PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 33/09

Nested Context Language 3.0

Reúso e Importação

Luiz Fernando Gomes Soares
Carlos de Salles Soares Neto

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 33/09 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena Março/2009

Nested Context Language 3.0
Reúso e Importação

Luiz Fernando Gomes Soares
Carlos de Salles Soares Neto

Laboratório TeleMídia DI – PUC-Rio
Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ - 22451-900.

{lfgs, csalles}@telemidia.puc-rio.br

Abstract. *NCL, standard declarative language of the Brazilian Terrestrial Digital TV System and ITU-T Recommendation for IPTV Services, provides a high level of reuse in the design of hypermedia applications. Not only static code reuse is possible, speeding up the application design time and minimizing the probability of programming errors by reusing already tested code spans, but also the reuse of code spans in execution, making easier the application understanding and the definition of relationships among application's components. Moreover, NCL allows not only the reuse inside the same application, but also the reuse among applications, besides the reuse of code spans stored in libraries external to the application. This paper discusses all the features provided by the language and implicitly proposes a methodology to take profit of these features.*

Keywords: *Ginga-NCL, Middleware, DTV, NCL, SBTVD-T.*

Resumo. *NCL, linguagem padrão do middleware declarativo do Sistema Brasileiro de TV Digital Terrestre e Recomendação ITU-T para Serviços IPTV, oferece um alto grau de reúso para projeto de aplicações. Não apenas o reúso de código estático é possível, acelerando o tempo de desenvolvimento de aplicações e minimizando a probabilidade de erros de programação com a reutilização de trechos de código bem testados, mas também o reúso de código em execução, em diversas partes de uma aplicação, tornando-a mais fácil de ser compreendida e muito mais fácil a definição de relacionamentos entre seus objetos. Mais ainda, NCL permite não apenas reúso em uma única aplicação, mas reúso entre aplicações, além da possibilidade de reúso de componentes definidos em bibliotecas externas à aplicação. Este relatório traz uma discussão de todas essas facilidades apresentadas pela linguagem, junto com uma proposta implícita de boas práticas de desenvolvimento de aplicações.*

Palavras chave: *Ginga-NCL, Middleware, DTV, NCL, SBTVD-T.*

Nested Context Language 3.0
Reúso e Importação

Laboratório TeleMídia

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rua Marquês de São Vicente, 225, Prédio ITS - Gávea

22451-900 – Rio de Janeiro – RJ – Brasil

http://www.telemidia.puc-rio.br

Table of Contents

Nested Context Language 3.0
Reúso e Importação

Luiz Fernando Gomes Soares
Carlos de Salles Soares Neto

Laboratório TeleMídia DI – PUC-Rio
Rua Marquês de São Vicente, 225, Rio de Janeiro, RJ - 22451-900.

{lfgs, csalles}@inf.puc-rio.br

Resumo. *NCL, linguagem padrão do middleware declarativo do Sistema Brasileiro de TV Digital Terrestre e Recomendação ITU-T para Serviços IPTV, oferece um alto grau de reúso para projeto de aplicações. Não apenas o reúso de código estático é possível, acelerando o tempo de desenvolvimento de aplicações e minimizando a probabilidade de erros de programação com a reutilização de trechos de código bem testados, mas também o reúso de código em execução, em diversas partes de uma aplicação, tornando-a mais fácil de ser compreendida e muito mais fácil a definição de relacionamentos entre seus objetos. Mais ainda, NCL permite não apenas reúso em uma única aplicação, mas reúso entre aplicações, além da possibilidade de reúso de componentes definidos em bibliotecas externas à aplicação. Este relatório traz uma discussão de todas essas facilidades apresentadas pela linguagem, junto com uma proposta implícita de boas práticas de desenvolvimento de aplicações.*

1. Introdução

O reúso de código ou reúso de software é um tema recorrente nas áreas de linguagens de programação e engenharia de software e é uma preocupação desde os primórdios da programação. O principal objetivo do reúso de código é acelerar o tempo de desenvolvimento com a reutilização de certos trechos de código bem testados. Como resultado, o reúso é um instrumento que assegura uma maior qualidade final do software, já que promove a utilização de partes de código mais confiáveis e dá menos margem para erros.

Linguagens para sistemas paralelos permitem ainda o reúso de códigos em execução, por diferentes fluxos de controle paralelos de uma aplicação, resultando em diferentes tipos de comportamento em cada um dos fluxos.

A área de engenharia de software é bastante robusta com relação a metodologias e técnicas que promovem o reúso de código (ou apenas reúso de agora em diante). Tipicamente e historicamente, tais abordagens sempre estiveram bem mais presentes em linguagens imperativas, apesar de serem igualmente úteis em linguagens declarativas. O paradigma de programação imperativo se baseia na especificação de código como uma sequência de sentenças ou comandos para a resolução de um problema. De forma diferente, no paradigma declarativo é expressa a intenção final, ou seja, o que é o problema a ser tratado e não como ele deve ser resolvido. Linguagens declarativas são em geral projetadas para um determinado domínio específico. Isso torna mais difícil o projeto de linguagens declarativas com maior reúso já que a promoção do reúso deve também ser conciliada com o foco para o qual a linguagem foi projetada.

Este trabalho está especificamente associado ao reúso quando do emprego do paradigma declarativo para a autoria de documentos hipermídia, como aqueles voltados para a confecção de programas interativos para TV digital. Um documento hipermídia é responsável pela especificação dos objetos de mídia (vídeo, áudio, texto etc.) que compõem uma aplicação e pela especificação dos relacionamentos entre esses objetos. Nesses documentos, o paralelismo na execução das aplicações é inerente. Assim, a tarefa de criação de documentos hipermídia possui várias facetas em que é desejável a promoção de um maior grau de reúso para o autor, nos dois aspectos levantados pelos dois primeiros parágrafos desta seção, como exemplificado a seguir.

