Uma abordagem MDE para Modelagem e Verificação de Documentos Multimídia Interativos

Delcino Picinin Júnior
DAS - CTC - UFSC
Bairro Trindade
Florianópolis - SC, Brasil
CEP: 88040-970
picinin@das.ufsc.br

Jean-Marie Farines
DAS - CTC - UFSC
Bairro Trindade
Florianópolis - SC, Brasil
CEP: 88040-970
farines@das.ufsc.br

Celso A. S. Santos
DMCC - UFBA
Av. Adhemar de Barros s/n
Salvador - BA, Brasil
CEP 40.170-110
saibel@dcc.ufba.br

ABSTRACT

This paper proposes an Model-Driven Engineering approach for modeling and verificating of interactive multimedia applications. The approach introduces a toolchain based on FIACRE language, that is used both as the target language of model transformation engines from NCL multimedia model, and as the source language of compilers into the targeted verification toolbox Tina. The paper illustrates the proposed approach with a case study of a NCL-based interactive TV application.

Categories and Subject Descriptors

H.5.4 [**Hypertext/Hypermedia**]: Information and Presentation; F.4.3 [**Formal Languages**]: Verification

General Terms

Verification,Languages,Design

Keywords

Model Driven Engineering, Model Checking, NCL

1. INTRODUÇÃO

O processo de desenvolvimento de aplicações de TV Digital Interativa (TVDI) envolve diferentes tipos de profissionais, com perfis bem distintos, tais como *designers* gráficos, jornalistas e programadores. Os *designers* gráficos trabalham em um nível de abstração mais elevado, preocupando-se mais com a qualidade audiovisual da aplicação, enquanto que os programadores trabalham em um nível mais baixo de abstração, preocupando-se com o desenvolvimento e testes da aplicação. As atividades desempenhadas pelo grupo de profissionais no processo de desenvolvimento devem ser integradas, de modo a produzir uma aplicação com qualidade no que se refere à apresentação do conteúdo, bem como garantir que a aplicação é correta no que se refere à consistência temporal.

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Full Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9642.
SBC - Brazilian Computer Society

As inconsistências temporais podem surgir quando estruturas mais complexas de sincronização (envolvendo eventos imprevisíveis, restrições de sincronização complexas, atrasos de codecs, de transmissão, etc.) são utilizadas na construção de aplicações hipermídia. A solução mais simples para evitar inconsistências temporais é reduzir o poder de expressão das linguagens para especificação da estrutura temporal de uma aplicação. Outra possibilidade é deixar a cargo do autor a tarefa de visualizar o comportamento temporal da aplicação por meio de projeções em eixos temporais (como nas ferramentas GriNs [5] e Composer [11]). A partir do comportamento observado nas projeções sobre os eixos, o autor pode inferir se existem problemas de consistência na estrutura temporal da aplicação. O tamanho e a complexidade da estrutura, além da impossibilidade de se avaliar exaustivamente todos os possíveis cenários envolvendo eventos imprevisíveis, são os fatores principais que limitam o uso dessa abordagem a cenários relativamente simples de sincronização. Por outro lado, é possível gerar estruturas temporais consistentes sem restrição de expressividade quando métodos de verificação são integrados ao processo de concepção de aplicações. O objetivo principal deste trabalho é apresentar uma abordagem baseada na Engenharia Dirigida por Modelos (*Model Driven Engineering* ou MDE) para concepção e verificação de aplicações hipermídia interativas.

A abordagem MDE visa melhorar o desenvolvimento de sistemas complexos, concentrando-se numa visão mais abstrata, através do uso de modelos de domínios ao invés da programação clássica, mais voltada a conceitos de computação. A abordagem MDE proposta nesse artigo está baseada no uso do ambiente de desenvolvimento do projeto TOPCASED [10], de código aberto e orientado para aplicações embarcadas, que tem como uma de suas principais técnicas de análise, a verificação, mas que integra também outras técnicas de análise e simulação. Conforme será apresentado, a cadeia de verificação proposta se inicia com uma especificação em alto nível (linguagem NCL [6]) que é transformada em um modelo intermediário (linguagem FIACRE [1]), que, por compilação, gera um Sistema de Transição Temporizado (TTS - *Timed Transition System*), a ser utilizado na verificação pela ferramenta TINA [2], de propriedades expressas na forma de formulas de lógica temporal LTL. A abordagem proposta é aplicada na modelagem e verificação de propriedades temporais num cenário de uma aplicação para TVDI.

O artigo está organizado da seguinte forma: na seção 2, a importância da verificação no projeto de aplicações hipermídia interativas é destacada; a seção 3 aborda metodologia de desenvolvimento baseada em modelos utilizada neste trabalho; a seção 4 apresenta os princípios de base da modelagem e verificação de aplicações escritas ma linguagem NCL; a seção 5 apresenta um estudo de caso; na seção 6, alguns trabalhos relacionados são discutidos, finalmente, a seção 7 apresenta as conclusões e perspectivas.

