send_compressed_directory>0</send_compressed_directory>
    <profiles>
      <profile>
        <id>1</id>
        <index>main.ncl</index>
      </profile>
    </profiles>
  </private>
  <encoder_id>ginga_enc</encoder_id>
  <source_selector>/home/rafael2k/drmcs/ginga</source_selector>
  <application_domain>0</application_domain>
  <application_data>0001</application_data>
  <data_unit_indicator>1</data_unit_indicator>
  <target_bitrate>5000</target_bitrate>
</component>
```

Listagem 19. Trecho do arquivo de configuração do *drmcs* que contém as informações válidas para multiplexação de uma aplicação NCL.

O fluxo MDI pode ser gerado e enviado via IP para o transmissor, ou gravado em arquivo. Quando o mesmo é gravado em arquivo, o seguinte comando é utilizado para iniciar o Dream no modo de decodificador MDI encapsulado no formato PCAP<sup>24</sup> (suportado pelo *drmcs*):

```
dream -f arquivo_mdi.pcap
```

O receptor Dream recebendo uma aplicação NCL é apresentado na Figura 20.

<sup>23</sup> Repositório com o código fonte do Dream: <http://sourceforge.net/p/drm/code/HEAD/tree/dream/>

<sup>24</sup> LibPcap file format: <https://wiki.wireshark.org/Development/LibpcapFileFormat>

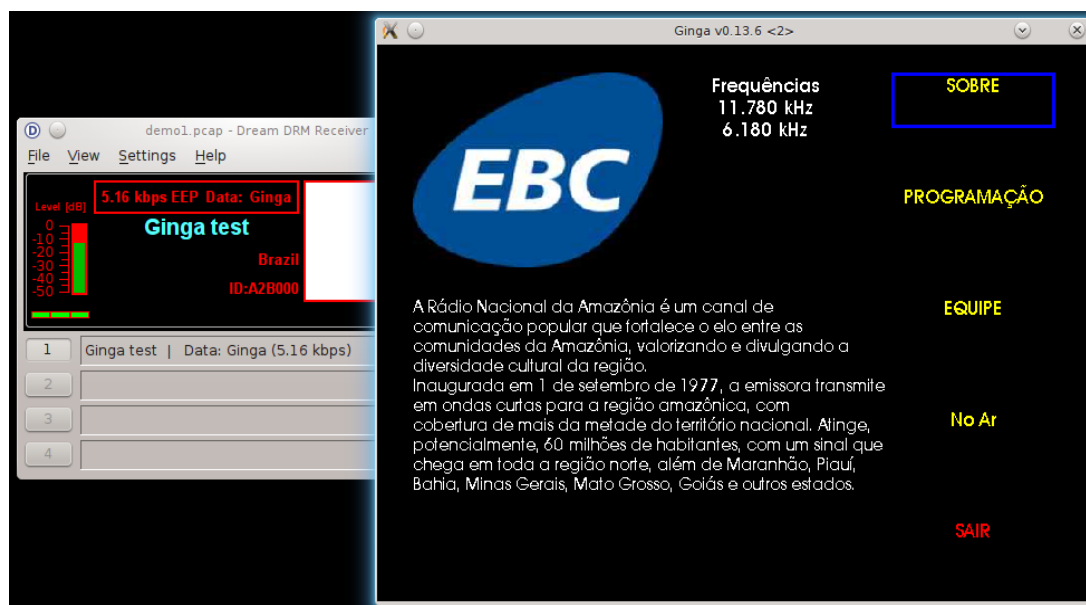


Figura 20. Receptor Dream recebendo uma aplicação NCL.

Dessa forma, é possível validar as partes relevantes da cadeia de transmissão DRM com relação à transmissão, recepção e execução de uma aplicação NCL. Alguns exemplos de fluxos MDI com aplicações NCL estão disponíveis para download<sup>25</sup>. A Figura 21 apresenta o esquema de teste utilizando somente o *drmcs* e o *Dream*.