Em primeiro lugar, o próprio conteúdo dos diversos objetos de mídia pode precisar ser reusado. É desejável para um autor que ele edite suas imagens, áudios e vídeos uma única vez e possa reutilizar tais arquivos de mídia facilmente em aplicações hiperimídia diversas. O reúso de conteúdo pode se dar não apenas em termos de especificação, indicando que os objetos se referenciam a um mesmo conteúdo a ser exibido independentemente por cada exibidor de cada objeto de mídia, mas também em termos de execução, indicando o reúso de um mesmo conteúdo em apresentação por diversos exibidores de objetos de mídia.

Mais ainda, o reúso pode se dar não apenas em termos de conteúdo de um objeto de mídia, mas do próprio objeto em si. Novamente, o reúso pode se dar a nível sintático, indicando duas instâncias do mesmo código que especifica o objeto (exibição independente dos objetos em duas instâncias de representação), ou pode significar o reúso da mesma instância de representação do objeto.

Um poderoso mecanismo para aperfeiçoar a usabilidade de aplicações é a adoção de padrões recorrentes de interface. Em aplicações interativas de TV digital, por exemplo, é fortemente recomendado o reúso do posicionamento de objetos em um dispositivo de exibição. Ainda sobre o leiaute, é desejável que tais aplicações determinem de forma separada não apenas *onde* (posicionamento na tela), mas *como* cada mídia vai ser inicialmente exibida. Sempre que houver coincidência na definição das regiões que particionam o dispositivo de exibição, é possível reusar tal especificação e enriquecê-la com a definição de *como* as mídias deverão ser exibidas, e essas definições também são altamente passíveis de reúso. Apenas a definição das regiões não é suficiente para definir o leiaute, uma vez que autor ainda pode precisar definir, por exemplo, com que grau de transparência a mídia deve ser exibida, ou mesmo informar parâmetros específicos de exibição de um tipo de mídia, como o tamanho da fonte de um texto.

Ao compor um documento multimídia, um autor precisa ter algum mecanismo na linguagem para descrever a organização hierárquica de seus objetos de mídia, a estrutura do documento. Como exemplo, suponha uma série de TV interativa. A organização natural desse documento passa por definir a série de TV como um conjunto de episódios e cada episódio contendo uma sequência de blocos e cada bloco englobando um conjunto de cenas. É conveniente, que essa estrutura, ou parte dela (uma sub-estrutura) possa ser facilmente reusada para facilitar inclusive a elaboração de novos episódios.

É também desejável que um documento multimídia possa ser facilmente adaptado de acordo com determinadas regras. Tais regras podem, por exemplo, levar em conta as restrições de processamento e memória do dispositivo de exibição ou fazer uso de preferências do usuário. Como exemplo, um documento multimídia exibido num monitor de computador pode ser adaptado para uma exibição bem diferente na tela de um celular.

Novamente, é útil que um autor de posse de tais regras, não precise reescrevê-las em novas aplicações ou em diferentes partes do documento, apenas as reusando conforme feito anteriormente.

A especificação de documentos multimídia é composta, em sua maioria, pela definição dos relacionamentos entre os diversos objetos de mídia que compõem os documentos. Em função disso, o impacto da redução ou simplificação da especificação de relacionamentos implica uma significativa aceleração no tempo global de desenvolvimento e na diminuição de erros de programação. Como os relacionamentos, no geral, são regidos e agrupados por relações, é possível tratar tais relações como entidades de primeira classe e assim reusar sua especificação ao definir os relacionamentos. Uma outra alternativa de reúso na especificação dos relacionamentos envolve torná-las implícitas na estrutura hierárquica do próprio documento, dando semântica a cada composição hipermídia, o que embute o inconveniente de não isolar a organização hierárquica da semântica de apresentação, como discutido na seção de trabalhos relacionados.

Algo típico no suporte ao reúso é o tratamento de documentos como bibliotecas de código que são facilmente importadas por documentos diversos. Também é desejável para uma linguagem de autoria multimídia que tais bibliotecas possam ser facilmente reaproveitadas. Mais ainda, é desejável que tais bibliotecas sejam tidas como bases de informação que possam ser facilmente incorporadas numa nova aplicação, selecionando-se apenas o aspecto que se quer importar. Como exemplo, se um autor quer reusar apenas o leiaute de um documento já existente, é desejável que ele possa ater-se a importar apenas essa base de uma biblioteca declarativa e não uma aplicação por inteiro, o que traria bem mais informação do que a necessária e dificultaria o reúso.

Ainda com relação a bibliotecas de código, o próprio aninhamento de documentos é um recurso também desejável. Como exemplo, a organização previamente descrita da série de TV poderia ser feita em documentos diversos de tal forma que um documento maior corresponderia à toda série, importando todos os documentos que especificassem, um a um, os episódios da série de TV. Cada episódio, por sua vez, seria um documento próprio, que aninharia, ou conteria, vários documentos, cada um dos quais representaria um dos blocos daquele episódio, que também aninhariam os documentos que representariam as cenas, e assim sucessivamente.

Como se vê, na autoria de documentos multimídia, o reúso deve transpassar vários aspectos. Reúso em linguagens declarativas baseadas no tempo (*time-based languages*) é discutido neste relatório tendo por base o suporte oferecido na linguagem NCL (*Nested Context Language*) [2]. NCL é a linguagem padrão do Sistema Brasileiro de TV Digital Terrestre [1] e recomendação ITU-T para serviços IPTV [5]. Este trabalho visa detalhar de que forma o projeto da linguagem e a concepção de seu modelo base promovem o reúso a nível declarativo. Em especial, o presente trabalho fornece um conjunto de boas práticas para promover um maior reúso no processo de autoria em NCL.

O relatório está organizado como se segue. A Seção 2 apresenta como se dá o reúso de código em um mesmo documento NCL. A Seção 3 descreve como se dá a importação e reúso de código entre documentos NCL diferentes. A Seção 4 fornece uma discussão sobre trabalhos relacionados, e a Seção 5 apresenta as considerações finais.