2. VERIFICAÇÃO DE APLICAÇÕES MULTIMÍDIA INTERATIVAS

A especificação da estrutura temporal de um documento hipermídia caracteriza, através de abstrações de alto nível (i.e., através de um modelo temporal), as ações e/ou condições necessárias à sincronização dos objetos que compõem esse documento [16]. A formalização dessa estrutura temporal para sua posterior verificação pode ser feita:

- De forma manual, utilizando uma técnica formal diretamente na fase de concepção do documento interativo, a qual é normalmente muito complexa para os autores [22] [21]. Além disso, uma descrição obtida por esta abordagem tem o risco de ser específica e não reutilizável.
- De forma manual e semi-automática, utilizando mecanismos de composição sucessiva de componentes para descrever o comportamento dinâmico do documento [8] [17]. Essa abordagem é mais simples e eficaz que a anterior, além de permitir a criação de objetos simples, reutilizáveis e de semântica precisa (biblioteca de componentes formais), que podem ser compostos facilmente para descrever estruturas temporais. Por outro lado, ela ainda está baseada na tradução semi-automática das descrições em alto-nível para um modelo formal, o que requer um grau de entendimento de ambos os modelos (de concepção e de formalização) a serem usados. Outro ponto é que cada alteração feita no nível de concepção exigiria testes exaustivos de todos os casos de execução, visando garantir a consistência temporal da aplicação. Dependendo da situação, tais testes demandariam muito tempo e trabalho, além do fato de alguns casos poderem não ser testados por negligência ou esquecimento.
- De forma automática, a partir de transformações de modelos de alto nível usado na concepção até modelos formais usados na verificação. Essa é a abordagem proposta neste artigo e que tem como ponto forte a independência de um modelo de concepção específico. A formalização da representação formal de cenários interativos permite: (i) assegurar a consistência entre diferentes visões dos cenários e (ii) melhorar a qualidade da aplicação final por meio da verificação da sua corretude durante o processo de autoria.

A representação formalizada dos cenários interativos pode ser aplicada para a verificação de propriedades de consistência temporal [8] [17], simulações e análises exaustivas [22] e ainda, para o controle da apresentação de um documento interativo [16] [13] [3].

A verificação da coerência temporal de documentos interativos pode ser feita por meio de projeções em eixos temporais (em geral, obtidos automaticamente pelo ambiente de autoria [5], [11] ou usados como base para a autoria [12]), grafos [4], Redes de Petri [22], autômatos (estados e transições) [8] [17], entre outros.

3. DESENVOLVIMENTO BASEADO EM MDE

MDE. A Engenharia Dirigida por Modelos (MDE) é uma metodologia de desenvolvimento de sistemas que permite construir uma aplicação, ou parte dela, a partir de modelos. Diversas Linguagens de Modelagem para Domínio Específico (DSML) podem ser utilizadas para representar diferentes aspectos durante o desenvolvimento de um sistema; por exemplo, uma linguagem de usuário, uma linguagem de simulação ou de verificação. Para utilizar modelos para poder gerar código ou documentação, validar, simular ou verificar, é necessário transformar os modelos de forma rigorosa. A transformação de modelos é também utilizada para definir mapeamentos e traduções entre diferentes linguagens; por exemplo de linguagens usuário para linguagens de verificação. A garantia de uma transformação rigorosa é baseada no uso de uma hierarquia de metamodelos na representação das linguagens de domínio DSML. A OMG definiu uma estrutura de modelagem baseada numa hierarquia de meta-modelos: o mundo real no nível M0; modelos que o representam no nível M1; metamodelos que define estes modelos no nível M2 e finalmente no topo desta estrutura um único meta-metamodelo, MOF (*Meta-Object Facility* que pode se auto-definir) [7]. Um metamodelo é um modelo que descreve os elementos que permitem definir as linguagens de modelagem. O ambiente TOPCASED utilizado na abordagem proposta fornece um arcabouço para a transformação de modelos, seguindo a estrutura hierárquica proposta pela OMG. Assim, o ambiente fornece a base para transformar de forma rigorosa, modelos escritos numa linguagem de usuário em modelos em linguagens adequadas a ferramentas de verificação.

A cadeia de verificação TOPCASED. Este trabalho utiliza uma metodologia e algumas das ferramentas encontradas na abordagem TOPCASED, para construir uma cadeia de verificação para sistemas hipermídia. Devido à diversidade de linguagens de usuário existentes (UML, SysML, AADL, etc.) e de formalismos para ferramentas de verificação (redes de Petri, autômato, algebra de processo, etc.), o ambiente TOPCASED utiliza a linguagem FIACRE [1] como um passo intermediário da transformação de modelos. A cadeia de verificação tem, consequentemente, dois níveis de transformação: um nível da linguagem usuário (NCL, para autoria hipermídia) para a linguagem intermediária FIACRE e, outro nível, de FIACRE para linguagens de verificação (TTS, utilizado pela ferramenta TINA). Os principais benefícios da utilização de FIACRE são: a redução da lacuna semântica entre linguagens de alto nível e formalismos para ferramentas de verificação, e a definição de uma única semântica para diferentes cadeias de verificação, tornando mais fácil a introdução na cadeia, de novas linguagens de alto nível e de novas ferramentas de verificação. Neste trabalho, FIACRE facilita a utilização da ferramenta TINA do projeto TOPCASED numa cadeia de verificação orientada

a aplicações interativas escrita numa linguagem de domínio como NCL.

FIACRE. A FIACRE é uma linguagem fortemente tipada que oferece uma representação formal de aspectos comportamentais e temporais de um sistema. As construções sintáticas de base de FIACRE são as seguintes: *process* , que descreve o comportamento de componentes sequenciais; e *component* que, representa um sistema como a composição de processos, possivelmente de forma hierárquica. O comportamento de um *process* é definido por um conjunto de estados e de transições. Para cada estado, uma expressão especifica as transições de estado e a passagem ao estado seguinte. Estas expressões são escritas por meio de construções determinísticas similares àquelas encontradas nas linguagens de programação clássicas (atribuição, condição, laços, composição sequencial, ...), construções não determinísticas (escolha, atribuição não determinística) e eventos de comunicação sobre portas. Um *Component* é definido como uma composição paralela de componentes e/ou processos, comunicando-se através de portas síncronas e de variáveis compartilhadas. A sintaxe dos componentes permite restringir o modo de acesso e a visibilidade das variáveis compartilhadas e portas. Restrições temporais podem ser associadas às comunicações sobre as portas. A definição de prioridades entre eventos de comunicação é outra das características que pode ser representada nos componentes.