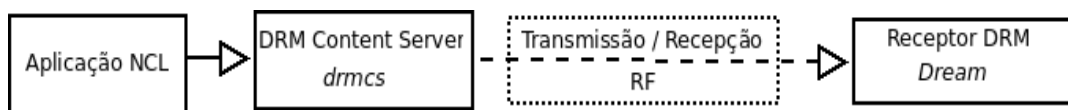


Figura 21. Esquema de teste que envolve os elementos relevantes (em linha contínua) da cadeia de transmissão/recepção DRM para aplicações NCL.

Nesse caso de teste de aplicação para o Ginga no DRM, os estágios de modulação e demodulação da cadeia de transmissão/recepção do rádio digital foram pulados, visto que não precisamos validar as camadas de RF do sistema.

No entanto, para efeito de demonstrações públicas, a transmissão pelo ar é essencial para a aceitação de pessoas leigas. Por esse motivo, foram montados dois arranjos para a transmissão e recepção DRM pelo espectro.

<sup>25</sup> Site com exemplos de fluxos MDI com aplicação Ginga: <http://www.telemidia.puc-rio.br/~rafaeldiniz/radio/>

Ambos os arranjos são baseados em arquitetura SDR (Software Defined Radio). O primeiro é composto pela Ettus USRP<sup>26</sup> (Universal Software Radio Peripheral), responsável pelas tarefas de conversor D/A e DUC (Digital Up-Converter) e o software Spark<sup>27</sup>, compatível com a USRP, responsável pela geração do sinal OFDM do DRM. A USRP se conecta via porta USB a um computador que executa o Spark. A Figura 22 apresenta uma foto da USRP utilizada.



Figura 22. Foto da USRP.

A segunda solução de transmissão testada foi provida pela NTi<sup>28</sup>, composta pelo USB DAC 48, que faz o papel de conversor D/A e o Diragen30, que faz conversão do sinal em banda base para o canal da emissora (Up-Converter). O software Spark também é utilizado nesse esquema de transmissão. A Figura 23 apresenta uma imagem com o USB DAC 48 e DiRaGen 30.

<sup>26</sup> Ettus Research USRP: <http://www.ettus.com/product/details/USRPPKG>

<sup>27</sup> Spark realtime modulator software for Digital Radio Mondiale: <https://www.drm-sender.de/>

<sup>28</sup> Rudolf Ille Nachrichtentechnik: <http://www.nti-online.de/edrg30.htm>

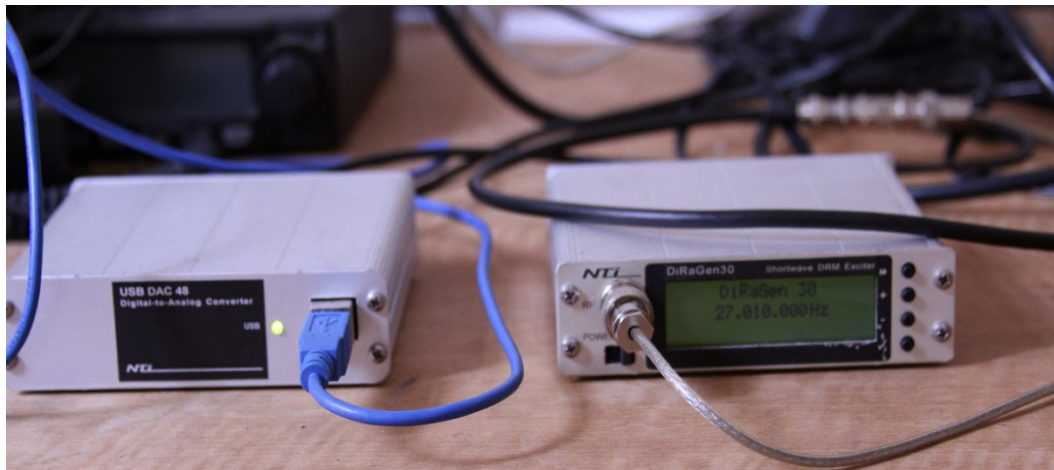


Figura 23. Foto dos equipamentos de transmissão DRM da NTi.

