

4

Aplicação da Proposta a um Reprojeito do Composer

4.1.

Objetivo

Como forma de avaliar os estudos feitos neste trabalho, aplicamos nossa proposta a um reprojeito da ferramenta de autoria hipermídia Composer.

4.2.

O Composer

O Composer (Guimarães, 2007) é uma ferramenta de autoria que apóia a construção de programas usando a linguagem NCL para TV digital interativa. Ela foi uma evolução de outro sistema, chamado S2T2 (Coelho, 2004). O Composer foi criado no Laboratório Telemídia⁶, da PUC-Rio, e encontra-se na versão 2.1.1. A seguir apresentamos uma visão geral sobre o ambiente do Composer.

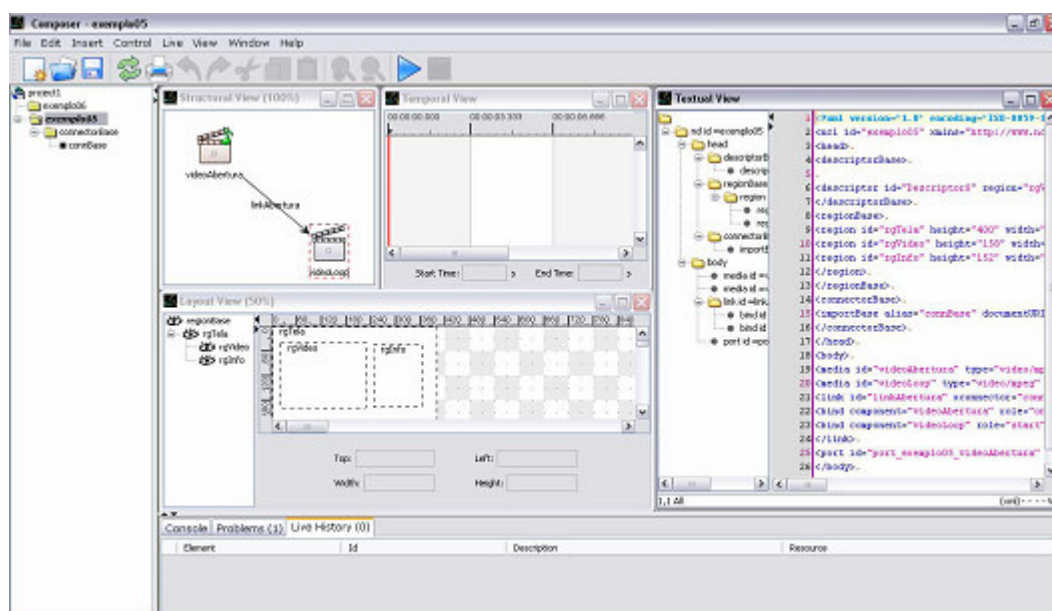


Figura 65: Ferramenta de autoria hipermídia Composer

⁶ www.telemidia.puc-rio.br

O Composer possui quatro visões: estrutural, temporal, de *layout* e textual. A visão estrutural (*Structural View*) permite a visualização dos objetos de mídias, nós de mídia, e elos, ligações entre os nós de mídia. Nesta visão, é possível, inclusive através de manipulação direta, inserir, apagar e editar os objetos do programa e suas propriedades. Os objetos de mídia são representados por um ícone gráfico e são ligados pelos elos através de arestas. As composições são representadas por pacotes, muito parecidas com a representação de pastas de um computador.

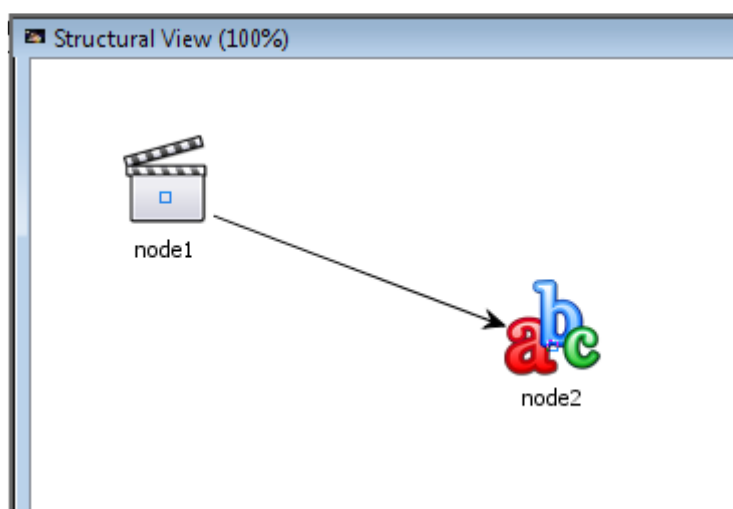


Figura 66: Visão estrutural do Composer

A visão temporal (*Temporal View*) mostra o programa que está sendo criado através de uma linha de tempo. Aqui é possível que o usuário entenda de maneira mais simplificada o sincronismo entre as mídias, do tipo, quando uma começa, quando termina, quais existem simultaneamente em um espaço do tempo etc. A interatividade também pode ser vista nesta visão. Cada elemento da NCL recebe um ícone gráfico específico para representação, o que facilita sua localização.

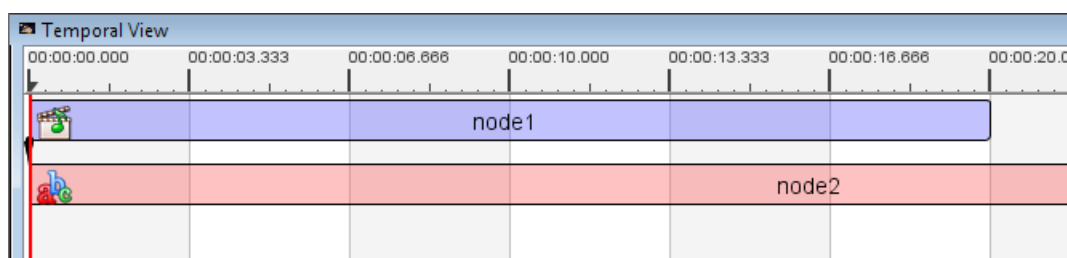


Figura 67: Visão temporal do Composer

A visão de *layout* (*Layout View*) auxilia o usuário na visualização gráfica das regiões da tela do programa. Ele pode também inserir, apagar e editar regiões

e sub-regiões. A visão textual (*Textual View*) permite que o usuário visualize e edite diretamente o código NCL do programa que está sendo criado.

Nem todas as entidades NCL, elementos da linguagem como nó objeto de mídia, nó de composição, entre outros, podem ser criadas na versão atual do Composer através da interface gráfica. Além disso, alguns problemas de interação podem ser encontrados através de uma simples inspeção deste software.

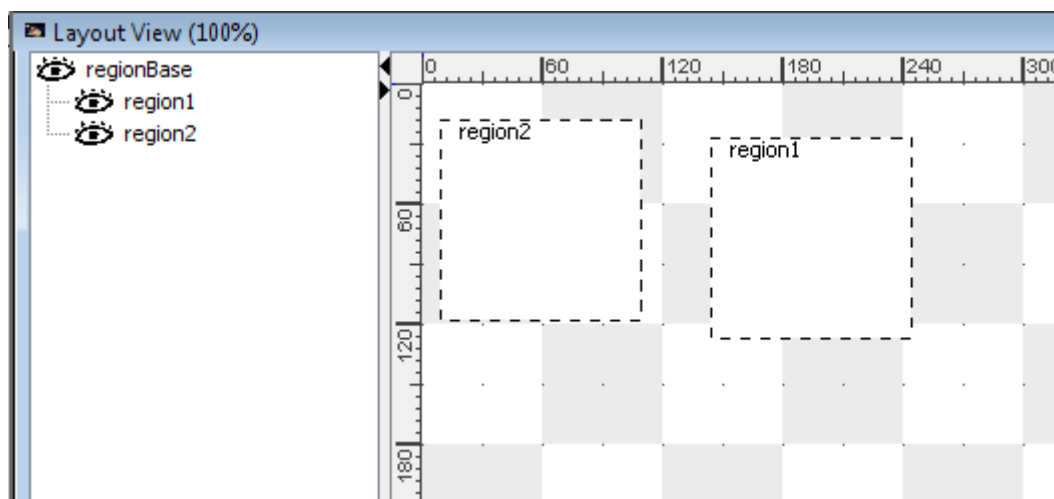


Figura 68: Visão de layout do Composer

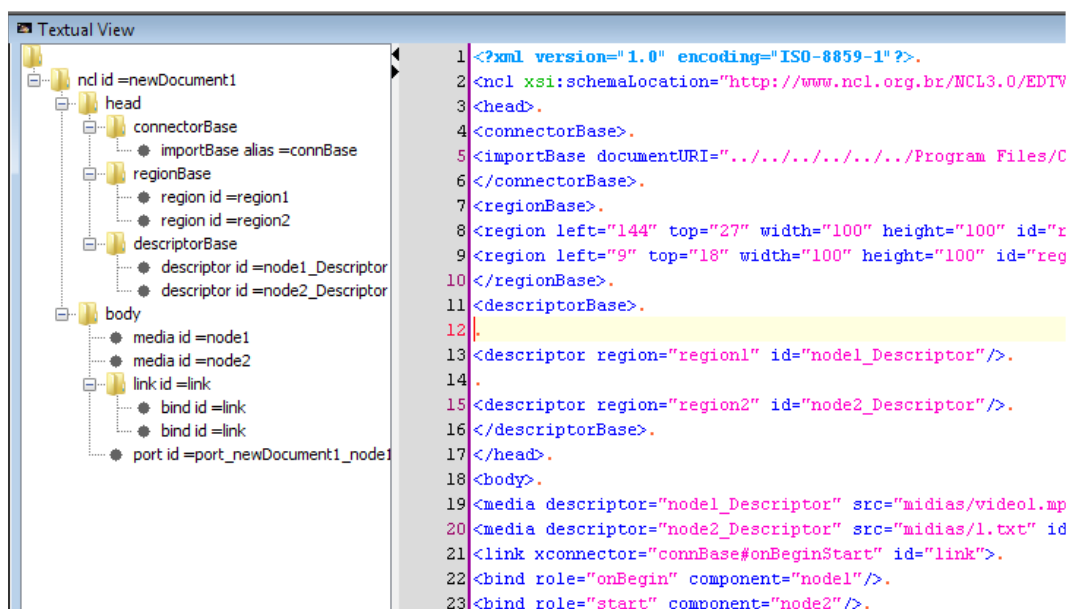


Figura 69: Visão textual do Composer

Antes de aplicar nossa proposta ao Composer, foi realizado um reprojetor como base em estudos preliminares juntos a usuários potenciais da ferramenta. Deve-se deixar claro, no entanto, que o objetivo principal deste trabalho não é

realizar um reprojeto da interface do Composer, nem estender o ambiente para cobrir toda a NCL, mas sim seguir a proposta apresentada no Capítulo 3 para incorporar alguns aspectos de adaptabilidade, de modo a obter uma avaliação inicial da nossa proposta.

4.3. Estudos Preliminares

Realizamos um estudo preliminar (Apêndice A Estudo Preliminar – Conhecendo os Usuários) para conhecer as experiências dos profissionais envolvidos na construção de programas audiovisuais, procurando entender como eles são construídos atualmente e explorar como programas audiovisuais interativos podem ser construídos no futuro. O objetivo era analisar como ocorre a interpretação e comunicação sobre o funcionamento desses programas, para entender a linguagem usada pelos profissionais envolvidos nessas atividades. Não havia interesse em avaliar o conhecimento dos participantes com relação à NCL, mas sim tentar capturar explicações sobre conceitos de programas audiovisuais interativos que possam ser relacionados aos elementos NCL, mesmo que os participantes desconheçam este relacionamento. Isto era importante para mapear o caminho a ser percorrido pelo autor do programa audiovisual interativo (usuário final da ferramenta de autoria) para atingir seu objetivo ao usar a ferramenta de autoria.

Neste estudo, foram convidados profissionais dos Departamentos de Artes e Comunicação da PUC-Rio, assim como outros da TV PUC, para participar de uma entrevista individual realizada no laboratório de Engenharia Semiótica da PUC-Rio – SERG. A entrevista foi realizada em três etapas.