4. MODELAGEM E VERIFICAÇÃO DE APLICAÇÕES INTERATIVAS NCL

4.1 A linguagem NCL

A linguagem NCL (*Nested Context Language*) é uma linguagem declarativa para autoria de documentos hipermídia baseada no modelo conceitual NCM (*Nested Context Model*)[20]. De maneira similar aos princípios adotados pelo W3C para a linguagem SMIL, a NCL foi desenvolvida utilizando uma estrutura modular para definir como os objetos de mídia se relacionam no tempo e no espaço durante uma apresentação. A concepção de documentos NCL se baseia no conceito de nó (ou *node*), que especifica um fragmento de informação ou um contexto, e de elo (ou *link*), que define relacionamentos entre os nós. Um documento NCL é estruturado basicamente em duas partes: cabeçalho (associado ao elemento <*head*>) e corpo (associado ao elemento <*body*>). As bases de regiões, descritores e conectores são definidos na parte do cabeçalho e os objetos de mídia, seus relacionamentos (por meio de *links*), além das portas de entrada de um contexto (de início de apresentação de uma composição de nós) no corpo do documento.

Os *links* são de fundamental interesse para o processo de modelagem proposto nesse artigo, já que a estrutura temporal de apresentação do documento é representada pelo conjunto de relacionamentos (restrições) descritos por meio dos *links* NCL. Um *link* relaciona objetos de uma aplicação NCL por meio de um conector (elemento *causalConnector*). Um conector define o tipo de relação (restrição) de sincronização (causalidade) entre objetos que compõem um documento NCL. Em outras palavras, um conector define a condição que deve ser satisfeita para que ações sejam disparadas durante a apresentação de um documento.

A semântica de um relacionamento NCL é especificada por meio de uma relação associada a um conector (atributo *xconnector* do elemento *link*) e pelos atores que definem a relação (restrição) de sincronização, especificados pelo elemento *bind*. Bases de conectores podem ser criadas para especificar qualquer tipo de restrição de sincronização envolvendo qualquer número de atores em NCL. Como exemplo, seja uma das relações de sincronização do documento "O Primeiro João" (apresentado em [19] e utilizado no estudo de caso nesse artigo) especificada da seguinte maneira: ao ser iniciada a exibição da animação (*animation*), a imagem de fundo (*bg*) e o áudio com o chorinho (*choro*) também devem ser iniciados, mas 5s depois. Os trechos relacionados ao *causalConnector* e o *link* em NCL para essa especificação são os seguintes:

```
 ...
<causalConnector id="onBeginStartDelay">
   <connectorParam name="delay"/>
   <simpleCondition role="onBegin"/>
   <simpleAction role="start" delay="$delay"
    max="unbounded" qualifier="par"/>
</causalConnector>
 ...
<link  id="lMusic" xconnector="onBeginStartDelay">
   <bind role="onBegin" component="animation"/>
   <bind role="start" component="bg">
<bindParam name="delay" value="5s"/>
   </bind>
   <bind role="start" component="choro">
<bindParam name="delay" value="5s"/>
   </bind>
</link>
```

O conector *onBeginStartDelay* estabelece uma relação de sincronização genérica para *links* do tipo: quando a apresentação do componente no papel(*role*) ativo *simpleCondition role="onBegin"* começar, a apresentação do conjunto de componentes no papel(*role*) passivo *simpleAction role="start" delay="$delay"* podem ser iniciadas, isso após um certo *delay*, a ser definido por cada instância de elo criada a partir do conector. No exemplo, após o início da apresentação do componente *animation*, irá demorar 5s para o início da apresentação dos componentes *bg* e *choro*.

Outro conceito importante para a sincronização em NCL é o de âncora (de conteúdo). Uma âncora marca uma parte (no tempo e/ou no espaço) do conteúdo de uma mídia, servindo de base para os elos e para definir uma interface para o acesso àquele conteúdo. Quando o conteúdo é interativo, a âncora define quando (tempo) e onde (espaço) o usuário poderá interagir com este conteúdo durante a apresentação. Em termos de sincronização, uma âncora define uma janela temporal de oportunidade de interação para o usuário.

Pode-se observar que relações envolvendo condições complexas de sincronização e conectores envolvendo muitos atores aumentam a complexidade dos cenários e a possibilidade de geração de inconsistências. Assim, um processo automático para obtenção de representações em outras linguagens que permitam a verificação de inconsistências temporais torna-se fundamental.

4.2 Princípios de base da modelagem

Arquitetura FIACRE de uma aplicação NCL . A arquitetura FIACRE gerada a partir de NCL, segue a mesma estrutura hierárquica encontrada num programa NCL, incluíndo o anhinamento dos nós. O componente FIACRE *main*, que representa a totalidade do sistema, é o resultado da composição paralela dos componentes FIACRE que representem os nós de mídia e de contexto do primeiro nível da hierarquia NCL, e de um componente controlador que representa a interação do usuário.

Todos os nós, de contextos e de mídias, são traduzidos em componentes FIACRE. Caso o nó de mídia não esteja no primeiro nível hierárquico NCL, seu componente será declarado dentro do componente que representa o nó do contexto onde está inserido. Os nós de mídia NCL geram também processos FIACRE, declarados dentro do componente correspondente a este nó de mídia.

