



Universidade Federal de Juiz de Fora - UFJF

Instituto de Ciências Exatas – ICE

Departamento de Ciência da Computação - DCC

Aplicações para Criação de Conteúdo Multimídia

Fernando Neiva de Paiva

Juiz de Fora, MG

Junho, 2015

Aplicações para Criação de Conteúdo Multimídia

Fernando Neiva de Paiva

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharel em Ciência da Computação

Orientador: Prof. Dr. Marcelo Ferreira Moreno

Juiz de Fora, MG

Junho, 2015

Aplicações para Criação de Conteúdo Multimídia

Fernando Neiva de Paiva

MONOGRAFIA SUBMETIDADA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Prof. Dr. Marcelo Ferreira Moreno (Orientador)

Doutor em Informática - PUC-RJ

Prof. Dr. Eduardo Barrere

Doutor em Engenharia de Sistemas e Computação - COPPE/UFRJ

Prof. Dr. Romualdo Monteiro de Resende Costa

Doutor em Informática - PUC-RJ

Juiz de Fora, MG

Junho, 2015

AGRADECIMENTOS

A Deus por ter me dado força para perseverar até o fim, à minha família que mesmos nos momentos mais difíceis esteve ao meu lado me amparando e me apoiando, aos meus colegas da Guiando Telecom por terem me dado a oportunidade de aplicar meus conhecimentos e a minha futura esposa Marcela Lovisi, pela compreensão, pelo amor e pelo carinho sempre confortante e renovador.

Ao professor Marcelo F. Moreno, pela paciência e dedicação, sem as quais este trabalho não se realizaria e aos professores do Departamento de Ciência da Computação pelos ensinamentos durante todos estes anos.

SUMÁRIO

RESUMO	7
ABSTRACT	8
1 INTRODUÇÃO	9
1.1 Motivação	10
1.2 Objetivo	11
2 TRABALHOS RELACIONADOS	12
2.1 NCL Eclipse	12
2.2 LuaComp	13
2.3 Composer	13
2.4 ITV Suite Author	15
3 PROPOSTA DE INTERFACE	17
3.1 Produção Jornalística e Modelos de Representação	17
3.2 Anéis de Interatividade	18
3.3 Limitações da Propostas de Interface e Sugestão de Melhorias	19
4 IMPLEMENTAÇÃO	23
4.1 Tecnologias	23
4.2 Arquitetura	23
4.3 O Anel de Interatividade	24
4.4 Geração de Código	26
5 CONCLUSÃO	30

LISTA DE FIGURAS

FIGURA 1 – Visão estrutural do NCL Composer.....	14
FIGURA 2 – Visão Textual do NCL Composer.....	15
FIGURA 3 – Interface do ITV Suite Author.....	16
FIGURA 4 – Convenção para simbologia de fluxograma.....	17
FIGURA 5 – Representação dos anéis de interatividade (REIS, 2015).....	19
FIGURA 6 – Representação do menu de ação.....	20
FIGURA 7 – Interface para edição de atributos de mídia estática.....	21
FIGURA 8 – Representação do anel de mídia estática.....	21
FIGURA 9 – Representação do momento da mídia no anel de interatividade.....	22
FIGURA 10 – Representação do anel com marcação do final de mídia.....	22
FIGURA 11 – Pacotes da aplicação.....	24
FIGURA 12 – Algoritmo de criação dos traços do anel de interatividade.....	25
FIGURA 13 – Interface da aplicação com três nós de mídia.....	25
FIGURA 14 – Modelo padrão de código gerado.....	26
FIGURA 15 – Classe para representar uma mídia.....	27
FIGURA 16 – Exemplo de aplicação construída pela interface.....	28
FIGURA 17 – Código NCL gerado através da interface.....	29

RESUMO

A criação de conteúdo multimídia para TV digital atualmente está limitada aos usuários que possuem competência técnica em programação, pois as ferramentas disponíveis ainda não são capazes de realizar uma abstração completa deste conhecimento de forma a permitir ao usuário trabalhar no domínio do conteúdo que está sendo gerado.

Este trabalho busca apresentar uma interface com a perspectiva estrutural e temporal das mídias de uma aplicação Ginga-NCL e seus relacionamentos, de tal forma que seja possível, construir uma aplicação sem a necessidade de ter conhecimento sobre como codificar em NCL.

Palavras-chave: Ferramentas de Autoria, TV Digital, Interatividade, Customização, Anel de Interatividade

ABSTRACT

Currently, multimedia content designing is limited to users that have programming skills, because available tools at the moment don't perform a complete abstraction between programming and multimedia content in such a way users can think just about the content that is being generated.

This work presents a user interface with structural and temporal perspective of medias and their links in Ginga-NCL application, in such manner users can build an NCL application without knowledge about programming.

Keywords: Authoring Tools, Digital TV, Interactivity, Interactive Ring

1 INTRODUÇÃO

A televisão, como um artifício tecnológico, tende em seu processo evolutivo a se enquadrar nas necessidades do mundo moderno. Com o desenvolvimento da internet e o sucesso das suas características de interatividade, onde o usuário tem a possibilidade de não somente escolher o conteúdo que deseja ver, mas também comunicar-se com quem produz o conteúdo, é natural que uma analogia entre essa experiência seja feita com a experiência de se assistir televisão. Porém, a idéia de interatividade na TV se antecede ao surgimento da internet, assim como destacado em Lima (2011), onde desde a década de 60 a programação era estabelecida de acordo com as pesquisas do Ibope.

Com a alteração do padrão de sinal analógico para o padrão digital de transmissão de TV, foi possível também transmitir dados multiplexados juntamente com o sinal audiovisual, oferecendo assim uma comunicação mais rica entre transmissor e telespectador.

Esse avanço tecnológico permite que sejam criadas aplicações para que a experiência de assistir TV seja interativa, isto é, para que o telespectador tenha a possibilidade de interferir no conteúdo que está sendo transmitido.

Com esta mudança tecnológica surgem novas demandas, as quais possuem cunho altamente multidisciplinar (MONTEZ; BECKER, 2004), pois envolvem pelo menos conhecimentos técnicos da área de computação e engenharia e também conhecimentos sociais de áreas como comunicação e publicidade.

Sendo assim, as ferramentas para produção de conteúdo multimídia interativo devem suportar os diferentes perfis de usuário e devem evoluir à medida que o conhecimento evolui.

Esta pesquisa visa modelar uma forma criação de conteúdo multimídia, com foco em TV Digital, para atender profissionais que não possuem conhecimentos técnicos em computação ou engenharia. Ou seja, deseja-se adaptar os métodos de autoria ao domínio do conteúdo que está sendo criado. Apesar do foco em TV Digital e no usuário sem conhecimento em programação, a proposta deste trabalho pode beneficiar outros meios de distribuição de conteúdo interativo, bem como outros perfis de autores de conteúdo multimídia.

1.1 Motivação

Com a aprovação do padrão brasileiro de TV Digital em 2007 e a realização de grandes eventos esportivos mundiais, tais como a copa do mundo de 2014 e as Olimpíadas em 2016, os estudos sobre TV Digital receberam destaque nacional.

Nesta área de pesquisa um dos grandes desafios está em como viabilizar que conteúdo multimídia com qualidade seja produzido sem que se tenha necessidade dos criadores serem profundos conhecedores de linguagens de programação e tecnologias de desenvolvimento de software. No auxílio para transpor estas dificuldades se apresentam as chamadas ferramentas de autoria.