A primeira consistiu em uma conversa com o participante para conhecer melhor como ele realiza seu trabalho e suas experiências. Foi elaborado um roteiro para que o entrevistador focasse as questões relevantes ao estudo. Esta etapa foi gravada utilizando um gravador de áudio.

A segunda e terceira etapas ocorreram com atividades práticas. Na segunda, o convidado assistiu a um programa (quantas vezes fosse necessário) desenvolvido em NCL, interpretado pelo formatador – o simulador da TV. Foi pedido a ele que explicasse o funcionamento deste programa, assim como todos os detalhes que fossem relevantes. O objetivo desta etapa foi capturar como o profissional se expressa com relação aos elementos que compõem o programa.

Desta forma, o profissional fornece indícios sobre como ocorre a leitura e a interpretação do programa.

Na terceira etapa, o pesquisador mostrou ao convidado alguns materiais como seqüência de fotos, textos, papéis coloridos no formato de botões, papéis em branco etc. Ele, então, pediu que o convidado construísse um determinado programa utilizando o material fornecido, narrando o que ele estava pensando e fazendo no momento. O objetivo desta etapa era analisar como ocorre o processo de concepção e montagem, a partir de uma idéia inicial e de mídias já produzidas.

Na segunda e terceira etapas, utilizou-se uma câmera para filmar as movimentações do convidado. As gravações foram permitidas pelos convidados, no momento de leitura e assinatura do termo de consentimento. Todo o material utilizado para a realização desse estudo, assim como a análise do material resultante encontram-se no Apêndice A Estudo Preliminar – Conhecendo os Usuários.

Em agosto de 2007, foi realizado na PUC-Rio o curso Oficina Ginga / NCL – Linguagens para Desenvolvimento de Conteúdo para TV Digital. Em outubro de 2007 foi realizado em várias capitais (e outras cidades) o curso Semana Ginga Brasil – Oficinas de Produção de Conteúdo Interativo para Comunidades Carentes. O conteúdo dos cursos foi o mesmo, porém o público-alvo foi bem distinto. No primeiro curso participaram diversos profissionais. No segundo participaram pessoas graduadas em diversas áreas, além de estudantes do Ensino Fundamental de comunidades carentes.

Os dois cursos foram assistidos por um pesquisador, para coletar dados sobre o aprendizado da linguagem NCL e utilização do Composer 2.3. No término do curso foi requerido aos alunos que respondessem um questionário online para contar um pouco sobre a experiência adquirida durante o curso, informar pontos positivos e negativos da ferramenta de autoria e propor recomendações de melhoria. O questionário, assim como o resultado e análise quantitativa encontra-se no Apêndice B Questionário sobre o Composer – Utilizado nos Cursos Ginga/NCL.

A partir do estudo preliminar e do resultado da análise dos questionários, definimos a análise de usuários que se encontra na seção a seguir. Além disso, como parte do projeto Ginga/NCL Composer, quatro pesquisadores analisaram o Composer (um dos quais tinha acesso aos dados do estudo preliminar) e definiram

um conjunto de objetivos dos usuários-alvo da ferramenta. Juntamente a esses objetivos, as recomendações dos usuários coletadas nas entrevistas e questionários foram destacadas e o resultado da análise de tarefas, assim como as etapas do reprojetor encontram-se na Seção 4.5.

4.4.

Resultado da Análise de Usuários e Tarefas

Perfil dos usuários

Nenhum estilo de interface agrada a todos os tipos de usuário, portanto, de acordo com as características coletadas durante as entrevistas e questionários, foi caracterizado um perfil de usuários para auxiliar na direção das idéias de design para a nova versão do Composer. Como iremos trabalhar com extensões, podemos atingir outros perfis de usuários que não foram usados como eixo das mudanças. Além disso, as idéias e/ou objetivos podem mudar com o tempo, e é por isto que precisamos tentar prever essas alterações no momento da definição das extensões. Ou seja, queremos propor mudanças na interface e interação relacionadas à versão atual do Composer para atingir um determinado grupo de usuários (e continuar contemplando esse grupo caso ela venha a sofrer alterações durante o tempo) e, possivelmente, outros grupos com perfis diferentes do escolhido.

A partir de seis características principais mostradas a seguir, formamos dois perfis de usuários. O primeiro perfil caracteriza usuários especialistas ou **conhecedores do domínio de edição de vídeos**, que estejam familiarizados com a produção de documentos audiovisuais lineares. O segundo perfil caracteriza usuários **conhecedores da linguagem NCL**, e que estejam acostumados a trabalhar com a mesma, através de outras ferramentas ou até mesmo através de outras versões do Composer.

Independente do grau de conhecimento da linguagem NCL em cada um dos dois perfis, tutoriais e sistemas de ajuda sempre deverão existir para que o uso das funcionalidades da ferramenta não seja corrompido pela falta de conhecimento deste domínio.

Papéis

A criação de programas interativos no Composer pode envolver diversas áreas. Profissionais de artes, jornalismo, comunicação social e informática são

alguns exemplos desta diversidade. Portanto, podemos ter usuários com vários papéis quando tratamos da tarefa global **criar um programa interativo**. Por exemplo, podemos ter uma ou mais pessoas responsáveis pela roteirização de um filme ou propaganda, enquanto temos outra pessoa responsável pela criação das imagens. Outra pessoa seria responsável pela sonorização, enquanto outras criariam os textos e diversos outros materiais necessários. Independente de qual papel cada uma dessas pessoas possa ter nesta tarefa, queremos construir um software que apóie qualquer uma dessas pessoas e, mais do que isto, que essas pessoas consigam realizar a tarefa global.

O usuário do Composer, então, terá apenas um papel, o de **autor** de um documento hipermídia.

Características pessoais

Pelas entrevistas, vimos que algumas pessoas gostam de ler manuais, enquanto outras preferem perguntar a outras pessoas ou participar de cursos de aprendizagem.

Características físicas

Não houve, nas entrevistas, nenhum indício de características físicas distintivas a serem consideradas.

Motivação

Apesar de existirem diversas ferramentas de autoria hipermídia, a única que apóia a construção de programas audiovisuais usando a linguagem NCL, até este momento, é o Composer. A alternativa hoje é redigir manualmente o código NCL.

E o fato de ser uma ferramenta livre, se torna mais um atrativo a sua adoção. As diferentes visões do Composer permitem adotar diferentes estratégias de construção de programas, ampliando os perfis de usuário a quem a ferramenta se destina.

Realização das tarefas

Como dito anteriormente, um perfil de usuários caracteriza o grupo de pessoas que têm experiência na construção de documentos de mídias lineares, enquanto o outro incorpora os usuários conhecedores da linguagem do domínio, a NCL.

Independente das diferenças existente na especialização da tarefa, o aprendizado e o histórico de evolução podem ser semelhantes para os dois grupos. Por exemplo, podemos ter pessoas conhecedoras da NCL mas que nunca realizaram determinada tarefa, e por isto, poderão aprender como realizá-la da mesma maneira que as que não conhecem a linguagem.

Como queremos apoiar as pessoas que possuem pouco ou nenhum conhecimento da linguagem NCL em questão, o Composer deve possuir uma estrutura para auxiliar essas pessoas, seja através de um sistema de ajuda, seja através de tutoriais, exemplos, dicas etc.

Conhecimento de ferramentas

Tanto as pessoas que responderam os questionários quanto as que participaram das entrevistas já usaram alguma vez uma ferramenta de autoria. A maioria delas usou o Adobe® Photoshop®, para editar e criar imagens e fotos. Há também as pessoas que já usaram outras versões do Composer.

Pelos questionários, vimos que a maior insatisfação com outras versões do Composer se dá devido a problemas de travamento que ocorreram, principalmente, na última versão disponível – 2.1.0. Outras pessoas incluíram como sugestão o desenvolvimento de funcionalidades para apoiar elementos da NCL que ainda não haviam sido incorporados no Composer.

De maneira geral, os usuários gostam da divisão por visões do sistema e acham que a interface facilita o entendimento sobre o que o sistema se destina a fazer. Além disto, acreditam que a estrutura dos menus é intuitiva e auxilia iniciantes.

Metamensagem

Podemos, então, reescrever a metamensagem do designer, incorporando os conhecimentos sobre os usuários.

Eis o meu entendimento de que você é um usuário de computadores que trabalha ou já trabalhou com a criação de documentos de mídias lineares, ou conhece a linguagem NCL (ou parte dela) e já construiu alguns programas audiovisuais interativos simples redigindo código NCL usando editores de texto. Aprendi que você deseja criar programas audiovisuais interativos usando a linguagem NCL. Aprendi também que você, que não conhece muito bem a linguagem do domínio, precisa de auxílio para a utilização do sistema e entendimento e aprendizado da linguagem enquanto o usa. E você, que conhece a linguagem NCL, precisará de ajuda para entender as funcionalidades do sistema.

Portanto, o sistema que criei para você requer um conhecimento apenas básico de NCL, que pode ser obtido lendo-se o tutorial de NCL. Para isso, o sistema apóia cada passo da elaboração de um programa audiovisual interativo através de diálogos e formas de interação semelhantes a aplicações que você já conhece. Além disso, como eu acredito que você tenha formas próprias de trabalhar, o sistema fornece algumas opções de adaptação ou extensão que podem tornar seu trabalho mais natural e produtivo.

4.5. Um Reprojetor do Composer

Esta seção apresenta um reprojetor parcial do Composer. Como o foco deste trabalho não é o reprojetor em si, ele ainda precisa ser devidamente avaliado.

4.5.1. Definição dos objetivos dos usuários

Como mencionamos no início deste capítulo, definimos os principais objetivos dos usuários a partir do estudo preliminar com profissionais de mídias interativas, do resultado da análise dos questionários e da observação dos alunos utilizando o Composer nos cursos Ginga/NCL e a partir do projeto Ginga/NCL/Composer juntamente com outros três pesquisadores. A Tabela 11 sumariza os objetivos elaborados pelos designers.

Projetos e documentos NCL	Objetivo 1:	criar um novo projeto;
	Objetivo 2:	abrir um projeto;
	Objetivo 3:	importar um documento NCL existente para dentro do projeto atual;
	Objetivo 4:	gerenciar os arquivos de mídia;
	Objetivo 5:	elaborar wizards ou modelos para facilitar a construção de (trechos de) documentos frequentemente utilizados;
	Objetivo 6:	inserir trecho de documento com base em wizards ou modelos (predefinidos ou definidos pelo usuário);
	Objetivo 7:	visualizar um documento em modo preview, substituindo as mídias não encontradas por outras mídias de exemplo (predefinidas conforme o tipo da mídia);
	Objetivo 8:	salvar um projeto;
	Objetivo 9:	empacotar o projeto e todos os arquivos dependentes num arquivo compactado para que possa ser reproduzido em qualquer outra máquina apenas pela descompactação do arquivo gerado.
Nós de mídia	Objetivo 10:	criar um nó de mídia;
	Objetivo 11:	modificar os atributos de um nó de mídia
	Objetivo 12:	indicar um nó de mídia para inspecionar seus atributos, propriedades e âncoras, bem como os elos em que está envolvido em papéis de condição ou de ação;
Âncoras de conteúdo	Objetivo 13:	acrescentar âncoras a um nó de mídia;
	Objetivo 14:	modificar âncoras de conteúdo de um nó de mídia (com as mesmas alternativas acima);
	Objetivo 15:	identificar quais nós têm âncoras de conteúdo definidas (nas