O software Spark possui suporte para a transmissão de um fluxo MDI recebido via IP ou gravado, e ainda suporta um modo completo, que inclui o Content Server, além do modulador. Também foi utilizado nos testes um pré-amplificador para elevar o sinal de saída para aproximadamente 100mW de potência, suficiente para prover uma cobertura de uma centena de metros, quando esse sinal é conectado a uma antena de ganho unitário. A Figura 24 apresenta o ambiente de teste de transmissão.

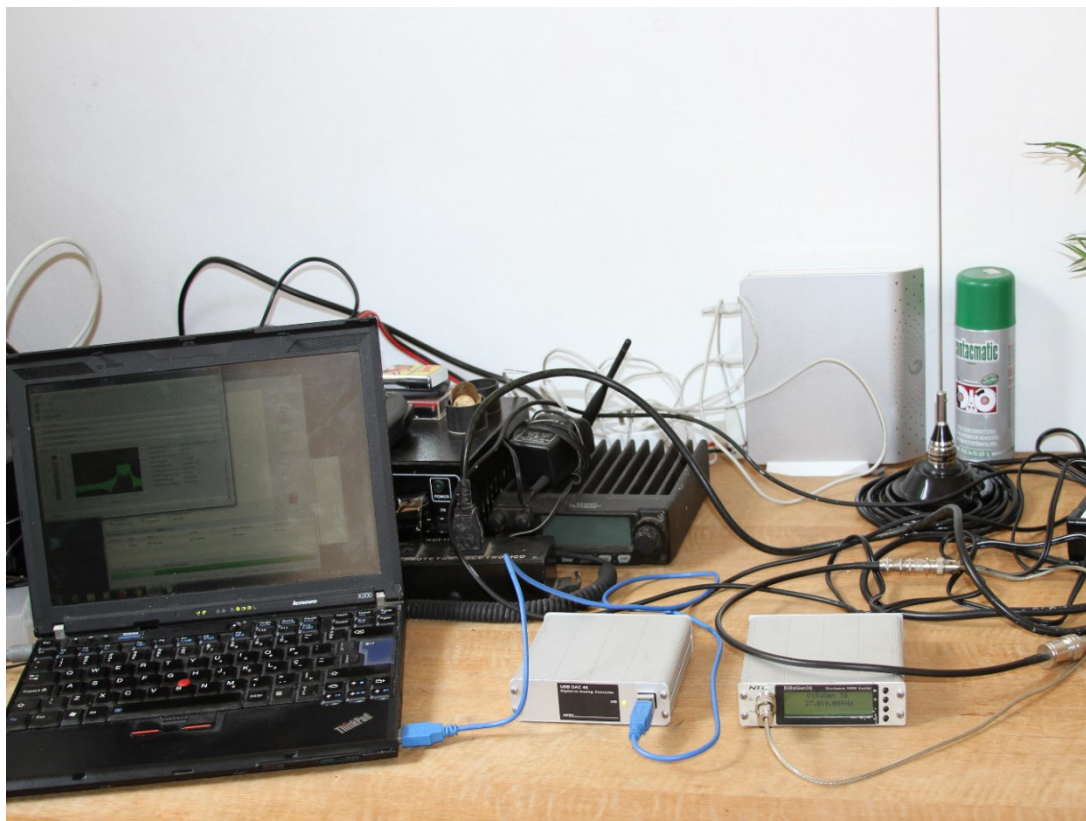


Figura 24. Ambiente de teste para transmissão DRM. Da esquerda para direita: computador executando o software Spark, USB DAC 48, DiRaGen30 e antena.

Para a recepção foi utilizado um receptor de rádio SDR de nome *FUNcube Dongle Pro+*<sup>29</sup>, que se conecta à porta USB de um computador, no qual deve ser executado o software Dream. A Figura 25 contém uma foto do receptor.