A comunicação entre processos em FIACRE é feita através de portas síncronas. Para dar suporte à modelagem da comunicação assíncrona da NCL entre os componentes que se comunicam, é necessário utilizar um mecanismo de variáveis compartilhadas ou um processo intermediário que os interliga (denominado processo *glue*). Esse processo se sincroniza com as duas mídias por meio de portas diferentes, garantindo desta forma, o assincronismo entre elas. No estudo de caso apresentado, a comunicação assíncrona é realizada por meio de um processo *glue*.

Regras de transformação de modelos: NCL à FIACRE . Na transformação de uma aplicação NCL para um modelo FIACRE, todas as informações contidas no arquivo NCL são analisadas, com destaque para aquelas que influenciam a estrutura da aplicação, o sincronismo e as restrições temporais para a apresentação das mídias. As regras de transformação propostas a serem implementadas no tradutor NCL-FIACRE visam manter a estrutura do modelo FIACRE o mais semelhante possível daquela da aplicação NCL.

Nas regras propostas, para todos os nós, de contexto e de mídia, são gerados componentes em FIACRE, com identificação similar a dos nós NCL. Além da criação do componente, cada nó de mídia gera a criação de dois processos dentro de seu referido componente. Desses dois processos, um representa o comportamento do nó de mídia e o outro representa um *glue*.

As informações dos descritores NCL são utilizadas para especificar o tempo de execução dos processos FIACRE a serem criados. As informações contidas nos conectores e nos *links* NCL, que relacionam as mídias, permitem criar as portas de comunicação entre os processos FIACRE. As âncoras associadas às mídias, utilizadas em NCL para muitas ações de sincronismo entre mídias, são usadas para definir o momento de emitir mensagens entre processos. Em FIACRE, os tempos entre os envios de mensagens por um processo podem ser especificados por meio da construção *wait*[*min, max*] ou por um intervalo de tempo com valor mínimo e máximo nas portas de comunicação. A segunda opção é a que será utilizada no estudo de caso.

4.3 Verificação de aplicações interativas

A cadeia de verificação para aplicações hipermídia interativas é representada na Figura 1. A partir de uma ferramenta de autoria é gerado um arquivo XML representando o código NCL da aplicação. Um tradutor, construído seguindo a abordagem MDE, transforma o código NCL em FIACRE, seguindo as regras e os príncipios de base enunciados anteriormente. A última transformação, de FIACRE para TTS utiliza o compilador FRAC, já desenvolvido no projeto TOPCASED. Atualmente, as fórmulas de lógica temporal, que expressam as propriedades a serem testadas, são escritas pelo projetista. Entretanto, o objetivo da abordagem é que essas fórmulas sejam geradas automaticamente a partir das especificações dos *designers* dentro do seu domínio. As propriedades a serem verificadas são, sobretudo, aquelas orientadas pela aplicação que permitem verificar a corretude do documento, em particular, do ponto de vista de sua consistência temporal e do controle das interações.

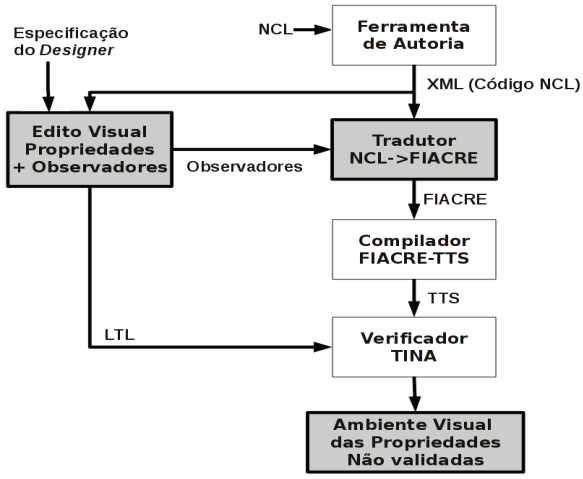

Figure 1: Cadeia de Verificação NCL-TTS

5. ESTUDO DE CASO: ANÁLISE DE APLICAÇÕES INTERATIVAS NCL

A Figura 2 ilustra a estrutura da aplicação "O Primeiro João" [20] usada no estudo de caso. Ela é composta por dois vídeos, um áudio, três imagens e dois formulários (cada um num idioma diferente) a serem escolhidos pelo usuário. Os elos entre as mídias são indicados por linhas direcionais, com rótulos contendo o tipo e o instante de envio da mensagem, por exemplo, *Início(aos 6 segundos)*.

A mídia "Vídeo Principal" é a primeira a ser ativada, sendo responsável pela ativação das outras quatro mídias: "Som" e "Imagem de Fundo" após 5s; "Imagem Foto", após 6s e "Icone Interativo", após 15s. As mídias "Formulário" e "Vídeo Chuteira" são ativadas no momento em que ocorre uma interação (*click*) sobre a âncora que marca o conteúdo da mídia "Ícone Interativo".

O término da apresentação de uma mídia ocorre após sua duração intrínseca ou após o recebimento de uma mensagem de outras mídias. O "Vídeo Principal" e o "Som" terminam a apresentação em 70s, a "Imagem Foto" em 10 ou 20s, dependendo da resolução do monitor e o "Vídeo Chuteira" em 14s,

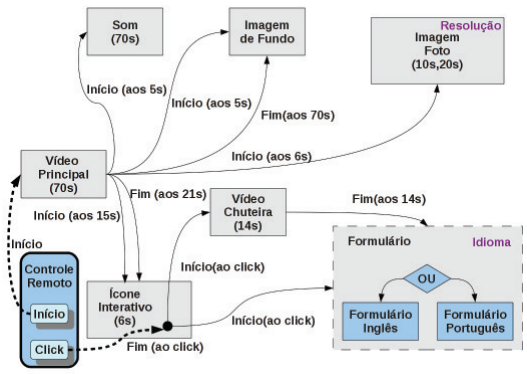

Figure 2: Estudo de caso (O Primeiro João)

correspondendo ao primeiro caso. As mídias que terminam por mensagem são: a "Imagem de Fundo", após receber uma mensagem de término do "Vídeo Principal"; o "Ícone Interativo", aos 21s após receber a mensagem de término enviada pelo "Vídeo Principal" ou ainda, por uma mensagem de si mesmo, ativada por um *click*; o "Formulário", após receber uma mensagem de término do "Vídeo Chuteira".