2. Reúso em um Documento NCL

Esta seção descreve como o projeto da linguagem NCL, separando vários aspectos do processo de autoria, é um mecanismo natural para a promoção do reúso.

2.1. Reúso de Conteúdo

Em NCL um objeto de mídia contém interfaces (propriedades e pontos de entrada em parte do seu conteúdo), um conteúdo e a forma com que deve ser apresentado (*onde* e *como*). Autores de documentos NCL definem o conteúdo de um objeto de mídia por *referência*, através de seu atributo *src*, à sua localização, esteja ele em um servidor, em um fluxo contínuo de bits ou sendo recebido por carrossel por meio de um protocolo para envio de dados sem solicitação (*pushed data*). Dessa forma, um mesmo conteúdo pode ser referenciado e reusado mais de uma vez em diferentes objetos, que recebem identificadores diferentes a cada referência. Como resultado dessa separação do conteúdo, uma aplicação NCL completa é formada pelo documento NCL que a descreve estruturalmente e semanticamente, e do conteúdo de cada mídia isoladamente.

Em geral, dois objetos de mídia com o mesmo valor de atributo *src*, para localização de seus conteúdos, terão suas exibições independentes, isto é, se os objetos são instanciados em momentos diferentes, eles terão o mesmo conteúdo apresentado em momentos diferentes. Assim, se, por exemplo, um vídeo de 100 segundos de duração for referenciado por dois objetos de mídia cujas apresentações são iniciadas 50s defasadas uma em relação à outra, ele terá duas apresentações diferentes, cada uma com a duração de 100 segundos e em paralelo durante os 50 segundos iniciais de um objeto e 50 segundos finais do outro.

Entretanto, a NCL também permite que, além de conter a mesma referência a um conteúdo, um objeto de mídia especifique que seu conteúdo é um espelho de outro objeto. Nesse caso, pode-se ter ainda duas apresentações em paralelo, mas idênticas. No caso do exemplo anterior do vídeo, o objeto de mídia que fosse iniciado 50 segundos depois teria a duração de sua exibição apenas de 50s, começando pela metade.

A Figura 1 ilustra o primeiro caso citado de reúso com os objetos “video1” e “video2”, e o segundo caso com os objetos “vídeo 2” e “vídeo 3”. O atributo *descriptor* é visto na próxima seção.

```
<media id="video1" src="../media/film.mp4" descriptor="descVideo1"/>

<media id="video2" src="../media/film.mp4" descriptor="descVideo2"/>

<media id="video3" src="ncl-mirror://video1"descriptor="descVideo3"/>
```

Figura 1 – Reúso de conteúdo.

2.2. Reúso de Leiautes

Um aspecto importante e que consome um tempo considerável no projeto de documentos multimídia em NCL é a definição espacial do posicionamento dos objetos de mídia. Outra característica relacionada é a definição de que forma um objeto de mídia deve ser inicialmente exibido. Esses dois aspectos de uma apresentação NCL são especificados, cada um deles, como entidades de primeira classe no cabeçalho de um documento NCL, respectivamente através das bases de regiões e de descritores.

Cada base de regiões permite que se divida a tela de um dispositivo de exibição em um conjunto de regiões retangulares. Tais regiões podem estar inclusive sobrepostas umas sobre as outras, caso em que a própria região carrega a informação de qual delas deve sobrepor a outra, por meio de um atributo zIndex. Geralmente um documento NCL possui apenas uma base de regiões, mas cada uma delas pode ser associada a um dispositivo de exibição diferente. Esse mecanismo garante o suporte nativo na linguagem ao uso de múltiplos dispositivos de exibição, o que significa que uma mesma apresentação pode sincronizar objetos de mídia sendo apresentados em mais de um dispositivo. É conveniente ressaltar que a forma como se dá a comunicação entre tais dispositivos foge ao escopo declarativo da linguagem e não é preocupação do autor.

A Figura 2 exemplifica um cenário com dois dispositivos de exibição. O dispositivo principal (definido nas linhas 1 a 6) é representado pela região “fullScreenRg” (linhas 2 a 5). Tal região é particionada em duas regiões filho, a primeira delas “centerRg” que corresponde à metade da tela centralizada e a outra “buttonRg”, onde um botão vai ser exibido no canto superior direito. As regiões do dispositivo de exibição secundário (identificado na linguagem por “systemScreen(2)”) são especificadas nas linhas 7 a 11. Tal dispositivo é representado pela região “remoteControlRg”, que possui uma região filho “merchandizingRg” ocupando o quadrante superior esquerdo da tela.

```
1:  <regionBase>
2:     <region id="fullScreenRg">
3:        <region id="centerRg" left="25%" top="25%" width="50%"
    height="50%"/>
4:        <region id="buttonRg" right="2%" top="2%" width="5%"
    height="5%"/>
5:     </region>
6:  </regionBase>
7:  <regionBase device="systemScreen(2)">
8:     <region id="remoteControlRg">
9:        <region id="merchandizingRg"height="50%" width="50%"/>
10:    </region>
11: </regionBase>
```

Figura 2 – Bases de regiões em múltiplos dispositivos.

O cabeçalho da Figura 2 poderia ilustrar um cenário em que um vídeo é exibido ao centro de uma TV, ocupando a metade da área da tela (região “centerRg”). Em um certo momento do vídeo, um ícone interativo apareceria no canto superior direito da TV (região “buttonRg”). Se um botão de interatividade fosse pressionado através de um controle remoto com tela de LCD, uma propaganda seria exibida naquele controle remoto ocupando o quadrante superior esquerdo de sua tela (região “merchandizingRg”). Nesse cenário, o acionamento de uma propaganda interativa não interromperia o programa de TV sendo

exibido e seria direcionada apenas ao telespectador que interagiu. Múltiplos controles remotos podem tornar mais individualizada a experiência de assistir à TV.

Cada documento NCL especifica uma base de descritores, cada um definindo *como* cada objeto de mídia deve ser inicialmente exibido. O conceito de descritor é importante porque separa do objeto de mídia as informações específicas de sua apresentação. Tem-se então um novo mecanismo de promoção de reúso, já que é muito comum a necessidade de que vários objetos de mídia se apóiem em uma mesma descrição de iniciação.