Ferramentas de autoria são aplicações que não exigem competência técnica em desenvolvimento de software por parte do autor para que este produza algum tipo de conteúdo, atualmente, as ferramentas disponíveis para desenvolver aplicações para TV digital interativa não abrangem esta característica em sua totalidade, isto é, ao menos em algum momento das etapas de criação o autor precisa de conhecimentos técnicos de programação ou áreas afins para seguir em frente, além disto o público potencial de produtores de conteúdo é muito diversificado, sendo assim, apenas um modo de desenvolvimento pode ser ineficiente para uma grande parte de usuários. Portanto, não só as ferramentas de autoria devem fornecer um ambiente menos tecnicista, como também deve ser flexível para atender a diferentes perfis de usuários.

No atual estágio de desenvolvimento de aplicações hipermídia para TV digital pode se fazer um comparativo com o que se tinha no início da web, onde as informações eram concebidas primordialmente por pessoas que possuíam *expertise* sobre a tecnologia e não por pessoas especialistas no domínio do conteúdo. A web veio a se popularizar quando usuários comuns passaram a serem capazes de criar suas próprias aplicações. (LIMA, 2011)

Sendo assim, melhorar o modo de interação com as ferramentas de autoria de tal forma que usuários como jornalistas, editores, designers e publicitários tenham contato com uma ferramenta em um nível de abstração mais próximo do produto final pode fornecer um impulso a mais na produção de conteúdo multimídia para TV digital e agir como catalizador para popularização dos recentes recursos multimídias disponíveis para TV digital.

1.2 Objetivo

O principal objetivo deste trabalho é elaborar conceitos que definam melhor a forma que o profissional de comunicação tem de interagir com ferramentas de autoria para desenvolvimento e edição de conteúdo multimídia de tal forma a permitir que usuários sem conhecimentos técnicos em programação possam criar conteúdo interativo para TV digital.

2 TRABALHOS RELACIONADOS

A primeira característica que pode ser destacada a respeito da TV digital é visual, a qualidade de imagem em relação a resolução e ao tamanho da imagem é significativamente superior ao que é possível obter com a TV analógica (MONTEZ; BECKER, 2004), além disto, é possível alcançar uma qualidade de áudio estéreo também muito superior.

Tais qualidades advêm, dentre outras características do sistema, devido a sua maior tolerância a ruídos e interferências. (MONTEZ; BECKER, 2004)

Porém, as mudanças propostas com a TV digital não somente contribuíram para melhorar pontos já existentes como também viabilizaram o surgimento de ideias inovadoras, segundo (SOUZA JÚNIOR, 2009) a real inovação da TV digital está associada interação do usuário com o conteúdo que está sendo veiculado, isso permite realizar comércio através da própria TV, isto é, o telespectador não ter que depender de acessar a página da internet do anunciante ou ter que se dirigir a uma loja para poder adquirir um produto anunciado, ou então capacidade de o governo oferecer serviços através da televisão, em fim tais recursos tornaram-se viáveis após a possibilidade de transmitir dados multiplexados com o sinal audiovisual. (MONTEZ; BECKER, 2004)

Para facilitar que estas novas tecnologias sejam usufruídas em seu total potencial existem as ferramentas de autoria que permitem que conteúdo seja criado de forma mais rápida, mais rápido pois segundo Stenning e Oberlander (1995) com uma ferramenta de autoria é possível abstrair do autor toda ou parte da complexidade representada pela utilização de uma linguagem de programação ao se desenvolver uma aplicação e também temos que ao utilizarmos uma abstração podemos encurtar etapas do raciocínio referencial. (SANTOS; GOMES; AZEVEDO; NETO; TEIXEIRA, 2010)

2.1 *NCL Eclipse*

Dentre diversas ferramentas para desenvolvimento de aplicações para TV digital temos o NCL Eclipse, um plugin para a IDE eclipse que tem por objetivo tornar a tarefa de construção textual de uma aplicação feita através de NCL mais rápida e eficiente. Porém tal ferramenta, como dito, possui característica de oferecer a apoio na construção do código, isto é, não faz nenhum tipo de abstração visual para afastas o autor da necessidade

de conhecimento em programação, portanto o autor precisa estar capacitado a construir este código[7].

2.2 *LuaComp*

O LuaComp é um ambiente de autoria feito em Java com objetivo de auxiliar autores com níveis variados de conhecimento em construção de aplicações procedurais ou imperativas de TV digital, esta ferramenta oferece três tipos de visões: textual, estrutural e leiaute.

Na visão de leiaute o autor pode movimentar elementos gráficos escolhidos de um menu ou paleta de componentes e alterar seus atributos, tais como: altura, cor, largura e posição dentre outros atributos básicos.

A visão estrutural fornece uma exibição das páginas e seus relacionamentos entre si, sendo assim esta visão estrutural é útil quando se tem um projeto que envolve uma grande quantidade de páginas.

E a visão textual permite o autor alterar o código NCL e NCLua da aplicação. Tal recurso é útil para profissionais familiarizados com programação, porém a ferramenta não oferece apoio no processo de codificação como auto-completar o código, por exemplo.[5]

2.3 *Composer*

Ferramenta de autoria construída pelo Laboratório TeleMídia da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) também com objetivo de que o autor não necessite de fortes conhecimentos em linguagens de programação para que seja capaz de desenvolver uma aplicação para TV digital com NCL.

O Composer faz uso das visões: leiaute, estrutural, textual e temporal, sendo que esta última tem por objetivo fornecer uma perspectiva que apresente os objetos de mídia sob um referencial de tempo dando a ideia de duração e linearidade. Nesta visão é possível editar características de interatividade e até mesmo simulá-las.

Uma das mais importantes características desta ferramenta é a sua possibilidade de extensão, isto é, é possível criar novas visões e integrar à ferramenta, portanto abrindo espaço para que novas formas de interagir sejam implementados no próprio ambiente.[1]

Na visão estrutural é possível visualizar quais mídias serão exibidas no projeto, quando há um relacionamento entre as mídias o *link* entre elas é representado graficamente através de uma seta, porém não é exibido neste diagrama em que momento das mídias ocorre o relacionamento entre elas, isto é, não é possível, por exemplo, perceber apenas observando o diagrama que uma mídia irá iniciar uma outra após decorrido 15 segundos do início da primeira.