As mensagens de início das mídias "Imagem Foto" e "Ícone Interativo" são enviadas a partir de âncoras existentes no "Vídeo Principal", após 6 e 15s, respectivamente. A mensagem de término da mídia "Ícone Interativo" é emitida, a partir de uma âncora temporal, aos 21s de apresentação do "Vídeo Principal".

5.1 Modelagem

Arquitetura. No primeiro nível da estrutura hierárquica FIACRE, o componente *main* é composto por vários componentes e processos interligados através de portas. Cada nó de mídia ou de contexto no primeiro nível hierárquico NCL é representado por um componente em FIACRE. Além dos componentes que representam os nós NCL, o *main* possui ainda um componente que representa eventos externos (interação com o controle remoto). O código do *main*, ilustrado a seguir, tem 8 componentes, dos quais 6 para os nós de mídia, 1 para um *switch* (considerado um nó de contexto) e 1 para os eventos externos.

As portas que interligam estes componentes são obtidas a partir dos *links* NCL. Na seção 4, foi apresentado um exemplo de *link* entre a mídia "Vídeo Principal" (animation) e as mídias "Imagem de Fundo"(bg) e "Som"(choro). Este *link* informa que após 5s do início da mídia "Vídeo Principal", as mídias "Imagem de Fundo"(bg) e "Som"(choro) devem começar. No código FIACRE, apresentado a seguir, este *link* é representado pela porta *ativa5s* com o valor 5s associado na forma de um intervalo [5,5].

O *switch* intitulado *form*, que contém os formulários, é transformado em FIACRE no componente c_form, que contém 2 processos, representando uma *rule* (responsável pelo controle da apresentação dos formulários) e um *glue*, e 2 componentes, representando os formulários existentes.

```
component main is

port
    a_inicio_anima:bool in [0,0], ativa5s:bool in [5,5],
    des70s:bool in [70,70],       ativa6s:bool in [6,6],
    ativa15s:bool in [15,15],     des21s:bool in [21,21],
    s_click_icon:bool in [2,4],   des14s:bool in [14,14],
    s_rep_click:bool in [0,0]


par * in
    c_interacao[a_inicio_anima,ativa15s,s_click_icon]
    || c_background[ativa5s,des70s]
    || c_music[ativa5s]
    || c_icon[ativa15s,des21s,s_click_icon,s_rep_click]
    || c_form [s_rep_click,des14s]
    || c_shoes[s_rep_click,des14s]
    || c_photo[ativa6s]
    || c_animation[a_inicio_anima,ativa5s,des70s,ativa6s,
                   ativa15s,des21s]
end
```

```
component c_form [chegaStart:bool,chegaStop:bool] is
...
par * in
  p_form[rep1,rep2,startPtf,stopPtf,startEnf,stopEnf]
  || glueAD[chegaStart,chegaStop,rep1,rep2]
  || c_ptform[startPtf,stopPtf]
  || c_enform[startEnf,stopEnf]
end
```

O formulário em português intitulado *ptform*, é transformado em FIACRE no componente c_ptform, que contém 2 processos, um representando o comportamento do formulário e o outro representando um *glue*.

```
component c_ptform[chegaStart:bool,chegaStop:bool]  is
...
par * in
  p_ptform[repassa1,repassa2]
  || glueAD[chegaStart,chegaStop,repassa1,repassa2]
end
```

O corpo do programa. Na transformação de uma mídia em um processo devem ser levados em conta os comportamentos temporais, bem como os relacionamentos entre esta mídia e as demais. Estas informações comportamentais e de relacionamento são obtidas diretamente do código NCL. No decorrer desta seção, apenas a parte da apresentação relativa à mídia "Vídeo Principal" será usada nos exemplos.

A duração da mídia é especificada no seu descritor NCL. Se esse não for o caso, é provável que essa mídia termine por uma restrição definida por um *link*, que especifica qual mensagem será responsável pelo término da apresentação dessa mídia. No estudo de caso, a mídia "Vídeo Principal" tem duração de 70 segundos.

A partir da declaração da mídia em NCL é possível se obter os eventos gerados por essa mídia. Toda mídia tem eventos de início e de fim de apresentação e, eventualmente, outros eventos associados às suas âncoras. Os eventos são usa-

dos para o envio de mensagens. A mídia "Vídeo Principal" (*animation*), apresentada a seguir, contém duas âncoras: a primeira possui um evento de ativação aos 6s e a segunda, um evento de ativação aos 15s e de desativação aos 21s. Levando-se em conta os instantes de início (0s) e de fim (70s), a mídia "Vídeo Principal" possui 5 eventos, que podem ser usados para gerar mensagens entre os componentes da apresentação.

```
<media id="animation" src="animGar.mp4"
          descriptor="videoPDesc">
  <area id="segPhoto" begin="6s"/>
  <area id="segIcon" begin="15s" end="21s"/>
  <property name="bounds"/>
</media>
```

Todos estes eventos e suas durações serão representadas em FIACRE pelas portas *a_inicio_anima, des70s, ativa6s, ativa15s, des21s*, com seus respectivos tempos, conforme mostrado no código *main*.