Um descritor pode ser enriquecido por parâmetros que são tratados especificamente pelo exibidor de cada tipo de mídia, deixando ainda mais versátil o suporte a novos tipos de monomídia. Alguns exemplos disso são quando se quer informar um certo grau de transparência à exibição de uma imagem ou determinar o volume do áudio.

A Figura 3 acrescenta ao cenário da Figura 2 uma base de descritores. O descritor "centerDs" aponta para a região "centerRg" com o volume de som inicialmente em 50% (linhas 2 a 4). O descritor "buttonDs" faz uso da região "buttonRg" e especifica uma transparência de 20% (linhas 5 a 7). O descritor "merchandizingDs" (linha 8), por sua vez, apenas indica o uso inicial da região "merchandizingRg", que está presente no dispositivo de exibição secundário.

```
1: <descriptorBase>
2:   <descriptor id="centerDs" region="centerRg">
3:     <descriptorParam name="soundLevel" value="0.5"/>
4:   </descriptor>
5:   <descriptor id="buttonDs" region="buttonRg">
6:     <descriptorParam name="transparency" value="0.2"/>
7:   </descriptor>
8:   <descriptor id="merchandizingDs" region="merchandizingRg"/>
9: </descriptorBase>
```

Figura 3 – Base de descritores.

2.3. Reúso de Objetos de Mídia

Ao especificar um objeto de mídia em NCL é necessário informar a localização de seu conteúdo (vídeo, texto, áudio, imagem, código imperativo, código declarativo etc.) e o descritor associado, que como visto indica como o objeto deve ser inicialmente exibido. Uma outra forma de especificar um objeto de mídia é como uma referência a um outro objeto de mídia existente. Nesse último caso, a especificação do objeto de mídia base é *reusada*.

A Figura 4 detalha ainda mais o cenário previamente introduzido. Na figura, vídeo da TV é especificado com o identificador "mainVideo", na linha 1, fazendo referência ao descritor "centerDs", que especifica o local da exibição no centro do dispositivo de exibição. O atributo *src* informa que o conteúdo deve ser obtido de um fluxo contínuo de bits, especificamente do programa 0x01, fluxo elementar 0x05. De forma similar, a linha 2 descreve o objeto de mídia "button" e a linha 3, a página HTML "merchandizing". Na linha 4 da Figura 4, o objeto de mídia "buttonRef" referencia o botão "button", previamente definido, reusando todas as suas características de iniciação, no caso aquelas definidas pelo descritor "buttonDs".

```
1: <media id="mainVideo" src="sbtvd-ts://0x01.0x05"
                                                descriptor="centerDs"/>
2: <media id="button" src="media/redButton.png" descriptor="buttonDs"/>
3: <media id="merchandizing" src="index.html"
                                              descriptor="merchandizingDs"/>
4: <media id="buttonRef" refer="button" instance="new"/>
```

Figura 4 – Reúso em objetos de mídia.

Em NCL, um contexto, representado pelo elemento <context>, agrupa outros objetos e os relacionamentos (definidos por elos) entre eles, inclusive podendo conter recursivamente outros contextos. O contexto cria uma perspectiva para cada objeto, que é um conceito parecido com a noção de escopo em linguagens de programação de propósito geral. Uma perspectiva é o caminho de aninhamento de contextos, desde o mais externo, até o objeto. O reúso de objetos de mídia permite que um mesmo objeto possa pertencer a mais de uma perspectiva e, portanto, a mais de um contexto. Essa característica é muito desejável na autoria. A única restrição é que os contextos de um caminho não podem se repetir.

Usando da terminologia NCL, o reúso de objetos pode envolver tanto o reúso de objetos de dados (apenas o reúso do código declarativo que especifica o objeto) quanto o reúso de objetos de representação (o reúso da mesma instância de apresentação do objeto de dados com seu descritor associado), caracterizando os dois tipos de reúso comentados nos dois primeiros parágrafos da Seção 1. O autor NCL pode usar o atributo *instance* para especificar se deseja o reúso do objeto de dados (*instance*=“new”) ou o reúso do objeto de representação. Nesse último caso, o reúso pode ser instantâneo (*instance*=“instSame”) ou gradual (*instance*=“gradSame”), como descrito a seguir.

Suponha que um objeto A faça reúso do objeto B. Quando o reúso é de uma nova instância (atributo *instance* igual a “new”) significa que o objeto A é uma cópia do objeto B, incluindo suas âncoras e propriedades, mas é um objeto de mídia completamente novo. Isso significa, por exemplo, que a exibição do objeto B não faz ser exibido o objeto A e vice-versa.

Quando o reúso é do objeto de representação (atributo *instance* igual a “instSame” ou “gradSame”), o autor indica que ambos os objetos, tanto o reusado e o referenciador são o mesmo. No caso de “instSame”, qualquer ação que atue sobre um dos objetos atua também sobre o outro. Como exemplo, a exibição de um dos objetos faria aparecer o mesmo conteúdo, único, usando o mesmo descritor imediatamente. No caso de “gradSame”, a diferença é que os dois objetos necessitam de receber explicitamente e individualmente uma ação para início de sua exibição para que passem a ser o mesmo objeto.

Note que no caso de reúso de objetos de representação, uma única exibição é realizada, mas como os objetos podem estar em diferentes perspectivas, eles podem participar de diferentes relacionamentos. Essa é a verdadeira utilidade do reúso de objetos de representação que se mostra muito mais útil na autoria do que o reúso de objetos de dados. Um exemplo que ilustra o uso desse recurso é quando se deseja criar vários pontos de interação em momentos diversos de um mesmo vídeo. Num determinado intervalo de tempo do vídeo, por exemplo, o usuário pode selecionar um botão para acionar um jogo interativo e em outro momento ele pode selecionar um botão para ver um comercial. Através do reúso do objeto de mídia que representa o vídeo, é possível definir cada um dos

pontos de interação (jogo e propaganda) em um contexto separado, com o vídeo presente em ambos. O objeto de mídia que representa o vídeo em cada um desses contextos pode ter apenas as âncoras importantes para o caso em questão. A modelagem desse cenário fica natural e limpa com o reúso.