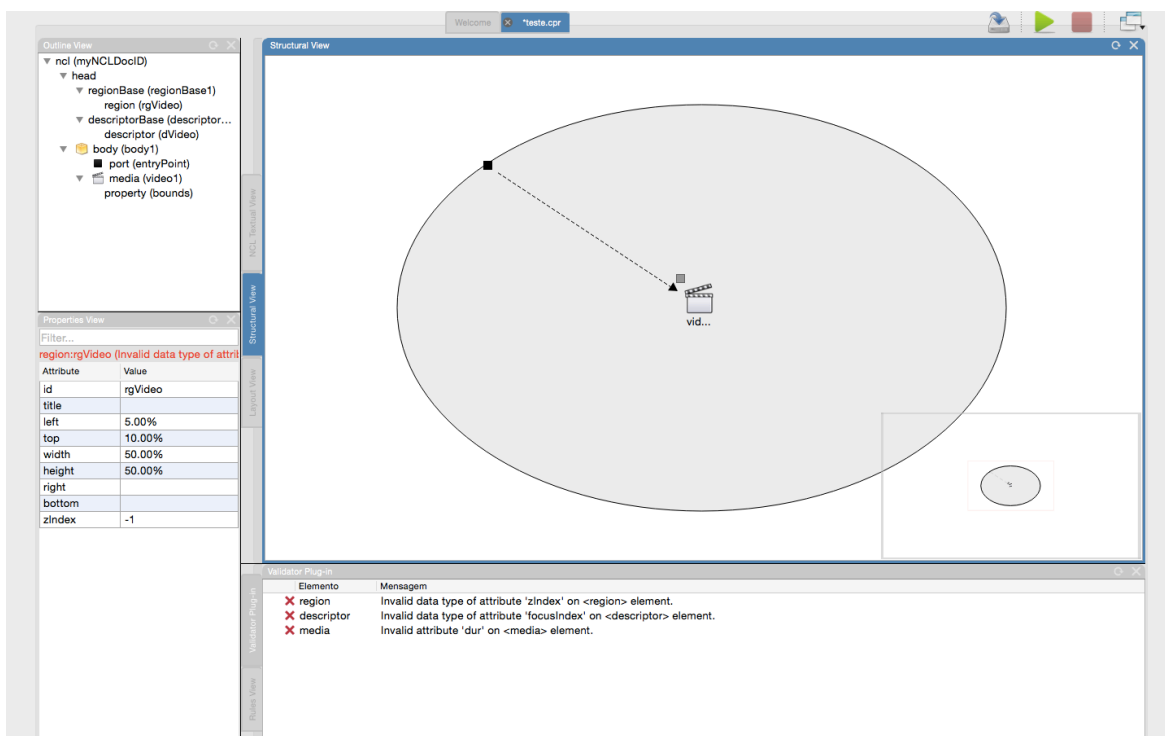


Figura 1 - Visão estrutural do NCL Composer

A visão textual do NCL Composer é aquela que mais exige conhecimento técnico por parte do usuário, pois nela se encontra o código NCL da aplicação. Para facilitar o entendimento do código por parte de um profissional, a interface colore, identifica o documento e coloca marcações com número de linha ao lado esquerdo. Também é válido destacar a visão de outline que apresenta uma visão geral do código de maneira mais agrupada.

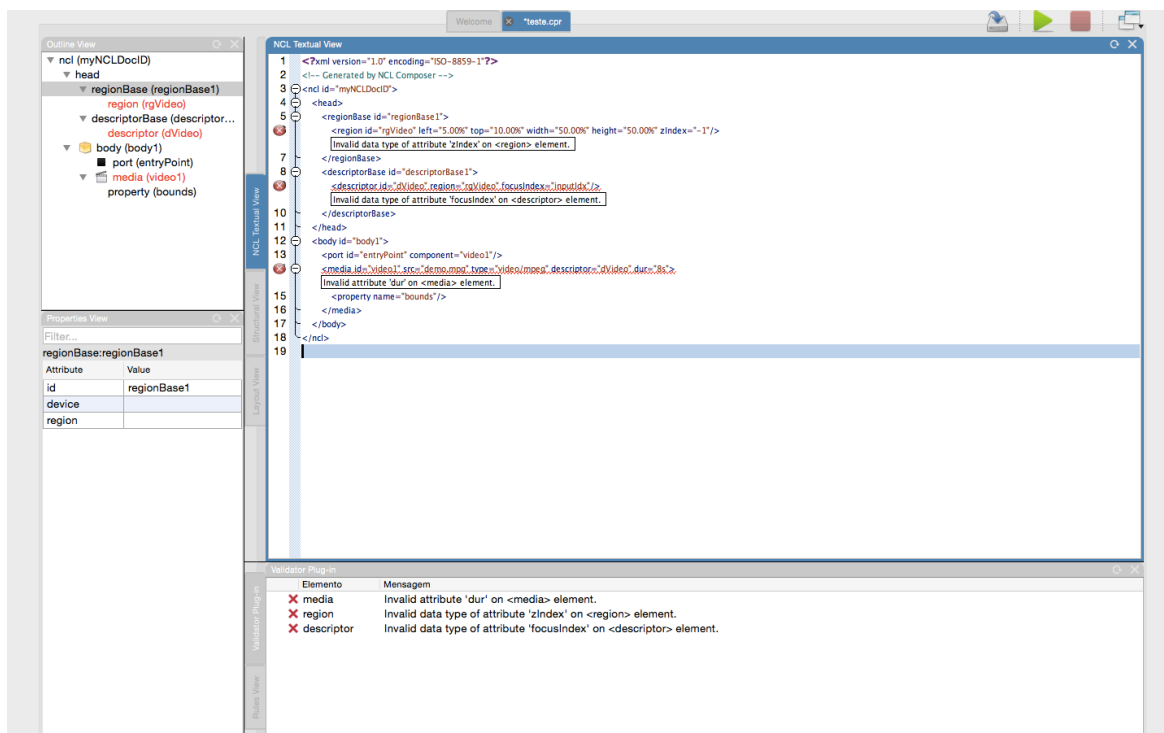


Figura 2 - Visão Textual do NCL Composer

2.4 ITV Suite Author

Ferramenta de autoria voltada para o middleware MHP (Multimedia Home Platform) do mercado europeu e possui sua abstração em forma de cenas, onde cada cena contém elementos que podem ser reaproveitados, com por exemplo o plano de fundo. A ferramenta apresenta um pré-visualizador que mostra em tempo de desenvolvimento as alterações feitas pelo autor e permite a simulação de interação com o usuário.

A parte central principal da interface gráfica exibe uma visão de *layout*, isto é, mostra a disposição das mídias na tela, mas a interface também conta com outras visões, para alterar propriedades dos elementos, listar os recursos utilizados no projeto, biblioteca de componentes, dentre outras.

O ITV Suite Author não conta com nenhuma visão que apresente uma visão estrutural dos elementos a qual seja possível visualizar um diagrama com as mídias utilizadas no projeto e o relacionamento entre elas.