Figura 25. Foto do receptor FUNcube Dongle Pro+.

A primeira demonstração pública que envolveu a transmissão e a recepção de um sinal de rádio digital com o Ginga ocorreu durante a Conferência Internacional Espectro, Sociedade e Comunicação II, que ocorreu na PUC-Rio em novembro de 2013<sup>30</sup>.

<sup>29</sup> FUNcube Dongle Pro+: <http://www.funcubedongle.com/>

<sup>30</sup> Conferência Internacional ESC II: <http://www.conferences.telemidia.puc-rio.br/esc2013/>



## 9 Conclusão

A interatividade na radiodifusão é um elemento de grande relevância para o futuro das comunicações no Brasil e na América Latina. Sem ela, o rádio e a TV tendem a ficar desconectados dos avanços tecnológicos dos sistemas multimídia e hipermídia desenvolvidos para a Internet.

O Ginga permite a convergência entre rádio, TV e outros meios. Com o Ginga as vantagens da radiodifusão sobre outros meios, tais como a abrangência completa da população brasileira, acesso gratuito e a ubiquidade dos receptores podem ser exploradas de forma ampla.

Este trabalho apresenta uma análise da radiodifusão brasileira e a partir dela extrai alguns requisitos para a interatividade no rádio digital. Esses requisitos são atendidos com o sistema proposto. São considerados aspectos relacionados à autoria da aplicação NCL, aspectos da linguagem em si e a forma de transmissão nos dois padrões de rádio digital em consideração pelo país, o DRM e o HD Radio.

De forma a atender os requisitos, são apresentados um perfil da linguagem NCL e uma implementação do middleware Ginga para aplicações interativas no rádio digital. Tal perfil da linguagem NCL é denominado DR (Digital Radio). Esse perfil, ao mesmo tempo que permite a autoria de aplicações NCL de forma não mais complicada que no perfil EDTV, possui menos elementos e pode levar a uma implementação do Ginga mais simples e com menor possibilidade de erros de *software (bugs)*.

Para a camada de multiplexação dos sistemas de rádio digital foram adicionados recursos antes inexistentes, tais como o envio de uma base temporal e comandos de edição. Foi definida uma nova base temporal de nome Radio Time Base (RTB), que permite a sincronização fina do fluxo principal de áudio com ações da aplicação NCL.

As definições contidas nos capítulos desta dissertação permitem que o rádio digital brasileiro com o Ginga seja uma plataforma IBB (*Integrated broadcast-*



*broadband*). De acordo com a pesquisa realizada, o rádio digital brasileiro, caso adote o Ginga, será o primeiro sistema de radiodifusão digital de banda estreita aberto aderente ao conceito IBB (*Integrated Broadcast-Broadband*). Outro sistema de rádio digital, de banda mais larga, que compartilha o mesmo *middleware* com a TV digital e é compatível com IBB é o ISDB-Tsb, utilizado no Japão.

Para a plenitude no uso de elementos de mídia em aplicações interativas para rádio digital, foi demonstrada a implementação de referência do Ginga adicionada de novos exibidores de mídia, com vantagens para o contexto do rádio digital, como os exibidores SVG (imagem vetorial), SSML (leitor de textos), BPG (imagem) e Opus (áudio). Esses formatos de mídia permitem um melhor aproveitamento do bitrate disponível no rádio.

Testes em laboratório foram realizados e comprovaram a viabilidade da introdução da interatividade com o Ginga no rádio digital. Todos os softwares desenvolvidos, tanto para a transmissão como para a recepção de aplicações NCL pelo rádio digital foram publicados como software livre.

As definições de multiplexação de aplicações NCL no sistema DRM foram submetidas oficialmente ao Consórcio DRM e os identificadores para aplicação NCL serão listados nas normas internacionais do padrão.