Conhecendo-se o tempo de duração da mídia e os eventos a ela associados, a transformação desse componente NCL em FIACRE deve levar em conta apenas os *links* que possuem ligações com os eventos desta mídia e os conectores que definem o comportamento do *link*.

Continuando a análise dos *links* do "Vídeo Principal", observa-se que os eventos *onBegin* associados às suas duas âncoras também estão sendo usados. Eles interligam a mídia "Vídeo Principal" às mídias "Ícone Interativo" e "Imagem Foto", nos instantes 6 e 15s(associados às âncoras).

Estes dois *links* NCL geram duas portas FIACRE, identificadas no código a seguir por *ativa6s* e *ativa15s*. Elas devem ser interligadas, respectivamente, aos componentes e processos que representam as mídias "Ícone Interativo" e "Imagem Foto" por meio de componentes do tipo *glue*, garantindo o assincronismo na comunicação entre as mídias. Os atrasos de 6 e 15s (declarados nas âncoras NCL) são representados em FIACRE por intervalos de tempo de atraso nas portas, declaradas no componente *main* apresentado no início dessa seção.

A parte comportamental do processo FIACRE mostrado abaixo (que representa a apresentação do "Vídeo Principal") foi obtido a partir de informações contidas nas estruturas NCL: *links*, conetores, descritores e âncoras.

5.2 Verificação de Propriedades

O término de todas as mídias, com a possibilidade de reinicialização, é uma propriedade importante para garantir a reprodutibilidade da aplicação. Ela pode ser verificada como uma propriedade de alcançabilidade de estados finais de todas as mídias ou do retorno destas ao estado (inicial) de repouso; a introdução de um processo observador com um estado *reinicio_observado*, que indica que todas as mídias estão no estado de repouso permite verificar essa propriedade. A inclusão do observador facilita a expressão LTL da propriedade que expressa que se o controlador encerra infinitas vezes, todas as mídias são reiniciadas infinitas vezes, representada por:
$\Box(\Diamond controlador\_idle) \rightarrow \Box(\Diamond reinicio\_observado)$

```
process  p_animation[a_inicio_anima:bool,ativa5s:bool,
                     des70s:bool,ativa6s:bool,
                     ativa15s:bool,des21s:bool]is

states idle,e1,e2,e3,e4,e5

var aa_inicio_anima:bool:=false, aativa5s:bool:=false,
    adesativa70s:bool:=false ,   aativa6s:bool:=false,
    aativa15s:bool:=false,   adesativa21s:bool:=false

init to idle

from idle  a_inicio_anima? aa_inicio_anima;   to e1
from e1    ativa5s! aativa5s;                 to e2
from e2    ativa6s! aativa6s;                 to e3
from e3    ativa15s! aativa15s;               to e4
from e4    des21s! adesativa21s;              to e5
from e5    des70s! adesativa70s;              to idle
```

As verificações também podem permitir descobrir eventuais erros na escrita do código NCL da aplicação, dificilmente percebidas na fase de autoria, como no exemplo a seguir. Nesse exemplo, quando o "Ícone Interativo" for selecionado: (i) a mídia "Vídeo Principal" deve ser redimensionada, ocupando uma área menor na tela e (ii) duas novas mídias devem entrar em execução, "Video Chuteira" e "Formulário". A seleção do "Ícone Interativo" provoca o fim da sua apresentação.

O aviso do término (*stop*) do "Ícone Interativo" é representado no *connector* NCL por:
`<simpleAction role="stop" max="unbounded" qualifier="par"/>`

No *link* NCL, é informado a associação de *stop* com o "Icone Interativo" (*icon*):
`<bind role="stop" component="icon"/>`.

Caso o desenvolvedor se esqueça de programar a desativação do "Ícone Interativo", este oferecerá a possibilidade de uma nova interação que causará um novo redimensionamento da mídia "Vídeo Principal" e a ativação das mídias "Vídeo Chuteira" e "Formulário". Nos experimentos feitos com o ambiente Ginga para essa aplicação inconsistente, foi observado que, o envio de uma segunda mensagem de início (novo *click* no "Ícone Interativo") irá provocar o travamento das mídias "Vídeo Chuteira", "Formulário" e "Imagem de Fundo". Esta situação de erro foi simulada a partir da retirada do *link* correspondendo ao controlador que gera o *click*, na seguinte linha:

```
<bind role= "stop" component="icon"/>
```

Neste caso, a verificação do código FIACRE (traduzido a partir do código NCL) permitiu detectar o erro: (i) a partir da não satisfação de uma fórmula que indica que sempre que um *click* for enviado ao "controlador", o "Ícone" deve esperar por um novo *click* do tipo:
$\Box(click\_botaoapertado \rightarrow \Diamond icon\_esperanovoclick)$

e (ii) a partir da análise de um contra exemplo, que descreve o caminho (sequência de estados) que leva a este resultado de erro. Desta análise foi possível localizar a parte do código na qual se encontrava a origem do erro.

Os dois exemplos anteriores têm como objetivo mostrar as potencialidades da metodologia proposta. Ressalta-se ainda que a verificação está baseada numa tradução rigorosa do modelo NCL no modelo TTS utilizado pela ferramenta de verificação TINA, seguindo os princípios da MDE.

6. TRABALHOS RELACIONADOS

Alguns trabalhos envolvendo a verificação de aplicações hipermídia foram propostos nos últimos anos. Entre eles, podem ser destacados aqueles voltados para validar aplicações desenvolvidas em SMIL e NCL (NCM).