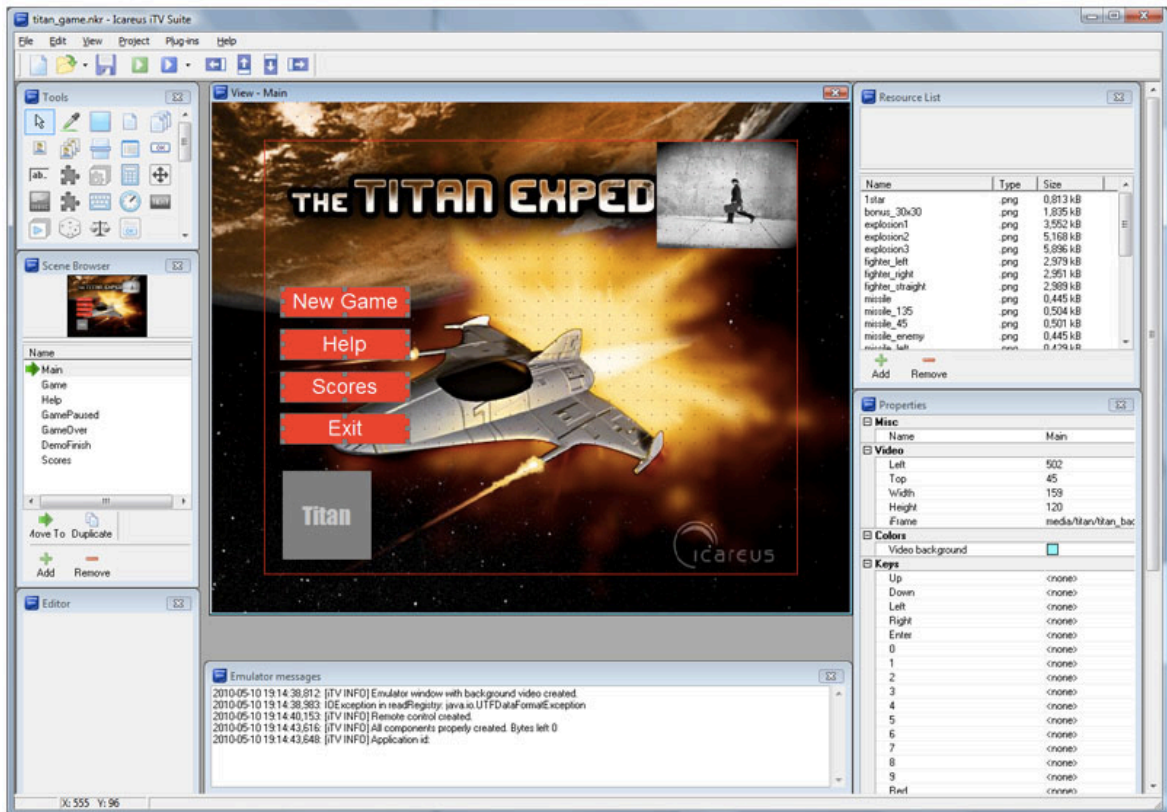


Figura 3 - Interface gráfica do ITV Suite Author

3 PROPOSTA DE INTERFACE

3.1 Produção Jornalística e Modelos de Representação

Com o objetivo de analisar a experiência dos profissionais de comunicação com modelos que supostamente possuem a abstração necessária para representar as etapas da produção jornalística em documentos de hipermídia, Reis (2014) realizou testes com alunos da disciplina de Técnica de Produção em Hipermídia, do curso de graduação em Jornalismo da Faculdade de Comunicação Social da Universidade Federal de Juiz de Fora e com turmas de pós-graduação em Jornalismo Multiplataforma da mesma universidade.

Os modelos utilizados foram diagramas inspirados no Modelo de Contextos Aninhados sugerido por Soares (2005) com uma simbologia própria convencionada por Reis (2014)..

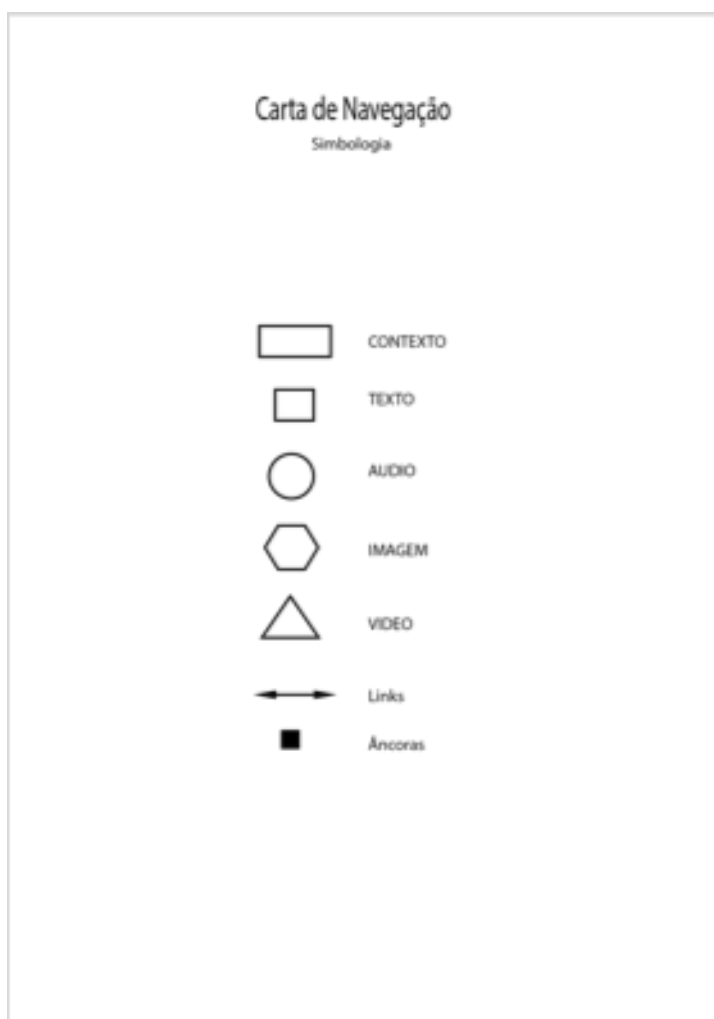


Figura 4 - Convenção para simbologia de fluxograma

Segundo Reis (2014), a respeito dos testes realizados, o modelo de contextos aninhados resolve, em grande parte, os problemas de organização e representação do projeto, porém ele é deficitário em alguns aspectos.

A primeira deficiência registrada por ele foi que os alunos tiveram dificuldades em trabalhar com o modelo, que segundo eles, aproximavam muito mais da matemática que da linguagem jornalística. Também foi destacado que o modelo não permite representar em qual momento poderia acontecer uma ação de interatividade, visto o indeterminismo da intervenção do usuário.

Baseado nestas análises foi feita uma nova proposta de modelo para representar a produção jornalística de um conteúdo multimídia interativo.

3.2 Anéis de Interatividade

Uma abordagem de interface para representação da visão estrutural e temporal de uma aplicação interativa para TV Digital foi apresentada por Reis (2014), onde um objeto de mídia é representado por uma ícone, que varia de acordo o tipo de mídia que está sendo representada, e em seu entorno há um conjunto de traços que representam o anel de interatividade.

Cada traço do anel de interatividade designa um momento da mídia contida neste anel e permite que seja criada uma interligação com outra mídia a partir daquele instante.

Na figura abaixo há a representação de quatro mídias: um objeto de mídia textual, um objeto de mídia de áudio, um objeto de mídia de imagem e outro objeto de mídia de vídeo. Os traços estendidos são aqueles onde se encontra uma âncora para realizar uma interligação com outra mídia.

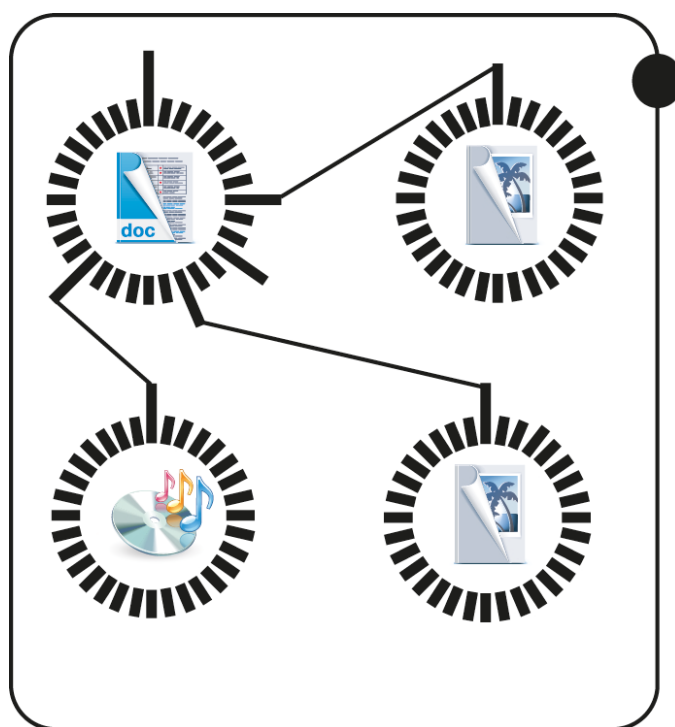


Figura 5 - Representação anéis de interatividade (REIS, 2014)

3.3 Limitações da Proposta de Interface e Sugestão de Melhorias

Apesar da abordagem de interface acima proposta simplificar a abstração dos conceitos da linguagem NCL para o jornalista, ela não aborda elementos fundamentais para que a partir do diagrama criado seja possível realizar uma transcrição direta daquilo que está representado na visão abstrata para um código NCL executável, além de não conter alguns aspectos que poderiam melhorar a experiência de uso de uma aplicação construída com esta abstração.