2.4. Reúso de Relações

Um aspecto inovador e que expressa bem a atenção dada ao reúso no projeto da linguagem NCL é a separação entre os conceitos de relação e relacionamento [3]. O cabeçalho do documento NCL carrega uma base de relações que podem ser usadas para a definição do sincronismo espacial e temporal entre as mídias que, por sua vez, é realizado no corpo do documento por meio da especificação de relacionamentos. Essa separação simplifica e acelera a definição de relacionamentos por meio do reúso da relação. Em NCL, as relações são descritas por um conector (elemento <causalConnector>. O conector define papéis e uma sentença causal. Tal sentença expressa que, dado que certas condições são satisfeitas por alguns dos papéis, então determinadas ações devem ser realizadas em outros papéis. O relacionamento, expresso em NCL por elos (elementos <link>), reúsa essa sentença causal e define apenas a associação entre cada papel e as respectivas interfaces de objetos que devem exercer o papel.

Como exemplo, suponha uma apresentação multimídia em que o início da exibição de um vídeo deve disparar o início sincronizado de uma imagem. Adicionalmente, o início de um áudio também deve sincronizadamente iniciar a exibição de uma animação. Esses dois relacionamentos distintos entre objetos de mídia diferentes se baseiam em uma mesma relação. Tal relação expressa a sentença causal “quando alguém com um papel X for iniciado, então inicie também alguém com o papel Y”. Com base nessa sentença, as definições dos dois relacionamentos anteriores diferem apenas em que objetos exercerão os papéis: no primeiro caso o papel X é realizado pelo vídeo e o papel Y é feito pela imagem. No segundo relacionamento, X e Y são exercidos, respectivamente, pelo áudio e pela animação.

Um caso de uso importante é quando NCL sincroniza objetos que representam código imperativo. Isso é particularmente útil quando a aplicação multimídia que se quer descrever foge ao escopo declarativo de NCL como, por exemplo, exigindo alguma computação específica (o cálculo de uma fórmula ou o desenho de um gráfico). Nesse caso, novamente o conteúdo é isolado do documento multimídia da mesma forma que em qualquer outro tipo de monomídia. Um uso comum dessa abordagem em aplicações interativas para TV digital é através de objetos de mídia NCLua, que são scripts escritos em Lua e que executam sincronizados com outros objetos de mídia graças a elos multimídia descritos em NCL. Esse recurso permite, por exemplo, que se definam componentes gráficos (ou widgets), tais como caixas de texto ou painéis de opção, uma única vez como código Lua. Cada componente gráfico poderia ser reusado e parametrizado em documentos NCL como um objeto de mídia orquestrado tal como qualquer outro.

2.5. Reúso de Estruturas

O conceito de objeto de contexto, previamente definido, é útil tanto para organizar estruturalmente o documento quanto também para encapsular sua semântica de

apresentação. Na prática, o contexto não é apenas um container para outros objetos, mas também reúne elos, o que dá a ele uma semântica de apresentação.

A organização coerente de documentos NCL em contextos é uma boa prática de uso da linguagem. Há exemplos diversos que exploram sua utilidade, alguns já mencionados neste relatório.

Um contexto pode ser reusado por outro contexto. Nesse caso o reúso é sempre de código, de objetos de dados como definido anteriormente para objetos de mídia, sempre criando uma cópia do contexto original. Ambos os contextos são novas instâncias, completamente independentes. Em um reúso de contexto, se há objetos de mídia internos que são a mesma instância ("instSame" ou "gradSame"), o comportamento segue a mesma lógica anteriormente apresentada. Isso favorece o reúso da estrutura hierárquica do documento, mantendo a semântica de apresentação de seus objetos de mídia internos.

2.6. Bases de Regras e de Transições

O cabeçalho de um documento NCL pode definir bases de transições e de regras, como já mencionado. As transições são recursos visuais aplicados no início ou no término da exibição de um objeto de mídia. As regras formam um dialeto lógico que pode ser usado no suporte à adaptação de conteúdo da linguagem.

A Figura 5 apresenta uma base de transições (linhas 2 a 4). A linha 3, um exemplo de efeito de transição "fadeIn", correspondente ao aparecimento gradual por 3 segundos do conteúdo, é apresentado.

Assim como na separação entre a região e o descritor, a separação dos efeitos de transição do descritor, constituindo uma base própria de informação, permite um maior reaproveitamento dessa especificação. É muito comum a definição de uns poucos efeitos de transição sendo reusados em diversos descritores.

Por sua vez, as regras formam uma base que expressa os critérios que permitem a adaptação de documentos [4]. As regras podem ser simples ou compostas. Regras simples são formadas por uma comparação lógica entre uma variável e um valor ou entre duas variáveis. Regras compostas são formadas por um "E" lógico ("*and*") ou um "OU" lógico ("*or*") entre regras compostas ou simples.

A Figura 5 apresenta uma regra composta "canPlayPtVideoR" que testa se o dispositivo de exibição pode exibir um vídeo em português (linhas 6 a 10). Isso é feito testando se a linguagem do sistema é português (regra "ptR", linha 7) e também se a capacidade de processamento é superior a 0,4 MPUs (regra "processingR", linha 8).

As regras podem ser usadas por objetos de alternativa (elemento <switch>) em NCL. Objetos de alternativa também são passíveis de serem reusados, conforme as mesmas definições que regem os objetos de contexto. Simplificadamente, um objeto de alternativa mapeia ou seleciona um objeto interno que passa a representá-lo.