Em [15] um documento hipermídia em SMIL 2.0 é transformado de maneira automática para o formalismo RT-LOTOS ($Real - TimeLOTOS$)[9]. Após a transformação, a verificação do documento é feita por *model checking*. No processo de transformação, um documento SMIL é analisado e seus componentes são classificados em duas categorias: Operacionais e Contextuais. Nesta classificação, os componentes operacionais descrevem a lógica e o comportamento temporal da aplicação, já os contextuais representam os elementos e atributos que descrevem o sincronismo espacial. Após a identificação dos diferentes tipos de componentes, estruturas intermediárias são criadas para representar os componentes e seus comportamentos temporais. Por fim, baseando-se nestas estruturas, a especificação RT-LOTOS é criada. A verificação das propriedades temporais é feita por um grafo de alcançabilidade gerado a partir do RT-LOTOS. O benefício da abordagem é a possibilidade da verificação do documento em RT-LOTOS de maneira implícita para o autor do documento.

Em [3] é apresentada uma ferramenta que permite a verificação incremental de documentos SMIL; esta ferramenta é baseada em um modelo chamado H-SMIL-Net *(Hierarchical SMIL Petri Net)*. Ela permite a edição de documentos SMIL através de diferentes níveis de abstração, e garante sua consistência temporal através de verificação incremental realizada durante a construção ou edição, e não apenas após a conclusão do documento. Uma verificação localizada permite uma independência na concepção dos diferentes componentes do documento, diminuindo o tempo necessário para a verificação do documento. De acordo com o autor, um ponto forte desta ferramenta é a possibilidade do uso de uma modelagem formal com um bom tempo de resposta.

No trabalho apresentado em [17], um documento hipermídia modelado em NCM (modelo conceitual no qual a linguagem NCL está baseada) é traduzido para o formalismo RT-LOTOS, a partir do qual o mesmo pode ser verificado. As restrições temporais intrínsecas, ou seja, pertencentes à aplicação, e também as extrínsecas, pertencentes à plataforma de execução da aplicação são verificadas. Como exemplo de restrição temporal extrínseca podemos citar o tempo de carregar um vídeo na memória. Os autores destacam como ponto forte do trabalho, o fato do uso do RT-LOTOS proporcionar um melhor entendimento dos comportamentos temporais críticos da aplicação e a possibilidade de efetuar uma verificação formal com características de diferentes plataformas.

Em [14] é apresentado um método para validação de documentos hipermídia baseado no uso de autômatos temporizados, e técnicas de *model-checking* para verificar sua consistência temporal. Nesse trabalho, o documento NCL é representado em uma linguagem chamada ROL (*Representation Objects Language*) e, uma vez obtido o modelo ROL, este é traduzido (automaticamente) para *B-timed-automata* (*Broadcaster Timed Automata*). Em seguida, é utilizado um conjunto de fórmulas em lógica temporal para verificar a consistência do documento no modelo em autômatos temporizados, usando o *Model-Checker* UPPAAL. Os autores destacam a importância do uso do contra exemplo da validação no auxilio da correção da inconsistência temporal, e consideram o uso de *B-timed-automata* adequado para este fim.

O trabalho [18] apresenta uma proposta de ferramenta para validação de documentos NCL. Tal ferramenta está sendo desenvolvida e usa TPN *(Time Petri Nets)* como formalismo na representação do documento NCL. Para a representação das propriedades a serem avaliadas esta ferramenta usa a linguagem VTS *(Visual Timed Scenarios)*. Após o obtenção do modelo em TPN e a especificação das propriedades em VTS, estas propriedades são transformadas com o uso da ferramenta de tradução VTS2TINA, e TINA é usado como ferramenta de *model checking*. Como trabalho futuro, os autores pretendem incluir a verificação da interação com dispositivos e *scripts* na linguagem Lua.

7. CONCLUSÕES E PERSPECTIVAS

Este artigo apresentou uma abordagem MDE para verificação de documentos interativos. Conforme foi apresentado, a abordagem MDE proposta permite a transformação de um modelo de concepção de alto nível (documento NCL) em uma descrição formalizada dos aspectos de comportamento e temporais desse mesmo modelo (especificação FIACRE). A vantagem da abordagem é possibilitar que diferentes equipes envolvidas no processo de concepção de aplicações interativas trabalhem com ferramentas e modelos adequados ao seu domínio.

A descrição FIACRE associada a um documento hipermídia possibilita que propriedades desejáveis para sua apresentação (não bloqueio da apresentação, ausência de conflitos de recursos, etc) sejam verificadas na fase de concepção.

Atualmente, a transformação de NCL para FIACRE e a especificação das fórmulas de lógica temporal, que expressam as propriedades a serem testadas, são feitas manualmente. Entretanto, o objetivo da abordagem proposta é que a transformação seja automática, e que as fórmulas sejam geradas a partir de especificações dos *designers*, dentro do seu domínio de conhecimento. Outro objetivo da cadeia proposta é que os contra exemplos, resultantes do processo de validação, possam ser apresentados aos *designers* (dentro de seu domínio de conhecimento) de modo a orientá-los na correção de possíveis problemas.

No que se refere a comparação com outros trabalhos, a abordagem proposta, por usar uma linguagem intermediária, permitirá que outras ferramentas de verificação possam vir a serem adotadas, além da ferramenta TINA. A abordagem, por usar uma metodologia baseada em MDE, também permitirá que futuras funcionalidades, a ser includas no NCL, possam ser traduzidas para FIACRE, e que as novas versões

do tradutor sejam corretas por construção.