Uma das limitações da abordagem anterior é que não é possível discriminar o evento que uma interligação aciona na mídia de destino, por exemplo, a âncora na mídia de destino ao ser acionada poderia estar sendo parada ou iniciada, de acordo com o conector escolhido, porém não foi apresentada uma forma de o usuário fazer esta escolha.

Para resolver este problema uma proposta é a utilização de um menu em que o usuário ao completar um ligação escolhe qual ação será disparada na mídia de destino.

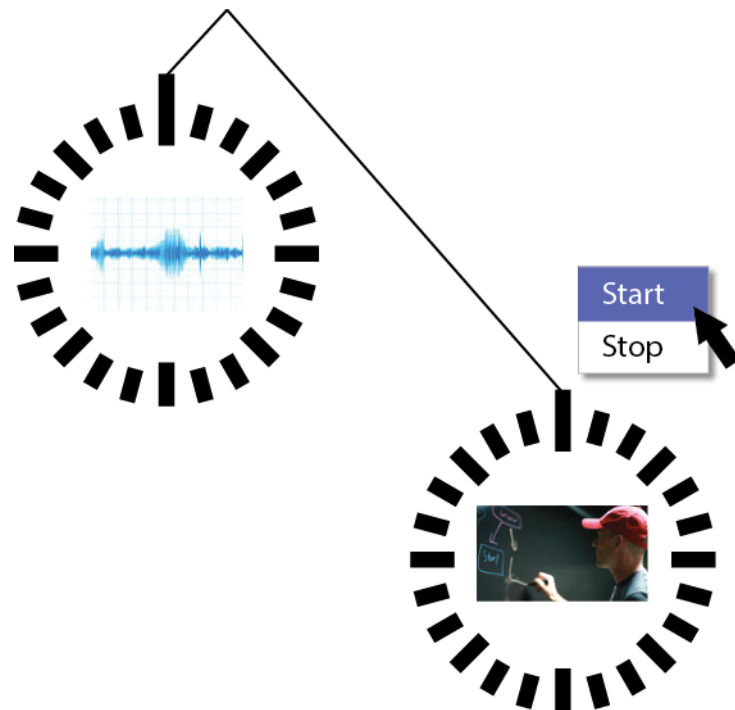


Figura 6 - Representação do menu de ação

Também não foi contemplado na proposta de interface através de anéis de interatividade, como as mídias de duração infinita, tais como texto e imagem são finalizadas, isto é, após serem iniciadas não há nenhum controle para se interromper a exibição deste tipo de mídia.

Uma solução simplificada seria a de permitir o usuário predefinir alterando um atributo de duração da mídia ao clicar duas vezes em um anel que seja deste tipo de mídia.

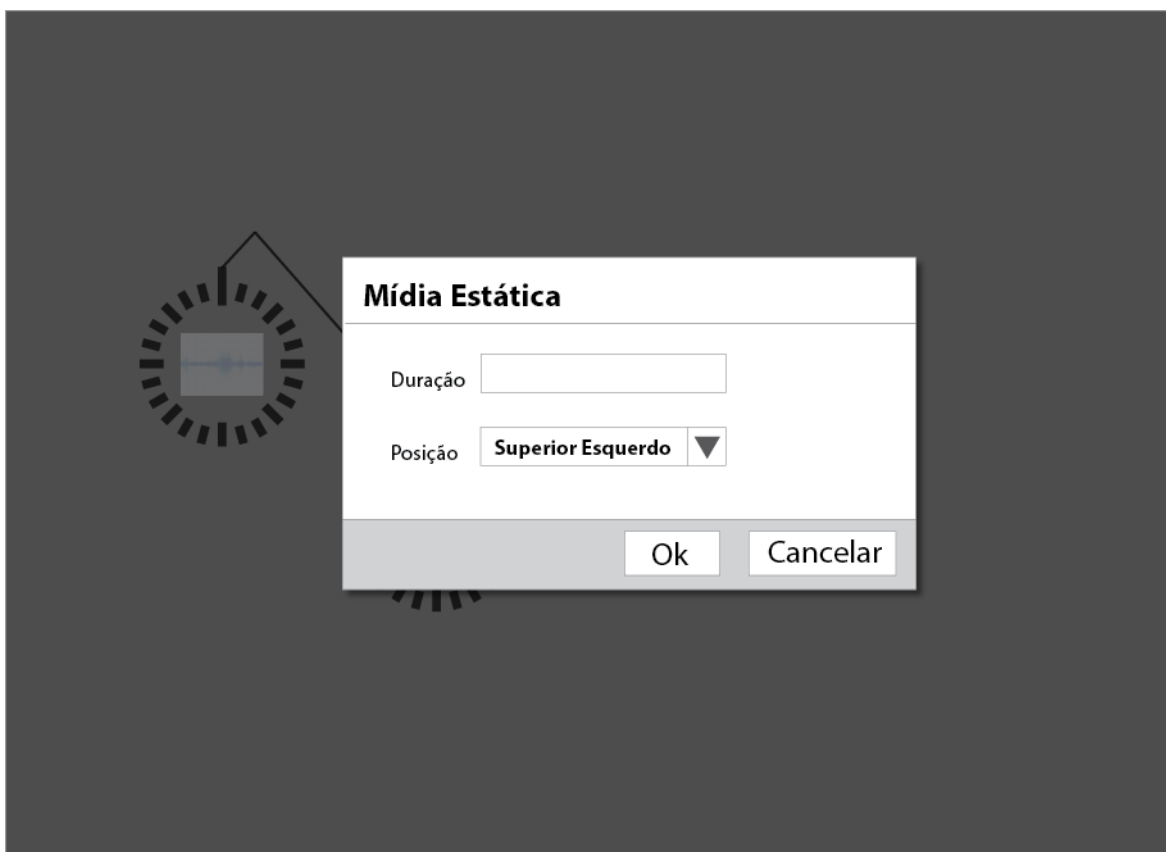


Figura 7 - Interface para edição de atributos de mídia estática

Em relação ao anel de interatividade ser uma representação de frações do tempo onde se pode colocar uma âncora, no caso das mídias de duração infinita o anel de interatividade confunde o autor do conteúdo ao não deixar claro as referências deste tipo de mídia em relação ao tempo.

O anel no entorno deste tipo de mídia deveria ser diferente de forma a mostrar que se trata de uma mídia contínua, somente a pós ter seus atributo duração definido que ele deveria se tornar em um anel como os outros.



Figura 8 - Representação de anel de mídia estática

Sobre a forma de representação fracionada do tempo no anel de interatividade, devido a essa discretização da duração, não é possível fazer ligações entre mídia em determinados

instantes, pois existem momentos que não estão representados nos traços do anel de interatividade.

Para permitir que todos os trechos fossem acessíveis para criação de uma âncora, uma possível solução de interface é de, no momento em que o usuário segurar o clique em um dos traços do anel sem mover, o anel expandir aquele trecho para permitir clicar em traços mais internos.

O anel de interatividade além de abstrair os conceitos de programação do usuário, também tem o propósito de ser uma interface simplificado, por essa razão é importante que o usuário esteja situado qual momento cada traço representa. Para oferecer ao usuário esta clareza, quando o mouse descansar sobre um dos traços deve ser exibido qual momento da mídia aquele traço representa.