	diversas visões);
Âncoras de propriedade	Objetivo 16: acrescentar âncoras de propriedades a um nó;
	Objetivo 17: modificar âncoras de propriedade de um nó de mídia (com as mesmas alternativas acima);
	Objetivo 18: identificar quais nós têm âncoras de propriedade definidas (nas diversas visões);
Nós de contexto	Objetivo 19: acrescentar portas a um nó de contexto;
	Objetivo 20: remover porta(s) de um nó de contexto;
	Objetivo 21: indicar um nó de contexto para inspecionar suas portas, atributos, e propriedades, bem como os elos em que está envolvido em papéis de condição ou de ação;
	Objetivo 22: editar os atributos e as propriedades de um nó de contexto;
	Objetivo 23: identificar a estrutura do documento, ou seja, o aninhamento dos contextos;
	Objetivo 24: na visão estrutural, visualizar os nós de contexto expandidos ou colapsados;
Elos	Objetivo 25: criar elos entre nós;
	Objetivo 26: modificar elos;
	Objetivo 27: copiar elos nas diferentes visões, facilitando a sua modificação conforme as alternativas acima;
	Objetivo 28: visualizar elos nas diferentes visões, indicando os conectores que eles utilizam;
Regiões	Objetivo 29: criar uma região;
	Objetivo 30: modificar os atributos de uma determinada região;
	Objetivo 31: escalar uma determinada região;
	Objetivo 32: vincular as posições e dimensões de uma determinada região às de outra (ex: manter a largura sempre igual à de uma outra região);
	Objetivo 33: alterar o aninhamento de uma região, movendo-a para outra região pai ou mesmo para a raiz da base de regiões (na árvore ou por manipulação direta);
	Objetivo 34: permitir copiar e colar regiões;
	Objetivo 35: configurar e armazenar layouts típicos em bases de regiões para re-uso em diversos programas;
	Objetivo 36: inspecionar os atributos de uma determinada região;
	Objetivo 37: inspecionar os descritores e nós associados a uma determinada região;
	Objetivo 38: visualizar e manipular a base de regiões do documento;
Descritores	Objetivo 39: visualizar a base de descritores do documento;
	Objetivo 40: inspecionar os atributos e parâmetros de um determinado descritor;
	Objetivo 41: criar um descritor, associando parâmetros predefinidos conforme o tipo da mídia a ele associado, caso haja ou caso seja especificado pelo usuário;
	Objetivo 42: modificar os atributos e parâmetros de um determinado descritor;
Regras	Objetivo 43: inspecionar as regras do documento;
	Objetivo 44: editar regras simples ou compostas, através de diálogo;
	Objetivo 45: visualizar os elementos que utilizam regras;
Conectores	Objetivo 46: criar novos conectores, com base nos papéis de condição e ação, parâmetros de conector e atributos de condições e ações;
	Objetivo 47: modificar conectores, conforme as alternativas acima;
	Objetivo 48: inspecionar a definição de conectores
	Objetivo 49: inspecionar o uso de conectores, identificando e destacando os elos que os utilizam;
Switches	Objetivo 50: criar switches
	Objetivo 51: modificar switches, conforme as alternativas acima;
Visões	Objetivo 52: visualizar uma representação de todas as cadeias temporais do documento;
	Objetivo 53: visualizar o documento na visão espacial;

	Objetivo 54: dado um momento de visualização da visão espacial, permitir ir para qualquer outra visão com os elementos relevantes visíveis e destacados;
	Objetivo 55: visualizar e manipular o documento na visão textual;
	Objetivo 56: editar o documento e ser alertado sobre eventuais erros (classificados em warnings e erros, com sugestões sempre que possível, e indicação do local da ocorrência), sem interromper a edição (sem diálogos modais);

Tabela 11: Objetivos dos usuários para o Composer

4.5.2. Esboços de tela

Layout geral do Composer e visões

A versão atual do Composer 2.2.1 é composta por visões (estrutura, leiaute, temporal e textual). O usuário pode arrumá-las como mostra a Figura 70 – área 4 – ou trabalhar com apenas uma visão em sua forma maximizada.

A área 1 mostra a árvore de projetos, contendo a estrutura do documento e da base de conectores e a área 2 representa o menu. A área 3 mostra três tipos de informações: uma aba de *console* informando o estado do programa, uma aba de problemas e uma área de histórico. A área 4 comporta as diferentes visões.

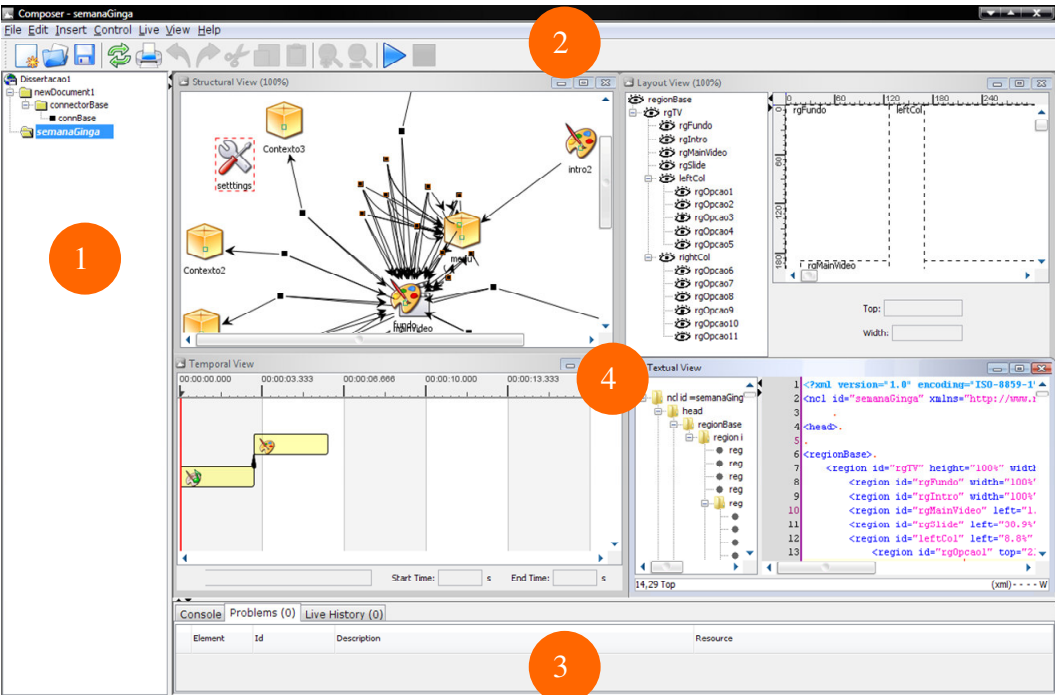


Figura 70: Tela inicial do Composer 2.2.1

Da mesma forma, a versão reprojetada do Composer permite a visualização das visões lado-a-lado (Figura 71).

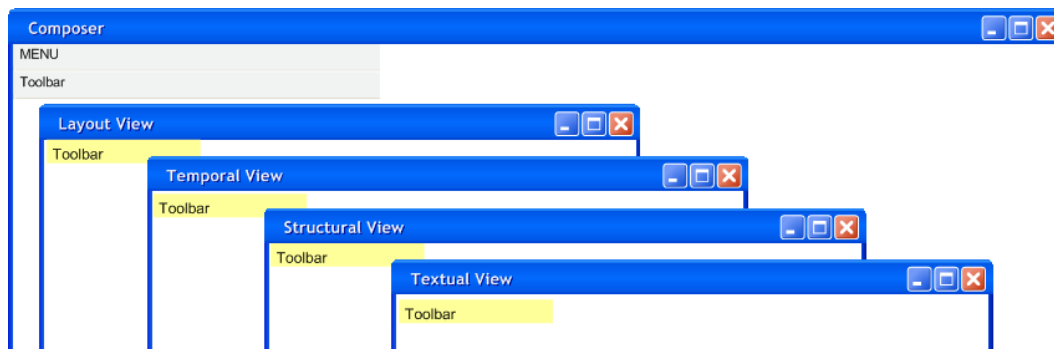


Figura 71: Composer (reprojeto) – Disposição das visões

Como extensão neste contexto, uma nova funcionalidade pode ser incorporada: o usuário poderá salvar a configuração do *workspace*, podendo escolher entre diferentes configurações. Esta funcionalidade tornou-se necessária a partir da inclusão de informações relevantes de cada visão, como mostra o painel 1 da Figura 73. Desta forma, podemos dar destaque a informações diferentes de acordo com a visão. Por exemplo, na visão de leiaute, estamos interessados em visualizar as regiões. Dessa forma, destacamos as informações relacionadas às regiões.

Para refletir sobre esta extensão, nos guiamos no fator tarefas vs. domínio, quando a tarefa se beneficia de um subconjunto específico de operações, ou seja, a organização do *workspace* remete ao usuário a possibilidade de configurar, em cada visão, apenas as operações e informações relevantes ou que merecem mais destaque para uma dada visão (Figura 72).

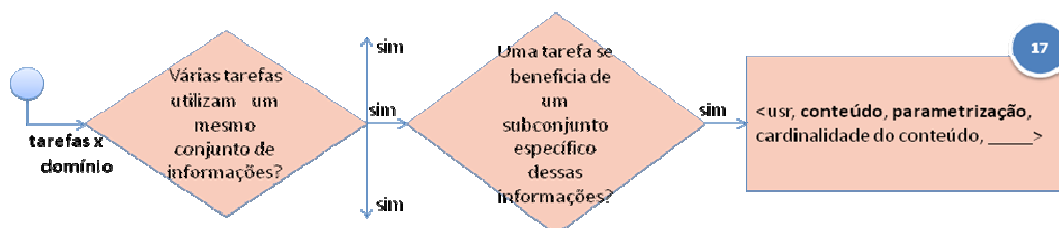


Figura 72: Fluxograma para organização do workspace

Isto pode ser realizado através da inclusão de um item de menu que permita que o usuário salve a configuração de *workspace* atual e forneça um nome adequado, de forma que possa: a) configurar posteriormente qual o *workspace* default o sistema deve apresentar ao iniciar, ou se ele será relacionado ao projeto ou documento em questão; e b) permitir que o usuário troque, em tempo real, o *workspace*, criando uma associação de tempo momentânea, ou seja, volátil. Esta

decisão poderá até quando a adaptação fica em vigor, fica a cargo de uma decisão futura de design.

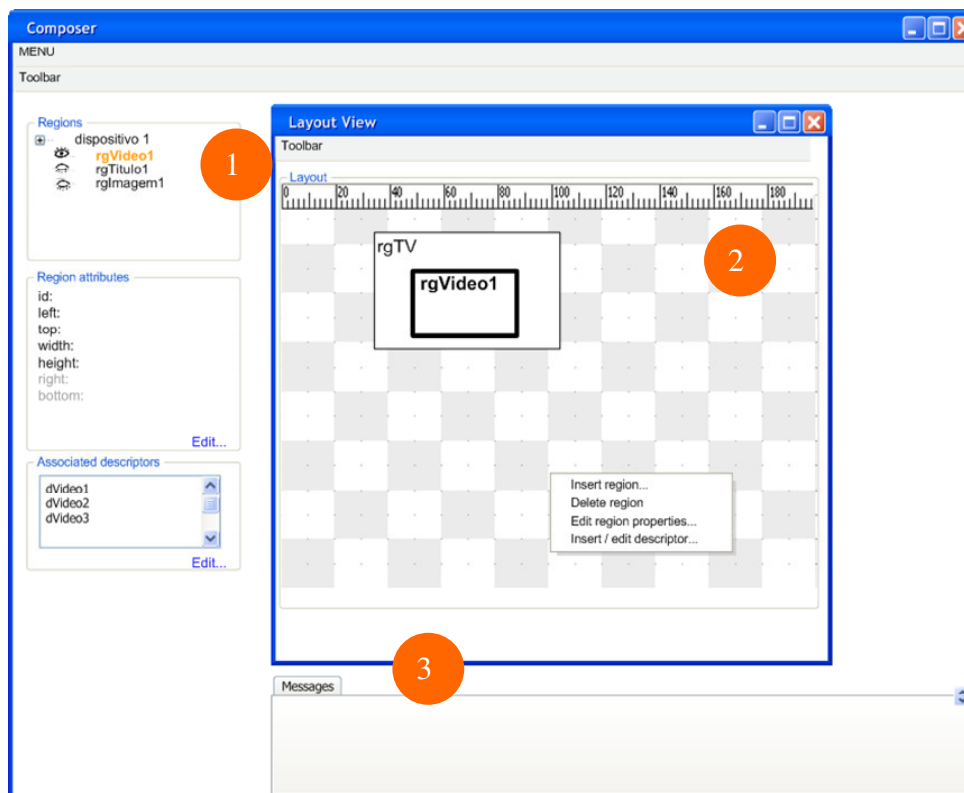


Figura 73: Composer (reprojeto) – Visão de leiaute

Assim como na versão anterior, a área 2 da Figura 73 mostra a disposição das regiões na tela, cujas posições e dimensões podem ser alteradas via manipulação direta. A diferença aqui é que uma região que estiver fora dos limites da sua região-pai será apresentada de forma diferente, pois mídias associadas a ela não ficariam visíveis. A Figura 74 mostra um exemplo deste destaque. A região *rgImage1* foi arrastada para fora da região *rgVideo1*.