O sincronismo de uma aplicação NCL é controlado por mecanismos de causalidade definidos pelos conectores. A vantagem dessa abordagem é que qualquer restrição de sincronização (inclusive envolvendo a interação do usuário) pode ser descrita em NCL, desde que um novo conector que contemple esse relacionamento seja adicionado à base de conectores. A desvantagem é que o uso de relações de causalidade para modelar a dinâmica da apresentação faz com que a noção de posicionamento das apresentações dos objetos no tempo seja perdida. O problema pode ser minimizado com o uso de uma visão temporal da estrutura do documento, mas isso nem sempre é possível, principalmente nos casos envolvendo eventos não previsíveis. Assim, cenários de sincronização intrinsicamente inconsistentes de difícil detecção pelos autores poderiam ser criados em documentos NCL com número elevado de nós, âncoras e elos. O problema de verificação de consistência pode ser ampliado considerando-se que um documento NCL pode ser exibido em múltiplos dispositivos, e que outras propriedades de consistência extrínseca poderiam ser violadas, tais como apresentação simultânea de dois segmentos de áudio concorrentes [17]. Como perspectiva, o trabalho apresentado deve evoluir no sentido de permitir a inclusão de restrições ligadas ao uso conflitante de regiões da tela, restrições dos dispositivos de apresentação e relacionada às boas propriedades de usabilidade na análise da consistência do documento.

8. REFERÊNCIAS

[1] B. Berthomieu, J.-P. Bodeveix, P. Farail, M. Filali, H. Garavel, P. Gaufillet, F. Lang, and F. Vernadat. FIACRE: an intermediate language for model verification in the topcased environment. *4th European Congress on Embedded Real Time Software - ERTS*, 2008.

[2] B. Berthomieu, P.-O. Ribet, and F. Vernadat. The tool TINA - construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research*, 14(42), 2004.

[3] Bouyakoub Samia and Belkhir Abdelkader. Smil builder: An incremental authoring tool for smil documents. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7:2:1–2:30, February 2011.

[4] M. C. Buchanan and P. T. Zellweger. Automatic temporal layout mechanisms revisited. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1:60–88, February 2005.

[5] D. Bulterman and L. Hardman. Structured multimedia authoring. *ACM Trans. Multimedia Comput., Commun. and Appl.*, 1(1):89–109, 2005.

[6] Carlos de Sales Soares Neto, Luiz Fernando Gomes Soares, Rogério Ferreira Rodrigues and Simone Diniz Junqueira Barbosa. *Construindo Programas Audiovisuais Interativos Utilizando NCL 3.0*, 2010.

[7] B. Combemale. *Approche de metamodelisation pour la simulation et la verification de modele*. IRIT Doctorate Thesis, 2008.

[8] J.-P. Courtiat and R. C. De Oliveira. Proving temporal consistency in a new multimedia synchronization model. In *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA '96, pages 141–152, New York, NY, USA, 1996. ACM.

[9] J.-P. Courtiat, C. A. S. Santos, C. Lohr, and B. Outtaj. Experience with rt-lotos, a temporal extension of the lotos formal description technique. *Computer Communications*, pages 1104–1123, 2000.

[10] P. Farail, P. Gaufillet, A. Canals, C. Le Camus, D. Sciamma, P. Michel, X. Cregut, M. Pantel, and F. Vernadat. The TOPCASED project: Toolkit in OPen-source for Critical Aplications and SystEms development. *3th European Congress on Embedded Real Time Software - ERTS*, 2006.

[11] R. Guimarães, R. D. R. Costa, and L. G. Soares. Composer: Authoring tool for itvprograms. In *Euro iTV*, Lecture Notes In Computer Science, vol. 5066., 2008.

[12] N. Hirzalla, B. Falchuk, and A. Karmouch. A temporal model for interactive multimedia scenarios. *IEEE MultiMedia*, 2:24–31, July 1995.

[13] C. Lohr and J.-P. Courtiat. From the specification to the scheduling of time-dependent systems. In *Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems: Co-sponsored by IFIP WG 2.2*, FTRTFT '02, pages 129–146, London, UK, UK, 2002. Springer-Verlag.

[14] M. F. Felix, E. H. Haeusler and L. F. G. Soares. Validating hypermedia documents: a timed automata approach. 2002.

[15] Paulo Nazareno Maia Sampaio and Jean-Pierre Courtiat. An approach for the automatic generation of rt-lotos specifications from smil 2.0 documents. *J. Braz. Comp. Soc.*, 9(3):39–51, 2004.

[16] C. A. S. Santos and J.-P. Courtiat. Da Análise de Consistência à Formatação Temporal de um Documento Hipermídia Usando RT-LOTOS. *SBMÍDIA*, pages 63–78, 2000.

[17] C. A. S. Santos, L. F. G. Soares, G. L. de Souza, and J.-P. Courtiat. Design methodology and formal validation of hypermedia documents. *Proceedings of the sixth ACM international conference on Multimedia - MULTIMEDIA '98*, pages 39–48, 1998.

[18] Sergio Yovine, Alfredo Olivero, Daniel Monteverde, Laura Cordoba and Gabriel Reiter. An approach for the verification of the temporal consistency of ncl applications. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, Belo Horizonte, Brasil, Outubro 2010.

[19] L. Soares, R. Rodrigues, R. Cerqueira, and S. Barbosa. Variable and state handling in ncl. *Multimedia Tools Appl.*, 50, December 2010.

[20] L. F. G. Soares and S. D. J. Barbosa. *Programando em NCL 3.0: desenvolvimento de aplicações para o middleware Ginga, TV Digital e Web*. Elsevier, Rio de Janeiro, Brazil, 2009.

[21] T. Tsang and R. Lai. Specification and verification of multimedia synchronization scenarios using time-estelle. *Softw. Pract. Exper.*, 28:1185–1211, September 1998.

[22] R. Willrich, P. de Saqui-Sannes, P. Sénac, and M. Diaz. Hypermedia document design using the htspn model. In *MMM*, pages 151–166, 1996.