Tanto a indústria brasileira como os radiodifusores poderão utilizar as implementações abertas que foram desenvolvidas para produzir e adaptar transmissores e receptores compatíveis com o Ginga.

Testes em campo em parceria com a EBC (Empresa Brasil de Comunicação) estão em curso e demonstrarão para uma ampla audiência os benefícios da interatividade no rádio. No final de maio de 2015, a EBC iniciou provas com o sistema DRM na faixa de Ondas Curtas, com o sistema Digital Radio Mondiale. Os testes em campo visam explorar o potencial social da interatividade no rádio principalmente junto de populações que vivem longe de grandes centros urbanos e tem o rádio em Ondas Curtas como maior fonte de informação.

As definições da NCL DR e do Ginga podem ser incorporadas ao Sistema Brasileiro de Rádio Digital assim que o modelo de referência de sistema for definido, seja ele o HD Radio ou o DRM.

Considera-se, portanto, que com este trabalho as portas se abram para a implementação do Sistema Brasileiro de Rádio Digital com todas as potencialidades que o rádio digital com o Ginga permite.

## 10

## Referências Bibliográficas

- AAS, J.; EGGE, N.; BOSSEN, F. **Lossy Compressed Image Formats Study**. Mozilla Corporation, July 2014. Disponível em: <[http://people.mozilla.org/~josh/lossy\\_compressed\\_image\\_study\\_july\\_2014/](http://people.mozilla.org/~josh/lossy_compressed_image_study_july_2014/)>. Acesso em Janeiro 2015.
- ABNT. **ABNT NBR 15606-1:2015**. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiofusão digital - Parte 1: Codificação de dados. 2015.
- ANDERSON, J. N. **Radio's digital dilemma: broadcasting in the 21st century**. Doctoral dissertation, University of Illinois at Urbana-Champaign. 2011.
- ATSC. **ATSC-Mobile DTV Standard A/153 Part 5**. Application Framework. 2009.
- BELLARD, F. **BPG Specification**. 2015. Disponível em: <[http://bellard.org/bpg/bpg\\_spec.txt](http://bellard.org/bpg/bpg_spec.txt)>. Acesso em Janeiro 2015.
- BIANCO, N. R. D.; ESCH, C. E. **Rádio digital no Brasil: análise de um debate inacabado**. Revista Brasileira de Políticas de Comunicação 1, Nº 2. 2012.
- CASTRO, C. **Televisão digital e as possibilidades de acessibilidade audiovisual no Brasil**. Revista Esferas, n. 5, 2014.
- COSTA, H. **Portaria número 290 do Ministério das Comunicações**, Institui o Sistema Brasileiro de Rádio Digital - SBRD e dá outras providências. 30 de março de 2010.
- DAVIES, K. **Ionospheric radio**. Electromagnetic Waves Series, Nº 31, IET, 1990.
- DINIZ, R. **Requisitos para um rádio digital interativo no Brasil e América Latina**. Conferência Internacional Espectro, Sociedade e Comunicação II, 2013.
- DUFOURD, J. **LASER**: The lightweight rich media representation standard [Standards in a Nutshell]. Signal Processing Magazine, IEEE, v. 25, n. 6, p. 164-168, 2008.

ETSI. **ETSI TS 101 498-1** 2.1.1. Digital Audio Broadcasting (DAB); Broadcast website; Part 1: User application specification. 2006a.

ETSI. **ETSI EN 301 234** 2.1.1. Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol. 2006b.

ETSI. **ETSI TS 102 979** 1.1.1. Digital Audio Broadcasting (DAB); Journaline; User application specification. 2008.

ETSI. **ETSI TS 101 968** 1.3.1. Digital Radio Mondiale (DRM); Data applications directory. 2009.

ETSI. **ETSI ES 201 980** 4.1.1. Digital Radio Mondiale (DRM); System Specification. 2014.

ETSI. **ETSI TS 101 499** 3.1.1. Hybrid Digital Radio (DAB, DRM, RadioDNS); SlideShow; User Application Specification. 2015a.