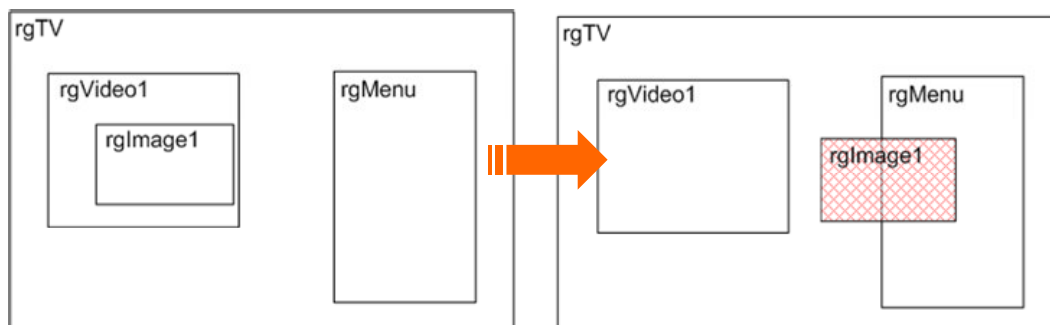


Figura 74: Composer (reprojeto) – Região-filha fora da área da região-pai

As principais informações a serem mostradas na visão de leiaute são as regiões, os atributos das regiões (para agilizar e facilitar a alteração pelo usuário) e os descritores associados a cada região. A tabela a seguir mostra essas informações desta visão.



Os nomes das regiões podem ser alterados diretamente aqui, assim como o aninhamento delas.

Regions

Esta caixa apresenta todas as regiões do documento, divididas por dispositivos. A região em destaque representa a região selecionada na área 2 (Figura 73). Ao clicar no (+) todas as regiões filhas são mostradas. Ao selecionar uma região, as informações (atributos) são mostradas na caixa logo abaixo desta, a [*Region attributes*].



Region attributes

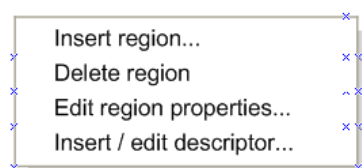
Esta caixa mostra alguns atributos da região selecionada na caixa [*Regions*]. No final da caixa há um botão [*Edit*] para acesso à janela de atributos. Os valores também podem ser alterados diretamente aqui.



Associated descriptors

Esta caixa mostra os descritores relacionados à região selecionada na caixa [*Regions*]. Para alterar os descritores, o usuário pode clicar no botão [*Edit*].

Menu pop-up (*right button*)



O menu pop-up pode ser acessado através do botão direito do mouse na área 2. Os principais comandos que aparecem neste menu são: (1) inserir região; (2) deletar região; (3) editar as propriedades

da região; e (4) inserir ou editar um descritor.

Tabela 12: Composer (reprojeto) – Principais componentes da visão de leiaute e descrição

Assim como na versão anterior, deixamos a área 3 destinada a mensagens do sistema, como erros, avisos e andamento do programa. Esta é uma área comum para todas as visões.

Duas funcionalidades que não existiam na versão anterior e merecem destaque nesta visão são (1) a criação de linhas guia para alinhamento dos objetos; e (2) a configuração do número de *pixels* da grade.

Além disto, é importante que as regiões possam ser posicionadas de forma automática, por exemplo, a Figura 75 mostra a janela para alinhamento de objetos. Os objetos podem ser alinhados verticalmente e/ou horizontalmente. Ao seleccionar uma das opções o usuário pode ainda escolher criar uma linha guia e colar os objetos a ela.

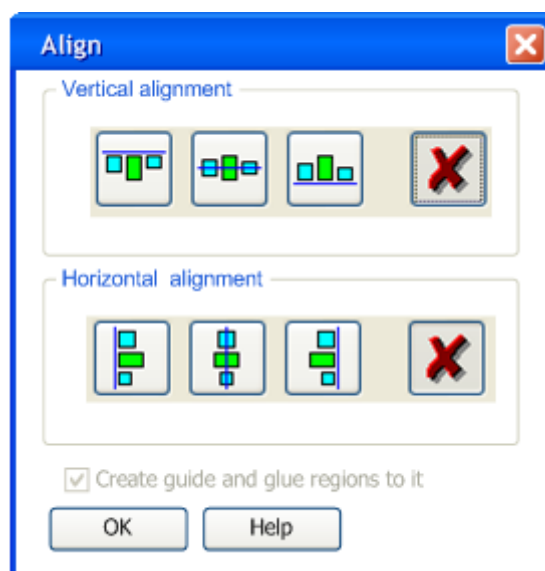


Figura 75: Composer (reprojeto) – Alinhamento de objetos

Seguindo o formato da visão anterior, a visão estrutural possui três áreas distintas (Figura 76). A área 1 é reservada para as informações referentes ao elemento seleccionado na área 2, que por sua vez mostra a estrutura do documento, de acordo com os nós e elos. A área 3 é reservada para as mensagens do sistema.

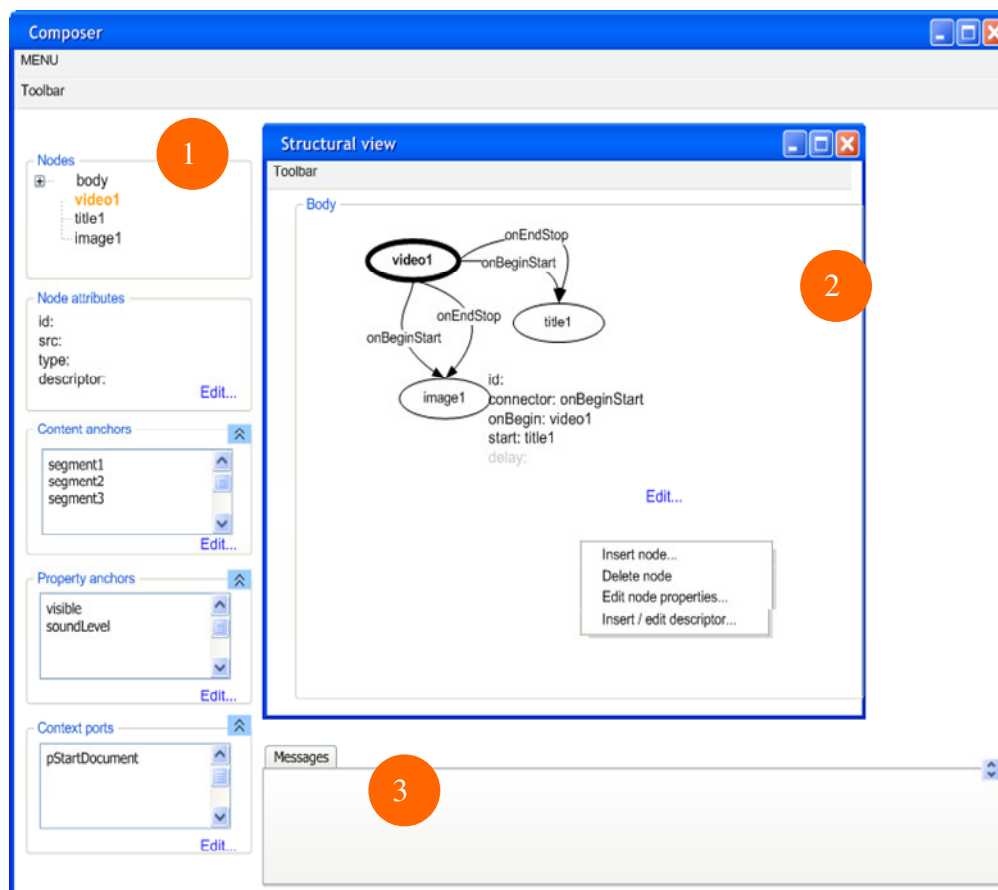


Figura 76: Composer (reprojeto) – Visão estrutural

As principais informações que esta visão se encarrega de mostrar são os nós de mídia e os elos. Desta forma, caso o usuário selecione um nó de mídia, as informações no painel 1 serão referentes ao nó. Caso o usuário selecione um elo, as informações do painel 1 serão referentes ao elo (Figura 77). Tanto na árvore quando no painel, os nós que participam do elo selecionado recebem destaque visual.

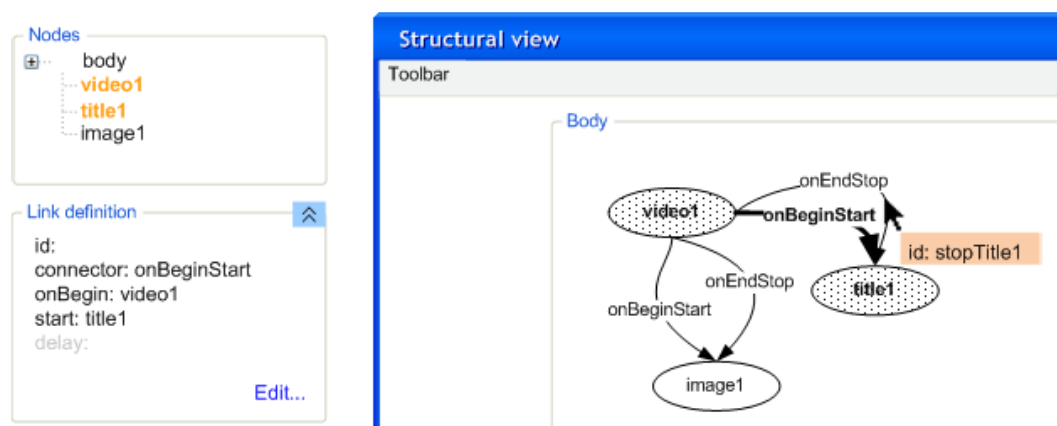
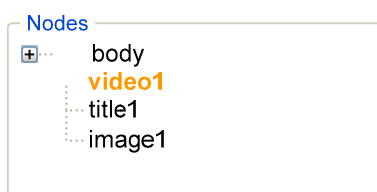


Figura 77: Composer (reprojeto) – Visão estrutural com link selecionado

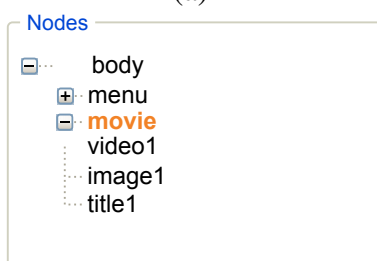
A diferença principal que podemos ver no painel 2 com relação a versão anterior é que a representação do elo mostra o conector usado e não o *id* do elo. Caso o usuário ative a opção para mostrar *tooltips*, o *id* do elo pode ser mostrado ao passar o mouse em cima dele.

Ao selecionar um nó, as seguintes informações aparecem no painel 1.

Nodes



(a)



(b)

Esta caixa apresenta todas os nós do documento. O nó de mídia em destaque representa o nó selecionado na área 2 (Figura 76).

- (a) Ao selecionar um nó, as informações (atributos) são mostradas na caixa logo abaixo desta, a [Node attributes]. Os nomes dos nós podem ser alterados diretamente aqui.
- (b) Ao selecionar um nó de contexto, a árvore mostra todos os outros nós que fazem parte do contexto selecionado.

Node attributes

Esta caixa mostra alguns atributos do nó selecionado na caixa [Nodes]. No final da caixa há um botão [Edit] para acesso a janela de atributos. Os valores também podem ser alterados diretamente aqui.

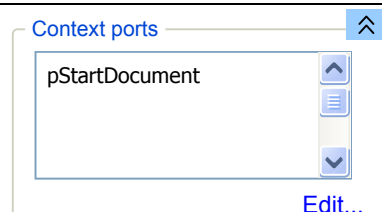
Content anchors

Esta caixa mostra a lista de âncoras de conteúdo relacionada ao nó selecionado. Para alterar as âncoras, o usuário pode clicar no botão [Edit]. Como nem sempre um nó possui uma âncora de conteúdo, esta caixa pode ser minimizada para aumentar o espaço visual das outras caixas.

Property anchors



Esta caixa mostra as propriedades da âncora selecionada na lista de âncoras de conteúdo da caixa [Content anchors]. Da mesma forma que as caixas anteriores, o usuário pode modificar as propriedades da âncora diretamente na caixa, ou acessando o botão [Edit].