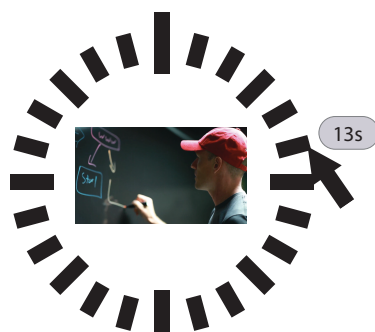


Figura 9 - Representação do momento da mídia no anel de interatividade

O anel de interatividade, da forma que foi proposto, também não deixa claro como criar uma âncora no instante final de uma mídia, para isso propomos a criação de um traço diferente que representa o final do tempo de execução da mídia.



Figura 10 - Representação do anel com marcação do final da mídia

4 IMPLEMENTAÇÃO

A fim de realizar uma prova de conceito das propostas deste trabalho, foi implementada uma interface que representa um perspectiva estrutural e temporal feita através de anéis de interatividade.

Esta interface poderia ser acoplada a um ferramenta mais completa ou poderia evoluir de forma a se tornar uma ferramenta auto-suficiente.

Neste capítulo é apresentada a descrição desta implementação com aprofundamento nas suas partes mais relevantes..

4.1 Tecnologias

A aplicação foi concebida utilizando como tecnologia principal a linguagem Java 8 e a plataforma de desenvolvimento de aplicações desktop com interfaces ricas, JavaFX. O principal motivador para a decisão da tecnologia a ser adotada foi a característica multiplataforma, os facilitadores para construção de recursos visuais nativos, além da popularidade da linguagem, o que facilita o entendimento e a continuidade do trabalho.

4.2 Arquitetura

A arquitetura da aplicação está dividida de maneira a separar o código de acordo com os contextos de domínio, a primeira divisão separa os recursos (*resources*) da aplicação do seu código fonte.

Em *resource* encontra-se os arquivos com extensão *.fxml* e *.css*. Os primeiros são documentos que definem a estrutura da interface gráfica, fazendo um analogia com a *web*, o *.fxml* são equivalentes aos arquivos *html*, porém em um ambiente de aplicação *desktop*. Os arquivos *.css*, assim como na *web*, definem a formatação da interface.

Em *source packages* se encontra o código fonte da aplicação dividido em pacotes de acordo com a responsabilidade de cada código, de forma a facilitar a manutenção do código.

O pacote principal é o *application*, que possui a classe *Main* e tem responsabilidade de instanciar as classes básicas de uma interface construída em JavaFX.

O pacote GUI, possui toda a parte de interface rica da aplicação, tais como os anéis de interatividade, os recursos visuais de relacionamento entre anéis e os eventos para manipular os elementos da interface.

Além destes pacotes, temos o CodeGenerator, responsável por tratar a geração das estruturas de código NCL, o Cache, onde as estruturas são temporariamente alocadas antes de serem escritas em um arquivo e a Util, que possui apenas classes para trabalhar com escrita de arquivos em disco.

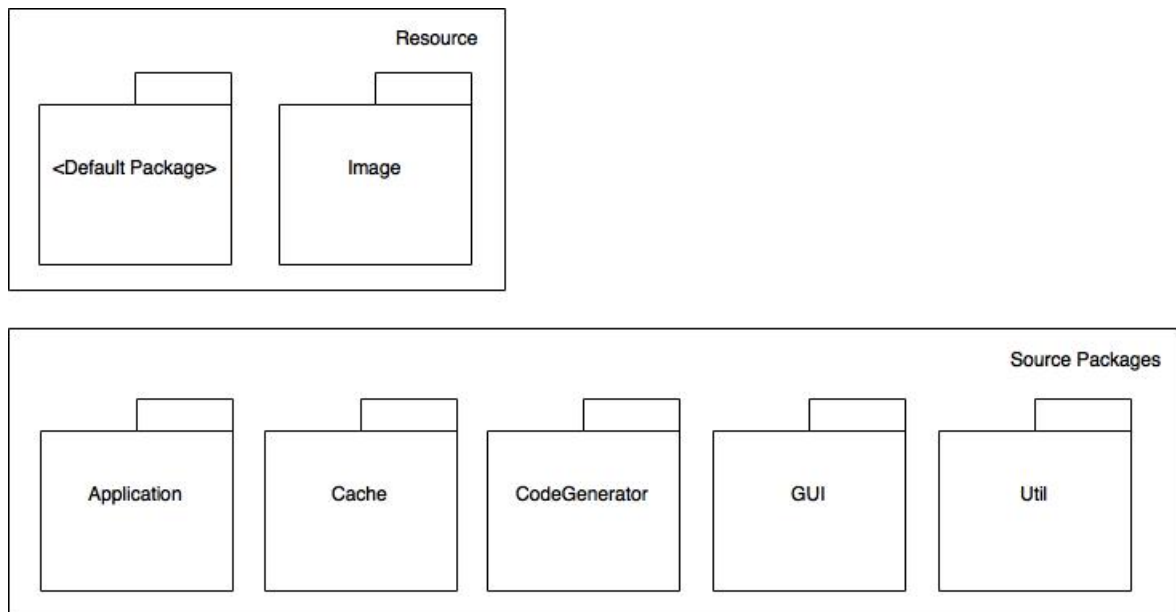


Figura 11 - Pacotes da aplicação

4.3 O Anel de Interatividade

As duas abordagens mais óbvias para implementação de uma estrutura gráfica que represente um anel de interatividade é através da utilização de uma imagem ou a por meio de um conjunto de vetores.

Com uma imagem teríamos perda de qualidade visual caso precisássemos redimensioná-la, além da dificuldade em manipular eventos de interação com os traços do anel, pois seria necessário mapear posições da imagem e adicionar eventos nestas regiões. Por esta razão, foi decidido utilizar a abordagem de representação através de vetores.

O anel é formado por 24 traços, cada traço gerado é rotacionado e posicionado de forma que o seu conjunto se alinhe como se estivessem em volta de uma circunferência.

Para facilitar a identificação do trecho que se deseja selecionar, a cada dois elementos um é destacado com um traço levemente maior.

```
private Node createTickMark(int n, double _topLeftX, double _topLeftY) {  
    return LineBuilder.create()  
        .layoutX(_topLeftX + 50)  
        .layoutY(_topLeftY - 10)  
        .startX(getSize())  
        .endX(getSize())  
        .startY(getSize() * 0.12)  
        .endY(getSize() * (n % 3 == 0 ? 0.3 : 0.2))  
        .transforms(  
            RotateBuilder.create()  
                .pivotX(getSize())  
                .pivotY(getSize())  
                .angle(360 / 24 * n)  
                .build()  
        )  
        .userData(n+"s")  
        .id(TICKMARK_ID+n)  
        .strokeWidth(getSize()*0.1)  
        .build();  
}
```