Na Figura 5, é definido o objeto de alternativa "merchandizing. A linha 14 especifica que se o dispositivo de exibição puder apresentar um vídeo em português, o que foi

previamente descrito pela regra “canPlayPtVideoR”, o objeto interno a ser selecionado é o “videoMerchandizing” (definido na linha 16). Caso contrário, o objeto interno padrão a ser selecionado é o formulário “formMerchandizing” (linha17).

```
1:  <head>
2:     <transitionBase>
3:        <transition id="fadeIn" type="fade" dur="3s"/>
4:     </transitionBase>
5:     <ruleBase>
6:       <compositeRule id="canPlayPtVideoR" operator="and">
7:         <rule id="ptR" var="system.language" comparator="eq"
                                                        value="pt"/>
8:         <rule    id="processingR"    var="system.CPU"    comparator="gt"
                                                        value="0.4"/>
9:       </compositeRule>
10:    </ruleBase>
11: </head>
12: <body>
13:    <switch id="merchandizing">
14:       <bindRule rule="canPlayPtVideoR"
                                    constituent="videoMerchandizing"/>
15:       <defaultComponent component="formMerchandizing"/>
16:       <media id="videoMerchandizing" src="merchandizing.mp4"
                                       descriptor="merchandizingDs"/>
17:       <media id="formMerchandizing" src="index.html"
                                       descriptor="merchandizingDs"/>
18:    </switch>
19: </body>
```

Figura 5 – Base de transições e base de regras para um objeto de alternativa.

3. Reúso entre Documentos NCL

Em todos os casos previamente mencionados, o reúso foi realizado dentro do escopo de um mesmo documento NCL. É possível, contudo, importar bases de informações especificadas em um outro documento e reusar seus componentes. Além disso, mesmo parte ou o todo da estrutura de um outro documento é passível de ser importada e reusada. Isso permite que se organize ainda mais a estrutura de apresentação em documentos diversos.

3.1. Documentos NCL Aninhados

Um documento inteiro pode ser reusado em um documento que o importa, o que reaproveita toda sua estrutura de composição e de apresentação. Há duas formas de fazer esse reúso: através da importação do documento original como um contexto de um novo documento; ou ainda espelhando um documento existente como um objeto de mídia de um novo documento.

A importação de um documento A como um contexto em um documento B permite que se crie uma cópia lógica do documento A. Esse contexto, que, como sempre em reúso de contextos, é uma cópia, é passível de sofrer ações derivadas de relacionamentos (elos NCL) como qualquer outro objeto. A referência ao documento importado é feita no formato alias#docID (indicando um *alias* arbitrário ao documento importado, que é informado no cabeçalho do documento importador) e o identificador (atributo *id*) do documento importado, indicando que do documento se quer reusar todo o seu corpo (como se verá na Seção 3.3, partes de um documento também podem ser reusadas em outro documento).

A Figura 6(a) apresenta um documento NCL, cujo identificador é “A”. Seu conteúdo está propositalmente omitido sem prejuízo de entendimento. A Figura 6(b) mostra como é a importação do documento A, definido com *alias* “docA” no escopo do novo documento “B” (linha 5). Um contexto “referDocA” referencia o documento “docA#A”. Isso define tal contexto como uma cópia passível de ser manipulada normalmente por elos e contendo os mesmos pontos de interface do documento importado.

O reúso de um documento pode também ser feito pela utilização de um objeto de mídia contendo código NCL (do tipo “application/x-ncl-NCL”), esse recurso é utilizado quando se quer aninhar a apresentação de um documento NCL em uma região de um outro documento, como se a apresentação do documento importado fosse de um tipo de mídia primitivo. A Figura 6(C) descreve o reúso do documento A por meio do objeto de mídia “mirrorDocA” na região apontada pelo descritor “centerDs”. Na prática, quando esse objeto de mídia é exibido, o documento A é exibido naquela região respeitando as regras normais de sobreposição de regiões.

(a) Documento A.ncl

```
1: <?xml version=”1.0” encoding=”ISO-8859-1”?>
2: <ncl id=”A” (...) >
3:    (...)
4: </ncl>
```

(b) Importando o Documento A em B.ncl

```
1: <?xml version=”1.0” encoding=”ISO-8859-1”?>
2: <ncl id=”reuse-NCL” (...) >
3: <head>
4:    <importedDocumentBase>
5:       <importNCL documentURI=”A.ncl” alias=”docA”/>
6:    </importedDocumentBase>
7: </head>
8: <body>
9:    (...)
10:      <context id=”referDocA” refer=”docA#A”/>
11:   </body>
12:   </ncl>
```

(c) Importando o Documento A para uma região em C.ncl

```
1: <?xml version=”1.0” encoding=”ISO-8859-1”?>
2: <ncl id=”C” (...) >
3: <body>
4:    <media type=”application/x-ncl-ncl” id=”mirrorDocA” refer=”docA#A”
                                                  descriptor=”centerDs”/>
5:    (...)
6: </body>
7: </ncl>
```

Figura 6 – Documentos NCL aninhados: (a) Documento A.ncl; (b) Importando o Documento A em B.ncl; (c) Importando o Documento A para uma região em C.ncl.

3.2. Importação de Bases

A Seção 2 apresenta diversas bases presentes no cabeçalho de documentos NCL relacionadas a informações específicas do processo de autoria, como o leiaute do documento, os efeitos de transição ou as relações hipermídia. Qualquer base é passível de ser importada, por meio do uso do elemento <importBase> de NCL. Tal elemento possui o atributo *documentURI*, especificando o caminho absoluto ou relativo para o documento a que se quer importar e possui também o atributo *alias*, fornecendo um identificador de referência para aquela base.

Bases de regiões e descritores podem ser definidas em arquivos separados e importadas para o cabeçalho de outros documentos como se fossem uma biblioteca de leiaute e de formas de apresentação. Isso é particularmente útil em famílias de aplicações similares para estabelecer um padrão único para todos os documentos daquela mesma família. Um exemplo típico é quando há várias aplicações interativas de TV de um mesmo telejornal. Nesse caso, é conveniente que todas as aplicações partilhem de um mesmo padrão de interface para facilitar e fixar melhor a usabilidade pelos telespectadores.