Context ports

Esta caixa mostra as portas de contexto associadas ao nó selecionado. É possível alterar as portas selecionando uma da lista e clicando no botão [Edit].

Menu pop-up (right button)

O menu pop-up pode ser acessado através do botão direito do mouse na área 2.

> Clique sobre *canvas*:

(1) inserir nó; (2) inserir elo; (3) inserir um descritor; (4) inserir nó de contexto.

> Clique sobre um nó de mídia:

(1) inserir nó; (2) deletar nó; (3) editar as propriedades do nó; e (4) inserir ou editar um descritor; (5) definir como nó de início.

> Clique sobre um nó de contexto:

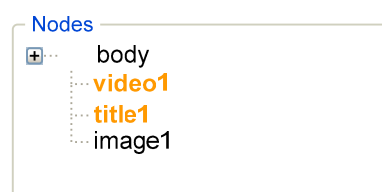
(1) expandir / colapsar; (2) inserir nó; (3) inserir um descritor.

> Clique sobre um elo:

(1) editar as propriedades do elo

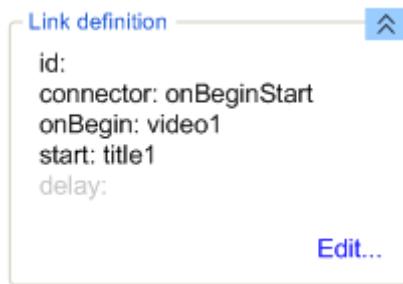
Tabela 13: Composer (reprojeto) – Principais componentes da visão estrutural e descrição (nó selecionado)

Caso um link seja selecionado, aparecerá a caixa de definição do elo no painel 1.



Nodes

Os nós que participam do elo selecionado recebem destaque visual.



Link definition

Esta caixa mostra a definição do elo, com destaque para o id, conector e papéis. Da mesma forma que nas outras caixas, os valores podem ser alterados *in place* ou através do botão [Edit].

Tabela 14: Composer (reprojeto) – Principais componentes da visão estrutural e descrição (link selecionado)

Seguindo a estrutura das visões anteriores, a visão temporal possui três áreas distintas (Figura 78). A área 1 é reservada para as informações referentes ao elemento selecionado na área 2, que por sua vez mostra a estrutura do documento, de acordo com os elos de sincronismo. A área 3 é reservada para as mensagens do sistema.

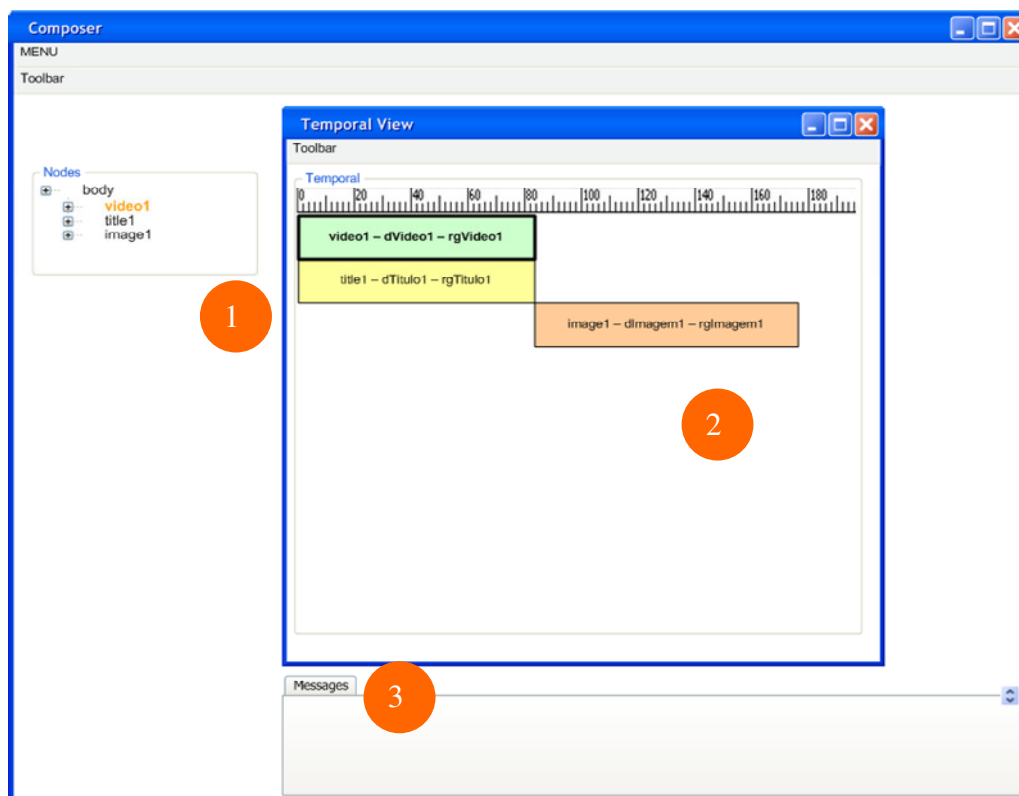


Figura 78: Composer (reprojeto) – Visão temporal

As principais informações que esta visão se encarrega de mostrar são as mudanças de comportamento dos nós e os pontos de interação. Desta forma, caso o usuário clique em um ponto da linha do tempo, as informações no painel 1 serão referentes ao ponto selecionado.

Seguindo a estrutura das visões anteriores, a visão textual possui três áreas distintas (Figura 79). A área 1 é reservada para as informações da estrutura do documento NCL. A área 2 se refere ao código NCL e a área 3 é reservada para as mensagens do sistema. Ao clicar em um elemento do painel 1, a linha referente ao elemento é destacada no código. Esta funcionalidade já existe na versão anterior e foi mantida.

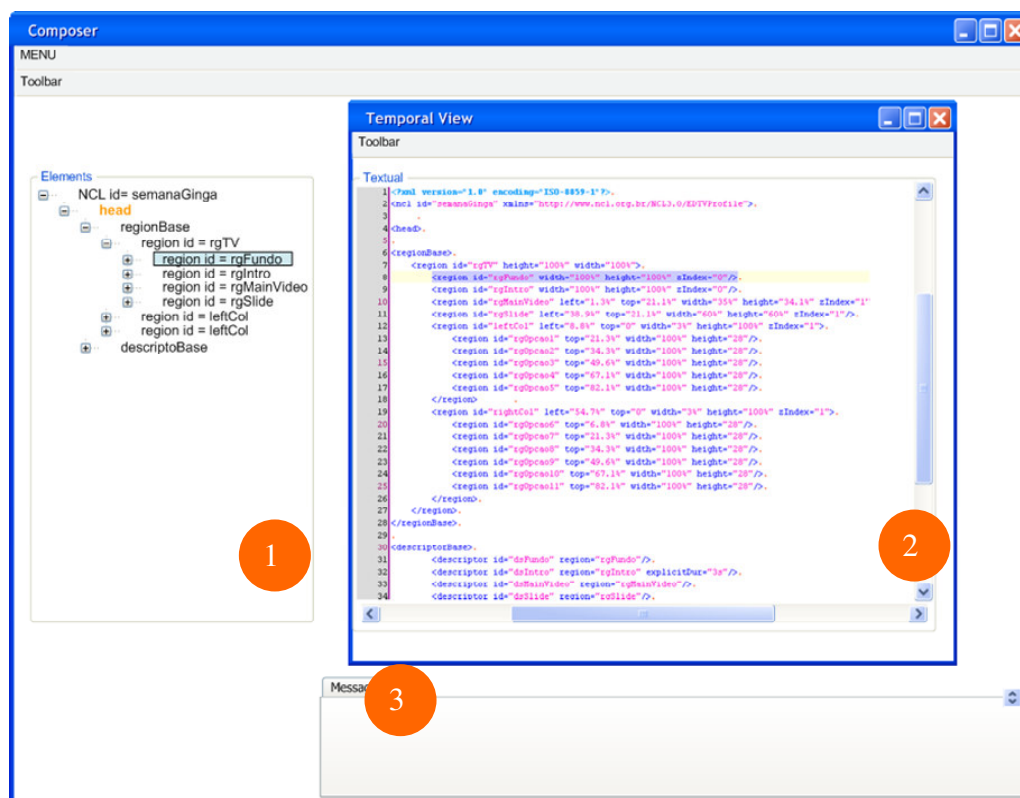


Figura 79: Composer (reprojeto) – Visão textual

Nesta visão podemos ter dois tipos de adaptação: a adaptação para a diferenciação do conteúdo e a adaptação do conteúdo de ajuda. A primeira se refere à capacidade do sistema de diferenciar os conteúdos, colorindo o código de acordo com o tipo de arquivo: arquivo NCL completo, arquivo de base de conectores, arquivo do tipo *wizard* entre outros.

A segunda é uma adaptação do conteúdo dependente do contexto, ou seja, fornecer um *auto-complete* do código para auxiliar os usuários novatos ou agilizar o trabalho dos experientes. Os usuários novatos podem ser tornar especialistas na medida em que utilizam as dicas oferecidas pelo *auto-complete*.

Os fluxogramas a seguir ilustram as perguntas e respostas que foram feitas para a escolha das adaptações descritas anteriormente.

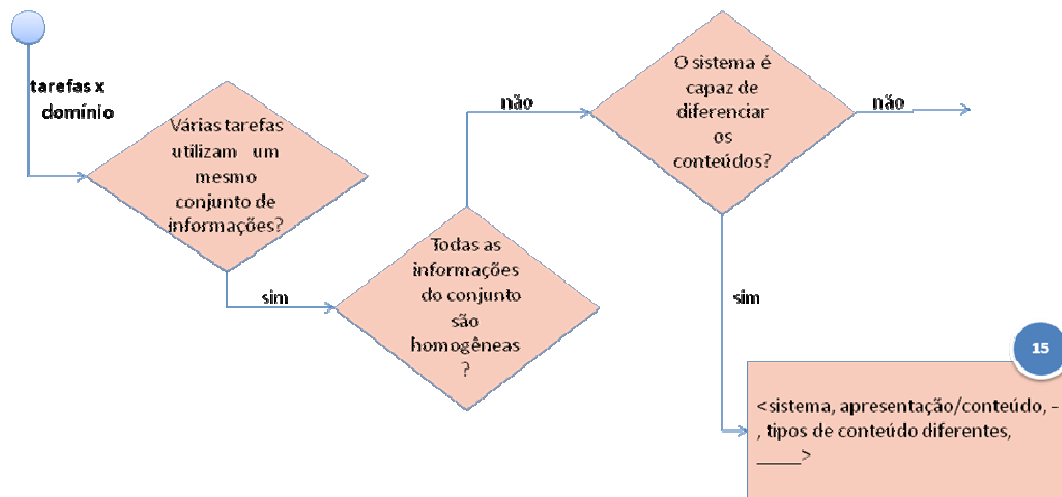


Figura 80: Fluxograma para a diferenciação do conteúdo da visão textual

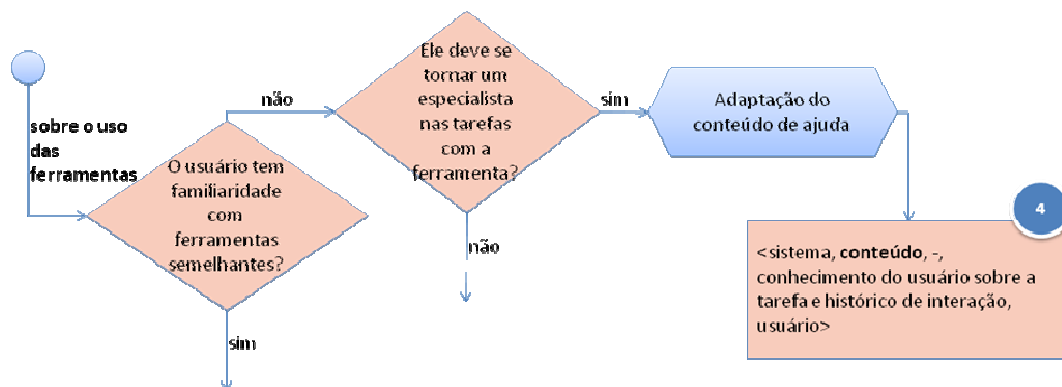


Figura 81: Fluxograma para a adaptação do conteúdo de ajuda da visão textual

Projetos e documentos NCL

Inserimos na janela de criação de projetos a opção para criação de uma pasta de mídias dentro do projeto. Caso o usuário não queria criar uma pasta, ele poderá selecionar uma pasta que contenha as mídias a serem usadas.