Figura 12 - Algoritmo de criação dos traços do anel de interatividade

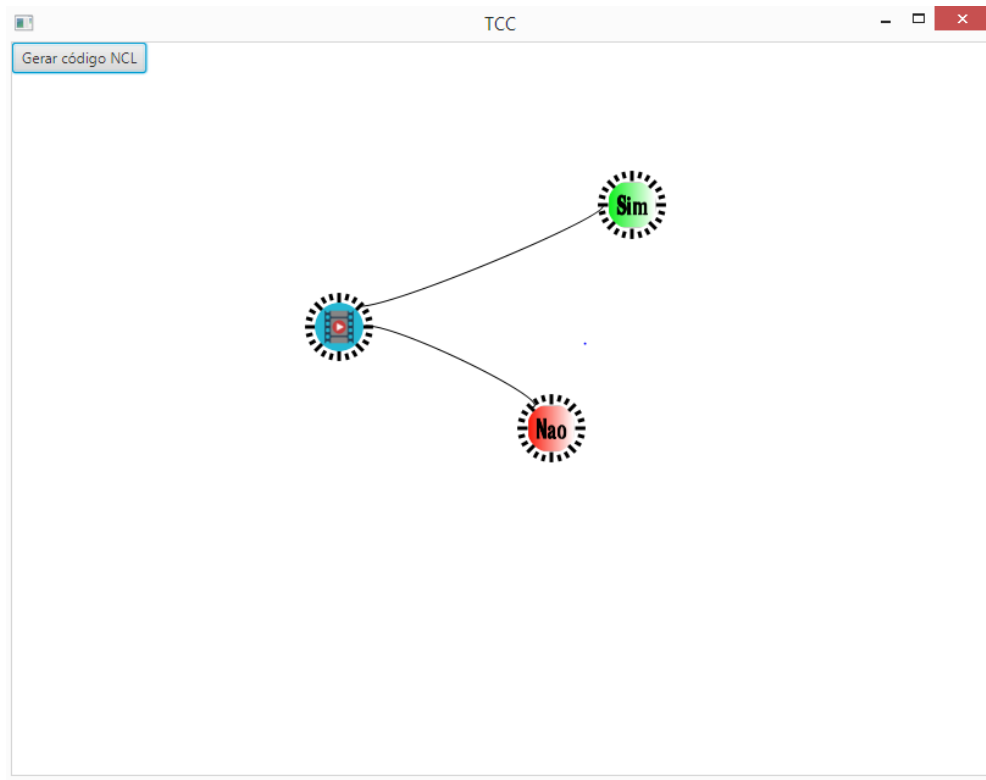


Figura 13 - Interface da aplicação com 3 nós de mídia

4.4 Geração de Código

Os elementos da interface, anéis e ligações, quando são adicionados pelo usuário na tela principal, geram uma estrutura que armazenam as informações necessárias para gerar o código NCL, de tal forma que no momento de produzir o código basta apenas percorrer as estruturas criando os nós do arquivo XML. Estas estruturas são a classe Media e a Classe NodeConnection, elas são armazenadas na classe NCLDataManager, a qual possui um dicionário com todas as mídias.

O momento de colocar a estruturas em cache são: quando uma mídia é adicionada na tela ou quando o usuário libera o cursor no anel de destino ao realizar uma conexão entre duas mídias.

Os atributos que não pertencem a visão estrutural e temporal, mas que são necessários para o funcionamento de uma aplicação, tais como volume de uma mídia de áudio, ou tamanho e posicionamento de mídias visuais, foram pré-definidos arbitrariamente para garantir que o código gerado fosse executado.

No caso do posicionamento, no modelo básico de código gerado já é produzido quatro descritores representando as regiões: esquerda superior, direita superior, esquerda inferior e direita inferior.

```
<?xml version="1.0" encoding="UTF-8"?>
<ncl xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile" id="TCC">
  <head>
    <regionBase>
      <region width="100%" id="rgTela" height="100%">
        <region top="0%" left="0%" width="25%" id="rgLeftTop" height="25%">
        </region>
        <region top="0%" left="50%" width="25%" id="rgRightTop" height="25%">
        </region>
        <region top="50%" left="0%" width="25%" id="rgLeftBottun" height="25%">
        </region>
        <region top="50%" left="50%" width="25%" id="rgRightBottun" height="25%">
        </region>
      </region>
    </regionBase>
    <descriptorBase>
    </descriptorBase>
    <connectorBase>
    </connectorBase>
  </head>
  <body>
  </body>
</ncl>
```

Figura 14 - Modelo padrão de código gerado

```

public class Media {
    private boolean isPort;
    private String id;
    private String descriptor;
    private String src;
    private Map<String, ArrayList<Node>> areas;

    public Media(String id, String descriptor, String src){
        this.id = id;
        this.descriptor = descriptor;
        this.src = src;
        this.areas = new HashMap<>();
    }

    public boolean isIsPort() {
        return isPort;
    }

    public void setIsPort(boolean isPort) {
        this.isPort = isPort;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getDescriptor() {
        return descriptor;
    }

    public void setDescriptor(String descriptor) {
        this.descriptor = descriptor;
    }

    public String getSrc() {
        return src;
    }

    public void setSrc(String src) {
        this.src = src;
    }

    public Map<String, ArrayList<Node>> getAreas() {
        return areas;
    }

    public void setAreas(Map<String, ArrayList<Node>> areas) {
        this.areas = areas;
    }

    public void put(String key, ArrayList<Node> nodes) {
        areas.put(key, nodes);
    }

    public Map<String, ArrayList<Node>> getAllAreas(){
        return areas;
    }

    public ArrayList<Node> getAreas(String cacheKey) {
        return areas.get(cacheKey);
    }

    public void clearAreas(String key) {
        areas.put(key, null);
    }

    public void clearAreas() {
        areas.clear();
    }
}

```

Figura 15 – Classe para representar uma mídia

A título de exemplo, produzimos um programa NCL com três mídias, um vídeo e duas imagens. O objetivo da aplicação é que após 3 segundos do início do vídeo fosse exibido a primeira mídia e depois de 6 segundos exibisse a segunda.