Bases de efeitos de transição podem ser importadas e referenciadas nos descritores do documento que as importam. Esse é um procedimento habitual já que é de praxe utilizar um

mesmo conjunto pequeno de efeitos de transição para não criar apresentações cansativas ou repetitivas visualmente. Criar um documento com esse conjunto restrito de efeitos facilita o posterior reaproveitamento.

As bases de regras formam um vocabulário de sentenças lógicas que também pode ser reaproveitado entre documentos. Há uma tendência natural que os mesmos critérios previamente utilizados por um autor sejam novamente requisitados em outros de seus documentos.

Bases de conectores permitem que as relações hipermídia sejam definidas como entidades de primeira classe para posterior uso na especificação dos relacionamentos entre as mídias. A importação de bases de conectores e reúso de seus componentes é a prática mais comum no projeto de aplicações NCL. Usuários de NCL costumam fazer uso de uma base de concetores ampla com identificadores conhecidos sem precisarem se preocupar em como os conectores são especificados. A tarefa de construção de conectores fica restrita a usuários mais experientes, uma vez que, de fato, raramente é necessário criar relações diferentes das mais usuais [1]. A importação e reúso de conectores não só têm as vantagens inerentes ao reúso, por demais mencionadas, mas também, colateralmente, apoiam o aprendizado da linguagem NCL e permitem o desenvolvimento de aplicações poderosas, mesmo por usuários pouco especializados.

3.3. Importação de Objetos Especificados em Outros Documentos

A Seção 3.1 discute a importação e reúso do corpo de um documento por inteiro, no entanto, partes de um documento importado também podem ser reusadas: um contexto, um objeto de alternativa, ou um objeto de mídia interno ao documento. Nesse caso, o reúso é feito da mesma forma que quando um documento inteiro é importado, bastando se referenciar ao identificador daquele objeto que se pretende reusar. Isso dá uma versatilidade ainda maior para o mecanismo de importação, uma vez que qualquer parte da estrutura de organização do documento é passível de ser reaproveitada em um outro documento.

Várias das formas de importação de documentos são exemplificadas na Figura 7. Duas bases são importadas para o documento: a base de descritores oriunda do documento “descritores.ncl” (linha 5); e a base de conectores do documento “conectores.ncl” (linha 8). Além das bases importadas, o documento “A.ncl” também é importado (linha 11).

Ainda com relação à Figura 7, no corpo do documento há a cópia do documento A com identificador “refDoc” (linha 15). Disparar a exibição de tal contexto é equivalente a tocar o documento A por inteiro. Um objeto de mídia declarativo “refCtxA” referencia um contexto específico dentro do documento A, de identificador “sameContext” (linha 16), reusando-o como uma cópia no documento atual. De forma similar, o objeto de mídia “refMediaA” referencia um objeto de mídia interno ao documento A, de identificador “sameMedia” (linha 17) como uma mesma instância imediata. O objeto de mídia “image” está relacionado ao arquivo de imagem “sameImage.png” que deve inicialmente ser exibido com base no descritor “centerDs”, importado da base de descritores do documento “descritores.ncl” (linha 18).

Um elo descreve um relacionamento entre três dos objetos de mídia previamente definidos (linhas 19 a 23). Tal relacionamento reúsa a relação “onBeginStopStart” expressa no documento “conectores.ncl”. Como se vê, a própria nomenclatura usada para dar nome ao

conector é satisfatória para que o autor possa entender e reusar a relação previamente especificada. Ao invés do relacionamento ter que especificar toda a sentença causal entre os objetos de mídia, precisando definir uma a uma suas condições e ações resultantes, basta associar a cada papel definido na relação o respectivo ator. Isso pode ser visto nas três associações entre o papel “onBegin” com o objeto de mídia “image” (linha 20), com “stop” e “refMediaA” (linha 21), e ainda com “start” e “refCtxA” (linha 22).

```
1: <?xml version="1.0" encoding="ISO-8859-1"?>
2: <ncl id="importing" (...) >
3:  <head>
4:    <descriptorBase>
5:       <importBase alias="desc" documentURI="descriptors.ncl"/>
6:    </descriptorBase>
7:    <connectorBase>
8:       <importBase alias="conn" documentURI="conectores.ncl"/>
9:    </connectorBase>
10:      <importedDocumentBase>
11:          <importNCL alias="docA" documentURI="A.ncl"/>
12:      </importedDocumentBase>
13:   </head>
14:   <body>
15:     <context id="refDoc" refer="docA#A"/>
16:     <media type="application/x-ncl-ncl" id="refCtxA"
                      refer="="docA#sameContext" descriptor="sameDs"/>
17:     <media id="refMediaA" refer="docA#sameMedia"
                                                 instance="instSame"/>
18:     <media id="image" src="sameImage.png"
                                           descriptor="desc#centerDs"/>
19:     <link xconnector="conn#onBeginStopStart">
20:       <bind component="image" role="onBegin"/>
21:       <bind component="refMediaA" role="stop"/>
22:       <bind component="refCtxA" role="start"/>
23:     </link>
24:   </body>
25:  </ncl>
```

Figura 7 – Importação de bases e documentos.

4. Trabalhos Relacionados

Essa seção apresenta de forma bem resumida as facilidades de reúso providas por linguagens com paradigma e escopo similares aos de NCL: linguagens declarativas baseadas no tempo (*time-based declarative languages*). Trabalhos relacionados ao reúso em linguagens imperativas são vários na literatura, mas muito pouco é encontrado para linguagens declarativas com o foco na apresentação de documentos hipermídia.