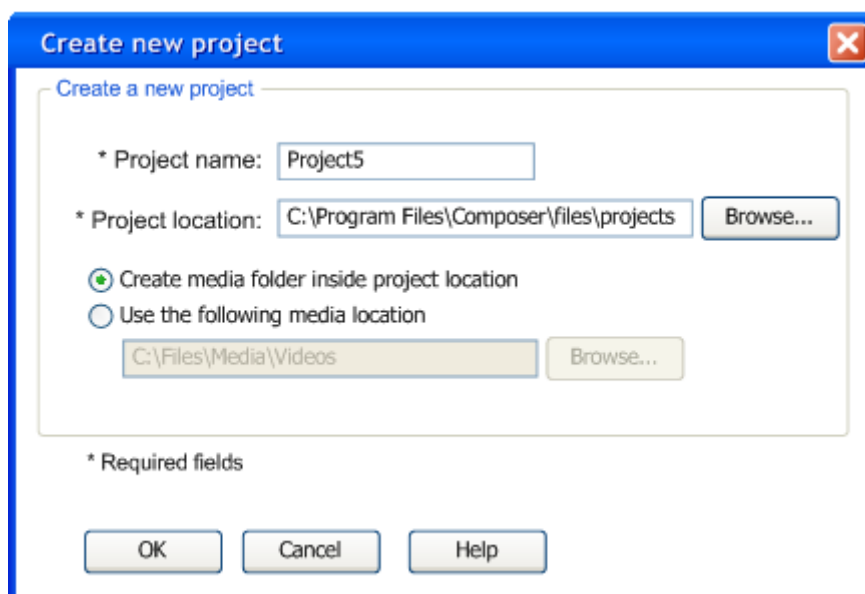


Figura 82: Composer (reprojeto) – Criar um novo projeto

Aqui torna-se possível incluir um mecanismo de parametrização, configurando o comportamento do sistema para a escolha de uma pasta *default* para a localização do projeto e nome da pasta de mídias. Isto pode ser adicionado a uma opção de item de menu *Options* dentro do menu *Tools*, por exemplo. A janela de opções conterá várias informações para configuração segundo a preferência do usuário. Destacaremos algumas informações importantes no decorrer da descrição deste reprojeto.

Esta adaptação surgiu da resposta que uma tarefa não possui uma forma única de ser realizada e não é possível mapear cada forma para um grupo específico de usuários.

Após a criação do projeto o usuário poderá: (1) começar a criar seu novo documento NCL; (2) importar um documento NCL; ou (3) rodar um wizard para criação de um documento NCL.

A criação de um novo documento será explicada ao longo desta seção, de acordo com as possíveis etapas como criação de regiões, criação de nós de mídias, criação de elos etc.

Caso queira importar um documento NCL, o usuário deverá escolher a opções mostradas na Figura 83. Primeiro o usuário deverá localizar o arquivo NCL.

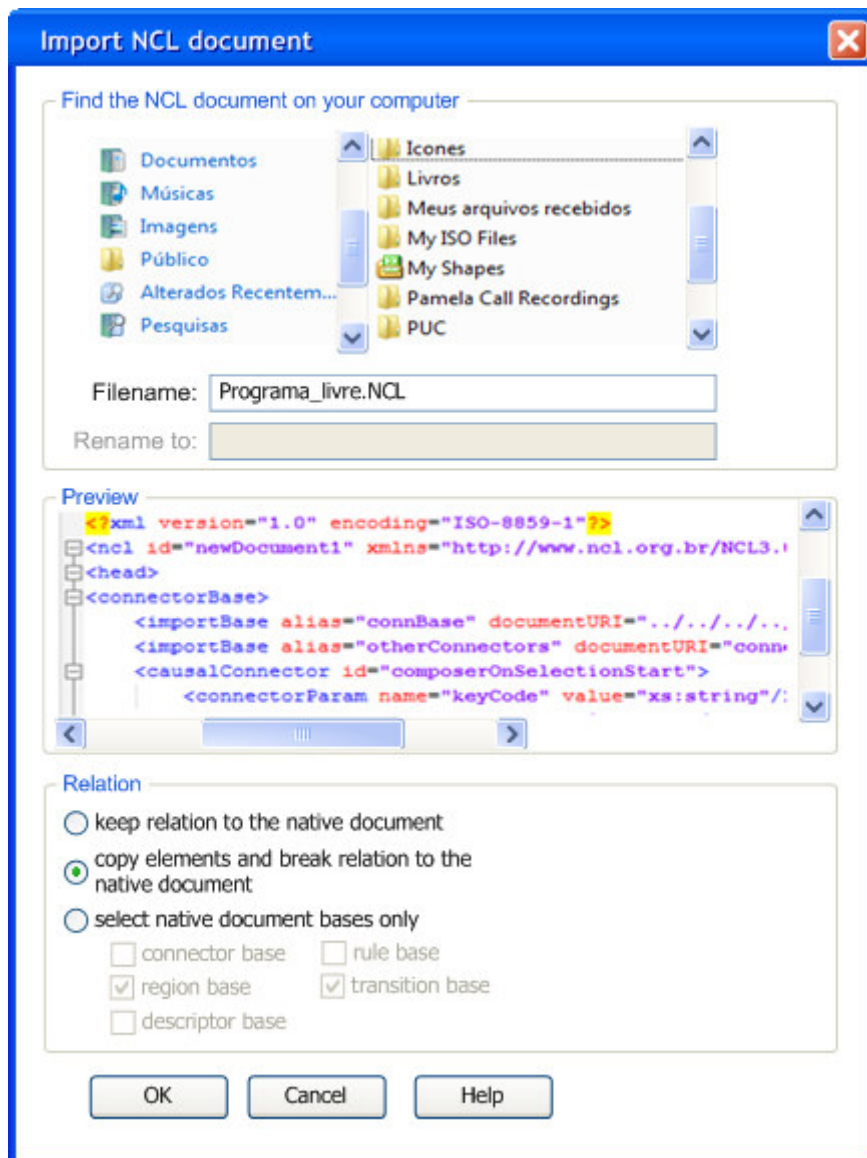


Figura 83: Composer (reprojetor) – Importar documento NCL para um projeto

Eventualmente ele poderá alterar o nome do arquivo **dentro** do projeto [*Rename to*]. Assim que posicionar o cursor no campo, ele será ativado. Abaixo da lista de diretórios, aparece o código do arquivo NCL para que o usuário se certifique que está escolhendo o arquivo correto. Depois ele deverá escolher o tipo de relação que deseja que este arquivo tenha com o documento: (a) manter o vínculo com o documento existente; (b) copiar os elementos e romper o vínculo com o documento existente (ou seja, remover o elemento *importBase*); ou (c) selecionar apenas algumas bases do documento existente, podendo escolher quaisquer bases do documento importado.

Quando o usuário escolhe a opção (a), estamos lidando com uma interação do tipo caixa preta, ou seja, pode ser que haja algum erro no documento a ser

importado. Nesse caso, a janela da Figura 84 irá aparecer e o usuário poderá escolher entre (i) abrir o projeto onde o documento está de fato para editá-lo; ou (ii) copiar o documento para o projeto atual, quebrando o vínculo com o documento original, para editá-lo.

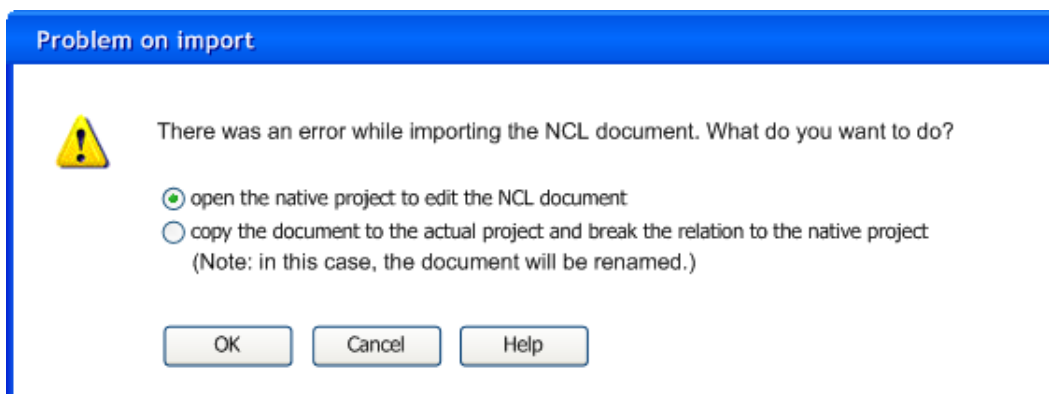


Figura 84: Composer (reprojeto) – Problemas ao importar um documento NCL

Caso o usuário escolha a opção (c), automaticamente o sistema reconhece quais são as bases existentes e ativa a opção. No exemplo da Figura 85 o documento a ser importado contém apenas as bases de regiões.

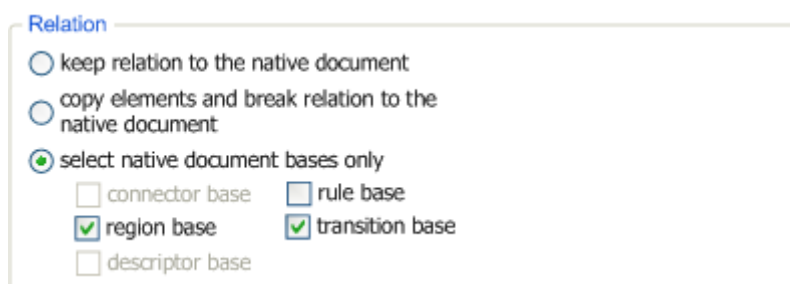


Figura 85: Composer (reprojeto) – Importar apenas as bases do documento NCL para um projeto

Pode ocorrer um erro crítico na importação de um documento NCL. Nesse caso, o usuário será avisado se o documento estiver corrompido ou se houver algum problema que não permitirá que o documento seja usado no projeto atual.

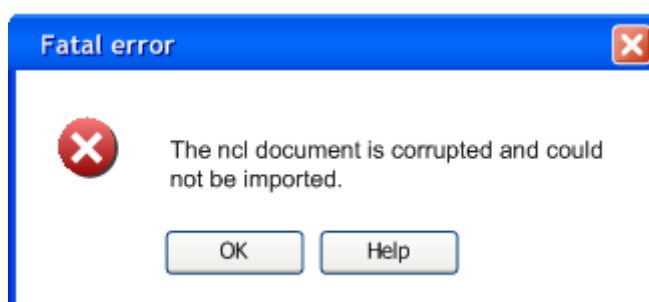


Figura 86: Composer (reprojeto) – Erro crítico ao importar documento NCL

Caso o usuário queira executar um *wizard* para criação de um documento NCL, ele deverá escolher esta opção a partir do menu. Aparecerá uma janela (Figura 87) na qual ele deverá escolher a localização do arquivo do *wizard* (.wzd – por sugestão). Será sugerido que o nome do documento NCL tenha o mesmo nome do arquivo .wzd, porém, é possível que o usuário altere. Ele poderá ver o código do arquivo selecionado através do *Preview*. Após a confirmação da escolha, o *wizard* será iniciado. Ao final do *wizard* será gerado um documento NCL para que o usuário possa testar e, eventualmente, fazer quaisquer tipos de alterações. Por isto, é importante que a pasta que contém as mídias seja informada logo no início da criação de um projeto.