ETSI. **ETSI TS 102 818** 3.1.1. Hybrid Digital Radio (DAB, DRM, RadioDNS); XML Specification for Service and Programme Information (SPI). 2015b.

ETSI. **ETSI TS 103 270** 1.1.1. RadioDNS Hybrid Radio; Hybrid lookup for radio services. 2015c.

FGV; ABERT. **Censo da Radiodifusão**. Publicado online: <<http://goo.gl/IisPjf>>. Acesso em Janeiro de 2015. 2008.

GORGEN, J. **Sistema central de mídia**: proposta de um modelo sobre os conglomerados e comunicação no Brasil. Dissertação de mestrado, UFRGS, 2009.

GORSACK, A.; HENDRIKS, T. **Bringing digital data services to life in North America**, Proceedings of the 11th Workshop Digital Broadcasting, Fraunhofer IIS, Erlangen, Germany. September 15-16, 2010.

HOEG, W.; Lauterbach, T. **Digital audio broadcasting: principles and applications of digital radio**. John Wiley & Sons, Third Edition. 2009.

HOFMANN, F.; HANSEN, C.; SCHAFER, W. **Digital radio mondiale (DRM) digital sound broadcasting in the AM bands**. IEEE Transactions on Broadcasting, v. 49, n. 3, p. 319-328, 2003.

IBGE, **Censo Demográfico de 2010**, Diário Oficial da União. 2010.

IETF, **IETF RFC 1952**. GZIP file format specification version 4.3, 1996.

IETF, **IETF RFC 5246**. The Transport Layer Security (TLS) Protocol version 1.2. 2008.

IETF. **IETF RFC 6716**. Definition of the Opus Audio Codec. 2012.

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

ISO. **ISO/IEC 23003-1**. Information technology -- MPEG audio technologies -- Part 1: MPEG Surround. 2007.

ISO. **ISO/IEC 14496-3**. Information technology -- Coding of audio-visual objects -- Part 3: Audio. 2009.

ISO. **ISO/IEC 23003-3**. Information technology -- MPEG audio technologies -- Part 3: Unified speech and audio coding. 2012.

ISO. **ISO/IEC 13818-1**. Information technology -- Generic coding of moving pictures and associated audio information -- Part 1: Systems. 2013.

ITU. Recommendation **ITU-R BS.1514-2**. System for digital sound broadcasting in the broadcasting bands below 30 MHz. 2011.

ITU. Recommendation **ITU-T H.761**. Nested context language (NCL) and Ginga-NCL. 2014a.

ITU. Recommendation **ITU-R BS.1114-8**. Systems for terrestrial digital sound broadcasting to vehicular, portable and fixed receivers in the frequency range 30-3 000 MHz, 2014b.

ITU. Recommendation **ITU-T H.265**. High efficiency video coding. 2014c.

ITU. Report **ITU-R BT-2267-2**. Integrated broadcast-broadband systems. 2014d.

KNELLER, H. **New Life for AM with Digital Transmission**. NAB Broadcast Engineering Conference, 2013.

KURPIERS, A.; FISCHER V. **Open-source implementation of a digital radio mondiale (DRM) receiver**, in 9th International IEE Conference on HF Radio Systems and Techniques, Bath, United Kingdom, June 2003.

LIMA, G. A. F.; SOARES, L. F. G.; NETO, C. S. S.; MORENO, M. F.; COSTA, R. R.; MORENO, M. F. **Towards the NCL Raw Profile**. In II Workshop de TV Digital Interativa (WTVDI)-Colocated with ACM WebMedia, vol. 10. 2010.

LIMA, G. A. F. **Eliminating Redundancies from the NCL EDTV Profile**. Master's thesis, Pontifical Catholic University of Rio de Janeiro (PUC-Rio). 2011.