Scalable Vectorial Graphics (SVG) [6] é uma linguagem XML, padrão W3C, para a descrição de gráficos vetoriais em duas dimensões. Em SVG, assim como na maioria das linguagens XML, o atributo *id* possibilita nomear um elemento de forma unívoca no documento. O reúso de elementos em SVG se dá através dos elementos <def> e <use>. Ao definir formas geométricas dentro de elementos <def>, elas não são prontamente renderizadas. Para isso, é necessário definir o elemento <use> com os atributos *xlink:href*, *x* e *y*. O atributo *xlink:href* especifica qual o identificador da forma, definida dentro do elemento <def>, que deverá ser renderizada. Os atributos *x* e *y* especificam a posição onde a forma geométrica deve aparecer na tela. Adicionalmente, o elemento <def> também permite incluir elementos de *namespaces* estrangeiros, tornando a linguagem facilmente extensível.

Synchronized Multimedia Integration Language (SMIL) [7] também é uma linguagem baseada em XML e padrão W3C. SMIL permite a criação de apresentações audiovisuais, assim como NCL. Documentos SMIL, como usual, são subdivididos em duas seções principais: o cabeçalho <head> e o corpo <body>. No cabeçalho são definidos elementos relativos ao leiaute da apresentação, efeitos de transição, metadados, entre outros. No corpo são definidas as mídias e a semântica da apresentação, através, principalmente, dos containeres temporais (<par> e <seq>). Os elementos relativos ao leiaute e às transições de mídia são exemplos de como se dá o reúso em SMIL, que é feito de forma similar à NCL. Em SMIL, os elementos associados às mídias fazem referência aos elementos do cabeçalho para definir suas apresentações.

SMIL não permite reúso de objetos de mídia, mas apenas reúso de conteúdo de mídia, mesmo assim, só reúso de código e não de instâncias em exibição. SMIL também não permite o reúso de relações e de suas estruturas de composição. Enfim, reúso em SMIL só é permitido dos elementos definidos no cabeçalho do documento e de conteúdos de mídia.

Do conhecimento dos autores, nenhuma linguagem declarativa, incluindo entre elas a mais usual, HTML, oferece qualquer forma de reúso de instâncias de exibição, quando muito, oferecem reúso de código estático.

5. Considerações Finais

A estrutura geral da linguagem NCL é formada por um cabeçalho com bases diversas e um corpo com toda estrutura de organização e de apresentação do documento. Essa estrutura diz muito sobre a preocupação intrínseca com o reúso no projeto da linguagem, em que o corpo do documento recorre constantemente a informações contidas no cabeçalho. Em especial, isso demonstra como o autor de NCL é guiado, desde o primeiro contato com a própria estrutura geral da linguagem, no sentido de criar documentos com alto grau de reúso.

A separação entre relação e relacionamento hipermídia é uma importante característica de expressividade e reúso de NCL. O fato dos relacionamentos serem entidades de primeira classe, ao fazerem parte de composições (contextos) hipermídia, dá a tais composições não só a natureza de organização hierárquica, mas também de semântica de apresentação. Relacionamentos externos a um contexto podem atuar sobre ele por meio de pontos de interface bem definidos, o que significa que um contexto encapsula seus objetos e relacionamentos internos. Esse conceito, aparentemente simples, garante aos contextos uma rica potencialidade para o reúso.

O aninhamento de documentos NCL e o suporte à importação de bases tornam bem mais escalável o processo de construção de documentos complexos e grandes sem trazer prejuízos para a elaboração de documentos simples e pequenos. Os mecanismos de reúso e importação da linguagem permitem que se racionalize e compreenda mais facilmente a apresentação final, agrupando informações semânticas e acelerando o raciocínio inferencial. Essa é mais uma consequência do encapsulamento oferecido pelos contextos.

O fato da linguagem NCL separar diversos aspectos do processo de autoria em bases isoladas poderia sofrer a crítica de pulverizar em pontos diversos do documento informações relevantes sob o pretexto de proporcionar o reúso de tais bases. Na prática, não é o que ocorre. Isso porque a informação contida em cada base é auto-contida. A referência que se faz a elementos de uma base por um outro elemento ajuda a enriquecê-lo com informações que podem ser entendidas como de primeira classe e não como informações parciais, sem significado por si só.

O reúso não apenas de código estático (objetos de dados NCL), mas de códigos em execução (objetos de representação NCL) conferem à linguagem uma facilidade ímpar de reúso não encontrada em outras linguagens declarativas para concepção de documentos hipermídia, tornando uma aplicação muito mais limpa, de fácil compreensão e menos propensa a erros de programação.

O projeto da linguagem NCL e do modelo hipermídia que fornece a base semântica da linguagem foram concebidos desde o início visando conduzir o autor de uma aplicação, naturalmente, à criação de documentos com alto grau de reúso. Durante a autoria, assim que um documento NCL começa a crescer, há a tendência imediata do autor organizar o documento em contextos. Isso é suficiente para promover o reúso e estabelecer que o próprio uso continuado de NCL conduza a boas práticas de programação.

Referências

[1] ABNT NBR 15606-2:2007. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Setembro 2007.

[2] Soares, L. F. G., Rodrigues, R. F., Moreno, M. F. Ginga-NCL: the Declarative Environment of the Brazilian Digital TV System. Journal of the Brazilian Computer Society. n.4, v. 12, Março, 2007.

[3] XConnector: Extending XLink to Provide Multimedia Synchronization (Inglês). D.C. Muchaluat-Saade, R.F. Rodrigues, L.F.G. Soares. II ACM Symposium on Document Engineering - DocEng2002. McLean, USA - Novembro de 2002.

[4] Tratamento de Variáveis em Linguagens Declarativas XML Baseadas no Tempo. XIII Simpósio Brasileiro de Sistemas Multimídia e Web - WebMedia2007. Gramado, Brasil - Outubro de 2007.

[5] ITU-T Recommendation H.761, 2009. Nested Context Language (NCL) and Ginga-NCL for IPTV Services. Geneva, Abril, 2009.

[6] W3C. 2008. “Scalable Vector Graphics (SVG): XML Graphics for the Web”. Disponível em http://www.w3.org/Graphics/SVG/

[7] W3C. 2008. Synchronized Multimedia Integration Language (SMIL 3.0) W3C Recommendation. Disponível em: http://www.w3.org/TR/2008/REC-SMIL3-20081201