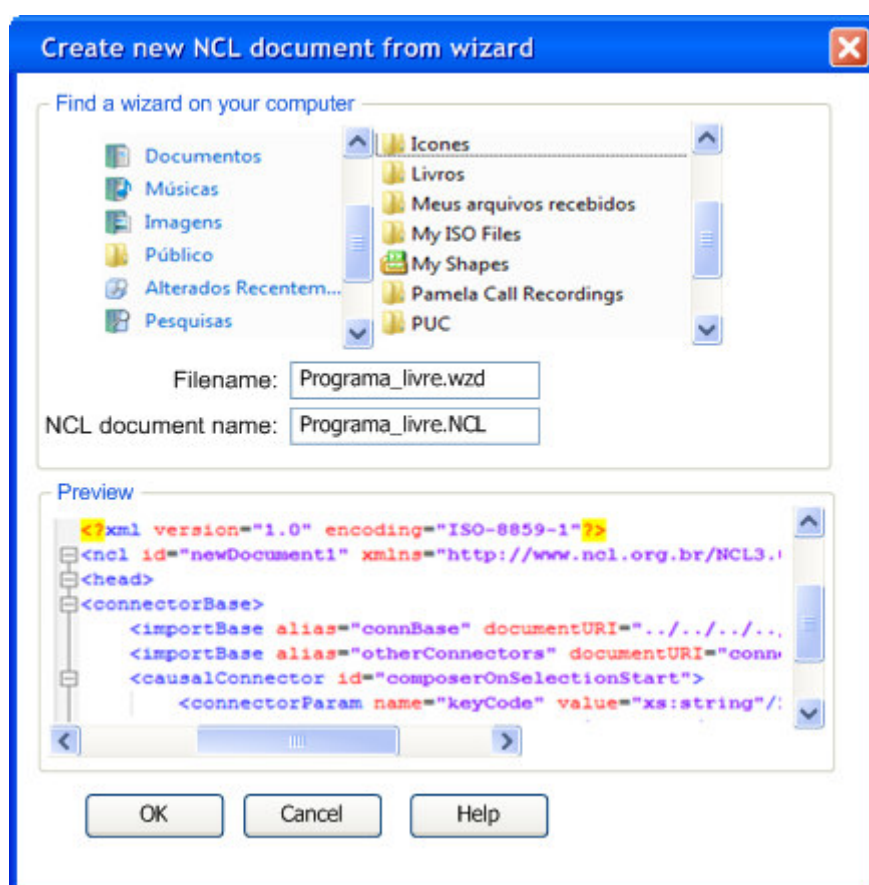


Figura 87: Composer (reprojeto) – Criar novo documento NCL a partir de um *wizard*

No Composer, um *wizard* pode ser visto como uma integração externa, do tipo componente. Não estamos, de fato, adicionando uma nova funcionalidade a cada *wizard* diferente que é executado, mas estamos incorporando no sistema um novo objetivo: o de automatizar uma tarefa. A utilização do *wizard* será relacionada aos interesses e necessidades dos usuários. Não podemos dizer, *a priori*, que todos os usuários precisarão de um apoio segundo um *wizard*, porém,

torna-se necessário a inclusão da funcionalidade: importar (e executar) um arquivo de *wizard* para atender os que precisarem.

Ao ativar a opção para abrir um projeto, o usuário deverá escolher a localização do projeto. Esta operação pode resultar em sucesso: onde todas as mídias e arquivos dependentes foram encontrados; ou em falhar podendo acontecer um dos dois casos descritos a seguir:

(1) Arquivos (mídias, bases etc.) não foram encontrados, sendo oferecido uma oportunidade de indicar diretórios onde eles estejam ou mesmo trocando um arquivo por outro.

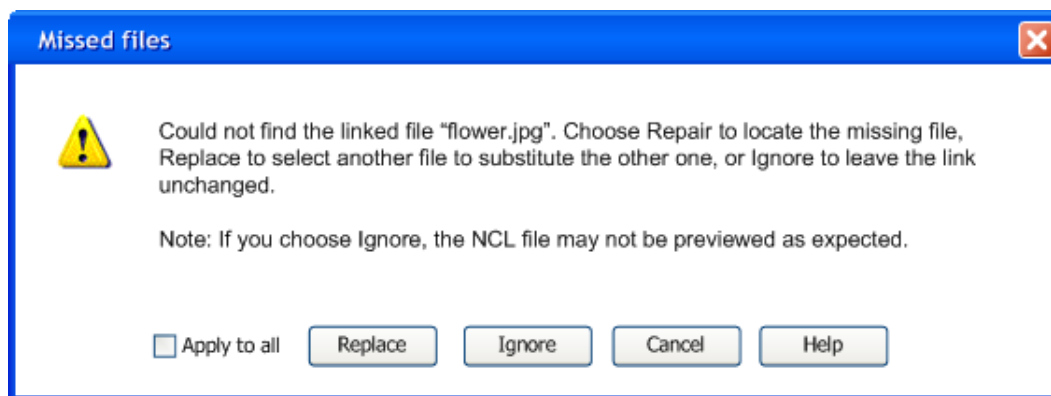


Figura 88: Composer (reprojetor) – Arquivos não encontrados ao abrir um projeto

Replace	Abre a janela para a localização do arquivo ou de um novo arquivo, substituindo o não encontrado.
Ignore	Ignorar o erro. Neste caso, o usuário será alertado novamente quando for rodar o programa.
Apply to all	Como vários arquivos podem não ter sido encontrados, o usuário pode marcar esta opção para que não seja alertado todas as vezes que um arquivo não for encontrado. (Esta opção só é válida nesta interação.)

Tabela 15: Erro ao abrir projeto – arquivos não encontrados (relação entre o botão e a ação referente)

(2) Um ou mais dos arquivos de projeto ou NCL está corrompido, sendo oferecida uma oportunidade de substituí-los.

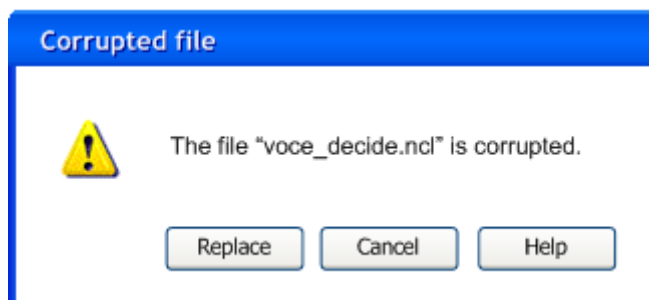


Figura 89: Composer (reprojetor) – Arquivos corrompidos ao abrir um projeto

Gerenciamento de arquivos de mídia

Os arquivos de mídia podem ser visualizados e gerenciados através de uma biblioteca de elementos de mídia, na qual as mídias podem ser identificadas por diversos *tags* e visualizadas a partir de diferentes hierarquias (por tipo de mídia - Figura 90 -, por *tag* - Figura 91 – ou por mídias de acordo com o projeto - Figura 92).

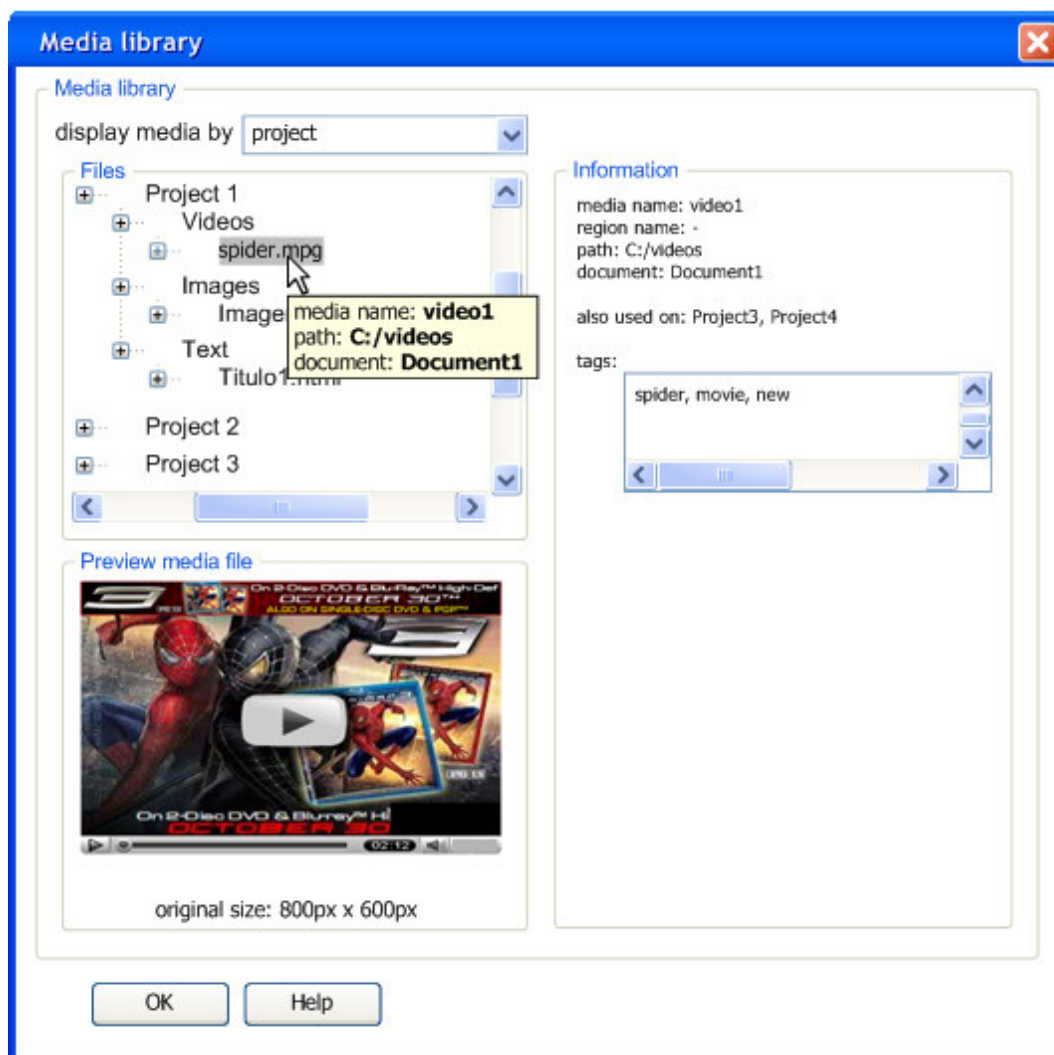


Figura 90: Composer (reprojeto) – Biblioteca de mídias (mostradas por projeto)

Para qualquer uma das três opções de ordenação das mídias, sempre será apresentado ao usuário um *preview* do arquivo (no caso de um vídeo ou arquivo de áudio, ele poderá ser tocado a partir desta janela) e as informações como o *id* do nó que está relacionado a mídia e a região, além do nome do documento a qual a mídia pertence e os projetos onde ela é usada.

A árvore de mídias é apresentada de acordo com os projetos. Depois cada mídia é separada pelo tipo (vídeo, imagem etc.). Por manipulação direta, é

possível que o usuário copie uma mídia de um projeto e cole em outro ou arraste uma mídia de um projeto para outro. Ao selecionar uma mídia da árvore, as informações são alteradas no painel [*Information*]. Ao passar o mouse sobre o nome de uma mídia, o *id*, a localização e o nome do documento relacionado são mostrados em uma *tooltip*.