LIMA, G. A. F.; SOARES, L. F. G.; AZEVEDO, R. G. A.; MORENO, M. F. **Reducing the Complexity of NCL Player Implementations**, In Proceedings of the 19th Brazilian symposium on Multimedia and the web (pp. 297-304). ACM. November 2013.

- MACARIO, G.; TORCHIANO, M.; VIOLANTE, M. **An in-vehicle infotainment software architecture based on google android**. IEEE International Symposium on Industrial Embedded Systems, SIES '09. 2009
- MARSHALL, P. D. **New Media Cultures**. Bloomsbury Academic, 2004.
- MAXSON, D. P. **The IBOC Handbook: Understanding HD Radio (TM) Technology**. CRC Press. 2007.
- MORENO, M. F. **Um middleware declarativo para sistemas de tv digital interativa**. Dissertação de Mestrado. Departamento de Informática, PUC-Rio. 2006.
- MORENO, M. F.; SOARES, L. F. G.; CERQUEIRA, R. **A Component Based Architecture for Ginga**. Conference on Software Engineering Research and Practice (SERP) p. 76-82. 2013a.
- MORENO, M. F.; SOARES, L. F. G. **Hierarchical control of focus and input events in hypermedia applications**. In Computing Conference (CLEI), 2013 XXXIX Latin American (pp. 1-10). IEEE. October, 2013b.
- MORENO, M. F.; SOARES, L. F. G. FERREIRA MORENO, Marcio; GOMES SOARES, Luiz Fernando. **Using Multiple Interleaved Time Bases in Hypermedia Synchronization**. In IEEE International Symposium on Multimedia (ISM). IEEE, p. 187-194, 2014.
- NRSC. In-band/on-channel Digital Radio Broadcasting Standard, **NRSC-5-C**, United States, April, 2011a.
- NRSC. **HD Radio Air Interface Design Description**. Advanced Application Services Transport. Rev. G. 2011b.
- PEJIC, B.; PEJIC, A; COVIC Z. **Uses of W3C's Geolocation API**. In Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on, pp. 319-322. IEEE, 2010.
- SET. **Revista da SET**, Dilma assina decreto de migração de AM para FM. pg. 26-30. Novembro de 2013.
- SHIRAISHI, S.; SAKUMA, D.; HASEYAMA, M.; KITAJIMA, H. **Bandwidth-efficient information-providing system for mobile phone users**. In Consumer Electronics, 2006. ICCE'06. 2006 Digest of Technical Papers. International Conference on, pp. 289-290. IEEE, 2006.

- SILVA, T. V. C. B. N.; COELHO, T. F. **Descriminalização das rádios comunitárias na construção dos direitos humanos**. ISSN: 1988-2629. Nº 12. Nueva Época. Diciembre-Febrero, 2013.
- SOARES, L. F. G.; RODRIGUES, R. **Nested Context Model 3.0: Part 1 – NCM Core**. Relatório Técnico, Laboratório TeleMídia, PUC-Rio, Brasil, 2005.
- SOARES, L. F. G.; COSTA, R. M.; MORENO, M.; MORENO, M. F. **Multiple exhibition devices in DTV systems**. In Proceedings of the 17th ACM international conference on Multimedia (pp. 281-290). ACM. October, 2009.
- SOARES, L. F. G.; MORENO, M. F.; SOARES NETO, C. S. **Ginga-NCL: declarative middleware for multimedia IPTV services**. Communications Magazine, IEEE 48.6, pp. 74-81. 2010.
- SOARES, L.F.G.; LIMA, G.F. **NCL Handbook**. Department of Informatics, PUC-Rio. ISSN 0103-9741. 2013. Disponível em: <http://handbook.ncl.org.br/>
- VAUGHAN-NICHOLS, S. J. **Will mobile computing's future be location, location, location?** IEEE Computer Magazine, 42, p. 14-17 (2009).
- W3C. W3C Recommendation. **Scalable Vector Graphics (SVG) Tiny 1.2 Specification**, 2008.
- W3C. W3C Recommendation. **Speech Synthesis Markup Language (SSML)**, Version 1.1, 2010.