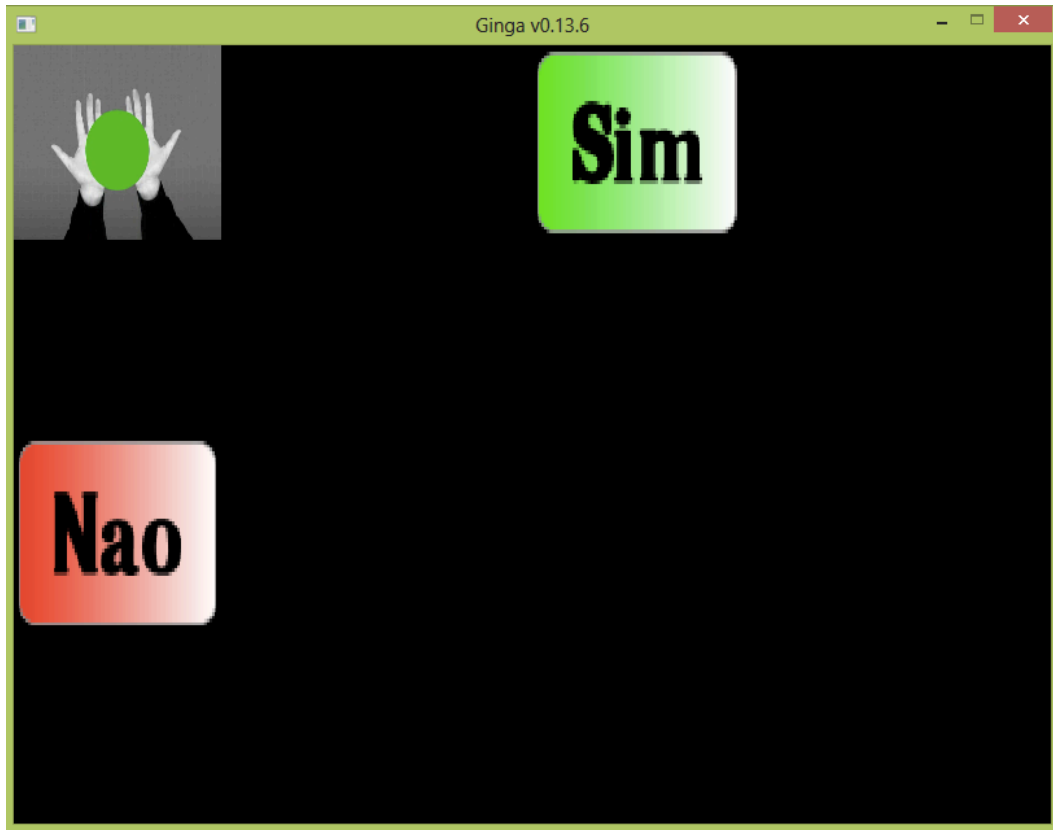


Figura 5 - Exemplo de aplicação construída pela interface

```

<?xml version="1.0" encoding="UTF-8"?>
<ncl xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile" id="TCC">
<head>
  <regionBase>
    <region width="100%" id="rgTela" height="100%">
      <region top="0%" left="0%" width="25%" id="rgLeftTop" height="25%"></region>
      <region top="0%" left="50%" width="25%" id="rgRightTop" height="25%"></region>
      <region top="50%" left="0%" width="25%" id="rgLeftBottun" height="25%"></region>
      <region top="50%" left="50%" width="25%" id="rgRightBottun" height="25%"></region>
    </region>
  </regionBase>
  <descriptorBase>
    <descriptor id="dCirole_demo.avi" region="rgLeftTop"></descriptor>
    <descriptor id="dCirole_sim.png" region="rgRightTop"></descriptor>
    <descriptor id="dCirole_ao.png" region="rgLeftBottun"></descriptor>
  </descriptorBase>
  <connectorBase>
    <causalConnector id="onBeginStartCirole_ao.png">
      <simpleCondition role="onBegin"></simpleCondition>
      <simpleAction role="start" max="unbounded" qualifier="par"></simpleAction>
    </causalConnector>
    <causalConnector id="onBeginStartCirole_sim.png">
      <simpleCondition role="onBegin"></simpleCondition>
      <simpleAction role="start" max="unbounded" qualifier="par"></simpleAction>
    </causalConnector>
  </connectorBase>
</head>
<body>
  <port component="Cirole_demo.avi" id="entryPoint"></port>
  <media src="media/demo.avi" id="Cirole_demo.avi" descriptor="dCirole_demo.avi">
    <area id="anchorCirole_ao.png" begin="3s"></area>
    <area id="anchorCirole_sim.png" begin="6s"></area>
    <property name="height" value="20%"></property>
    <property name="width" value="20%"></property>
  </media>
  <link xconnector="onBeginStartCirole_ao.png">
    <bind component="Cirole_demo.avi" role="onBegin" interface="anchorCirole_ao.png"></bind>
    <bind component="Cirole_ao.png" role="start"></bind>
  </link>
  <link xconnector="onBeginStartCirole_sim.png">
    <bind component="Cirole_demo.avi" role="onBegin" interface="anchorCirole_sim.png"></bind>
    <bind component="Cirole_sim.png" role="start"></bind>
  </link>
  <media src="media/sim.png" id="Cirole_sim.png" descriptor="dCirole_sim.png">
    <property name="height" value="20%"></property>
    <property name="width" value="20%"></property>
  </media>
  <media src="media/ao.png" id="Cirole_ao.png" descriptor="dCirole_ao.png">
    <property name="height" value="20%"></property>
    <property name="width" value="20%"></property>
  </media>
</body>

```

Figura 6 - Código NCL gerado através da interface

5 CONCLUSÃO

Este projeto propõe a criação de uma nova forma de interação dos usuários produtores de conteúdo para TV digital com ferramentas de autoria para esse tipo de produção. A interface é composta por estruturas gráficas que representam as mídias e os relacionamentos entre elas, de forma a abstrair detalhes de linguagens de programação para o usuário que possui conhecimento no domínio do conteúdo que está sendo desenvolvido, mas não em programação. A proposta é um refinamento do modelo preliminar descrito em Reis (2014).

No presente trabalho, foram abordados apenas aspectos de interface para lidar com as características temporais e estruturais da aplicação, isto é, o usuário pode manipular o momento que cada elemento de mídia é iniciado ou terminado e como cada elemento se relaciona com os outros. Assim, não fazem parte deste trabalho proposições sobre os aspectos relacionados ao layout, tais como a disposição dos elementos na tela e o tamanho, por exemplo. Isto é, o escopo aqui proposto se limitou a uma visão gráfica que estaria inserida em uma ferramenta, sendo que esta ferramenta, para atender as necessidades do usuário de maneira mais completa, precisa ser mais abrangente e envolver vários outros módulos aqui não contemplados. Portanto, a evolução da interface aqui apresentada de forma a se tornar uma ferramenta mais completa, ou a sua inserção como perspectiva de uma outra ferramenta, representam oportunidades para trabalhos futuros.

Também não foram realizados neste trabalho testes com o usuário final, a fim de verificar a usabilidade da ferramenta e nível de dificuldade que pode ser encontrado pelos profissionais de comunicação para trabalhar com essa forma de abstração. Além dos testes de usabilidade, uma análise mais ampla e aprofundada dos aspectos em torno da usabilidade da interface podem ser explorados em trabalhos futuros.

Outro trabalho futuro de interesse consiste na inclusão de funcionalidades para permitir que o usuário possa interagir com o anel de interatividade em qualquer momento da mídia. Além disso, seria interessante criar uma representação na própria interface de anéis de interatividade capaz de simular a apresentação multimídia. Essa simulação poderia indicar o progresso da apresentação de cada mídia, por meio de uma coloração gradual dos anéis e arestas envolvidos, de acordo com o andamento da apresentação como um todo.

REFERÊNCIAS BIBLIOGRÁFICAS

- LIMA, Bruno Seabra Nogueira Mendonça. **Composer: Aspectos não-funcionais em um ambiente de autoria para aplicações NCL**. 2011. Dissertação (Mestrado em Informática) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2011.
- MONTEZ, Carlos; BECKER, Valdecir. **TV Digital Interativa: Conceitos e Tecnologias**. In: WebMidia e LA-Web 2004 – Joint Conference. Ribeirão Preto, SP, Outubro de 2004.
- SANTOS, R. C. M. GOMES, T. A. AZEVEDO, R. G. A. NETO, C. S. S. TEIXEIRA, M. A. M. **NCL Eclipse: Ferramenta de Autoria Textual para a Linguagem NCL**. 11º Fórum Internacional de Software Livre (FISL 11). Julho, Porto Alegre, RS, Brasil. 2010.
- SOUZA JÚNIOR, P. J. **LuaComp: uma ferramenta de autoria de aplicações para TV digital**. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM 375/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Março de 2009.
- STENNING, K., OBERLANDER, J. **A cognitive theory of graphical and linguistic reasoning: Logic and implementation**. Cognitive Science: A Multidisciplinary Journal 1995, Volume 19.
- REIS, Ricardo Nogueira. **Design de Hipermídia: A Construção Não-linear da Notícia**. 2014. Dissertação (Mestrado em Informática) – Universidade Federal de Juiz de Fora, Minas Gerais, 2014.
- SOARES, Luiz Fernando. **Nested Context Model 3.0 Part 1 – NCM Core**. 2005. Monografia em Ciência da Computação. Rio de Janeiro. Pontifícia Universidade Católica do Rio de Janeiro, 2005