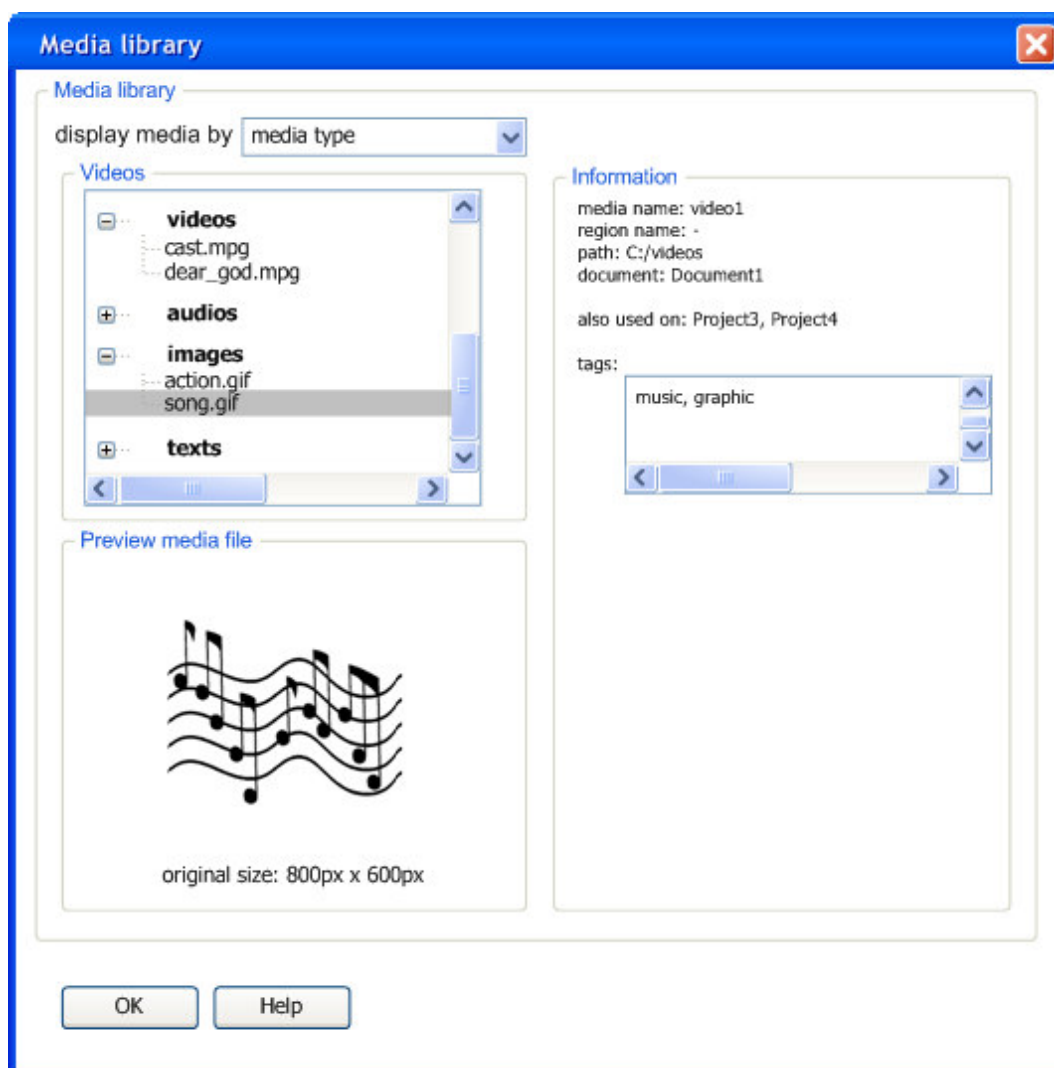


Figura 91: Composer (reprojeto) – Biblioteca de mídias (mostradas por tipo)

Quando o usuário escolhe a opção de visualização por tipo de mídia, a árvore se organiza de acordo com os possíveis tipos de mídia, e abaixo de cada tipo encontra-se a lista de mídias.

Ao escolher a visualização por *tags* são mostradas as *tags* em ordem alfabética. Abaixo de cada *tag* encontra-se a lista de mídias que foram categorizadas com a *tag* em questão. A caixa que lista as *tags* permite a alteração do nome, quando o usuário clica duas vezes sobre a ela ou solicita a opção [*Rename*] pelo botão direito do mouse.

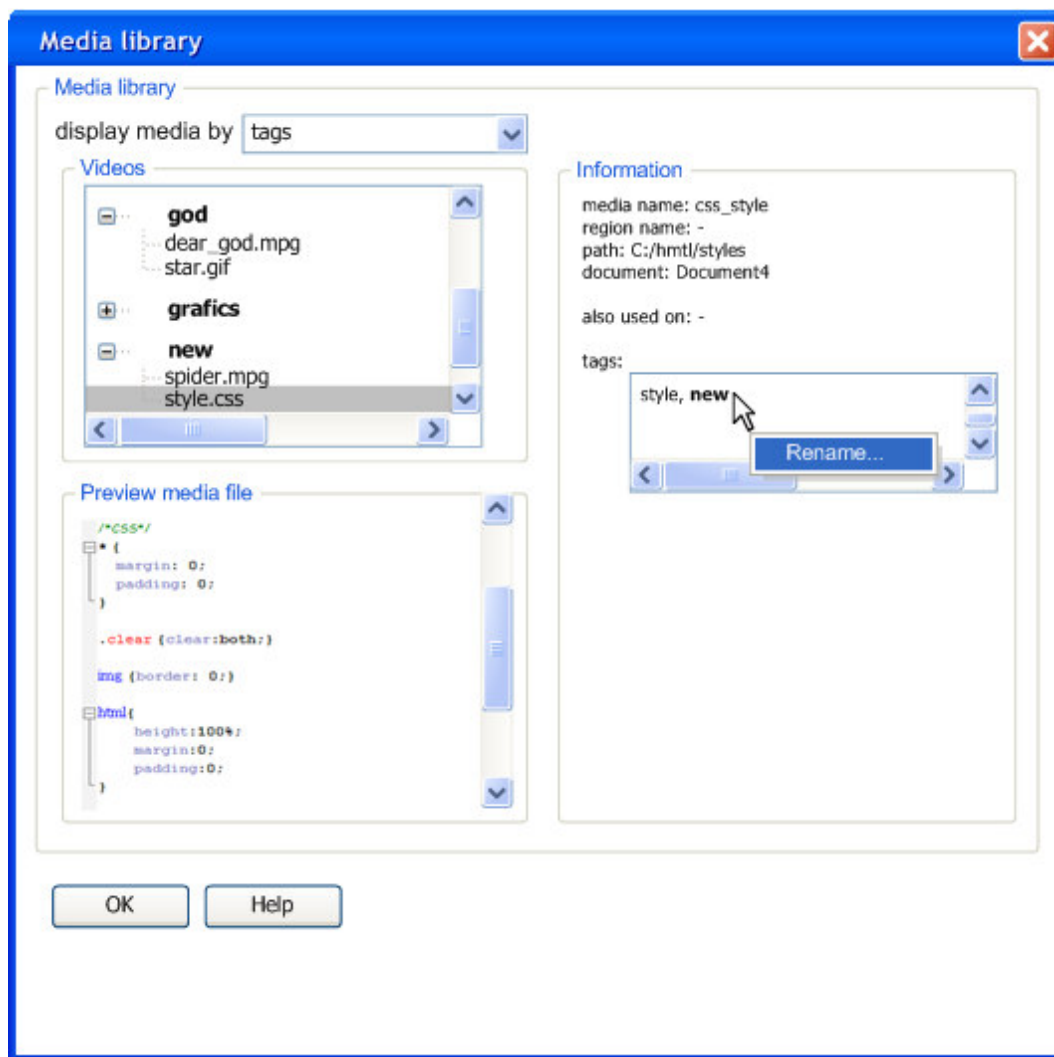


Figura 92: Composer (reprojeto) – Biblioteca de mídias (mostradas por *tags*)

Manipulação de regiões

A janela de criação de uma região sofreu algumas modificações importantes. Primeiro, ela recebeu imagens ilustrativas para cada propriedade que necessita que o usuário tenha domínio sobre a linguagem NCL. Além disto, qualquer modificação pode ser assistida por um *preview*, que é desfeito caso o usuário saia da janela de criação de uma região através do botão [*cancel*].

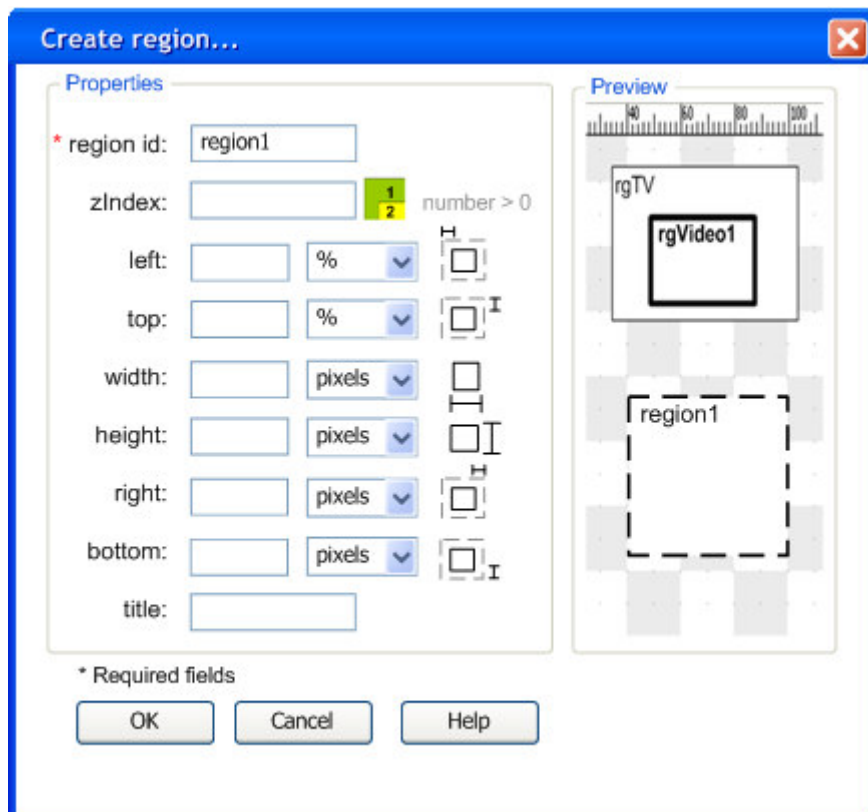


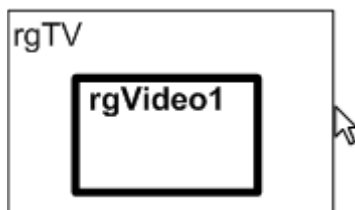
Figura 93: Composer (reprojeto) – Criar região

A criação da região através do menu direito do mouse cria uma região com tamanho *default* e *id* automático. Dessa forma, o usuário pode criar todas as regiões rapidamente para só depois alterar suas propriedades. Ele pode fazer isto clicando duas vezes sobre o desenho de uma região ou através da opção [*Edit region properties...*] do menu *pop-up* (botão direito do mouse) ou do menu principal. A Tabela 16 exemplifica algumas interações e o efeito ou passos referentes.

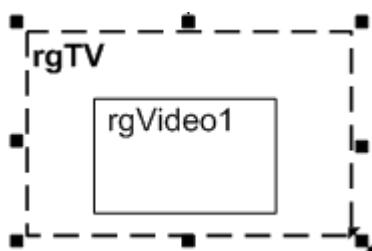
Escalar uma região

É possível redimensionar uma região arrastando suas linhas limítrofes. A seguir exemplificamos:

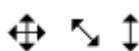
- (1) Quando o usuário clica em uma região.



(1)



(2)

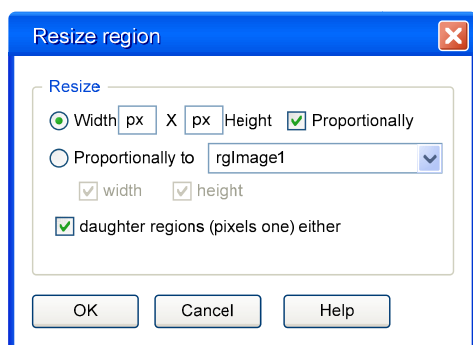


(3)

- (2) Ela se destaca e recebe linha tracejada. Em pontos determinados do contorno aparecem referências para redimensionar o tamanho da região.

- (3) Exemplos dos possíveis cursores: mover; redimensionar pelos canto direito; e redimensionar pelos pontos centrais horizontais.

Outra forma de redimensionar uma região é através da opção de menu (4).



(4)

O usuário pode escolher: (a) proporcionalmente ou não, com base em dimensões previamente definidas; (b) com base em outra região e determinar se deseja copiar a largura e/ou a altura; e (c) para ambas as opções ele pode escolher que as regiões filhas que estiverem definidas em pixels também redimensionadas proporcionalmente.



(1)



(2)



(3)

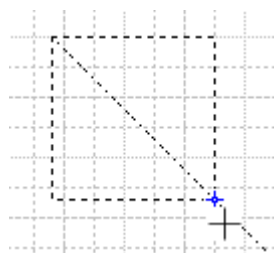
Alterar o aninhamento de uma região

É possível alterar o aninhamento de uma região movendo-a para outra região pai ou mesma para a raiz da base de regiões na árvore de manipulação direta. A figura ao lado exemplifica:

- (1) O usuário clica em uma região e ela se destaca.
- (2) Ele arrasta a região para a posição entre a região *rgVideo1* e *rgTitulo1*. Uma linha aparece para guiar o usuário com relação a posição de inserção.
- (3) A região *rgImagem1* reposicionada.



(1)



(2)

Desenhar uma região

Através da ferramenta *drawing*, ativa na visão de leiaute, o usuário poderá abrir a janela de desenho de regiões (1), que ficará sobre o palco das regiões. Clicando no botão [*create region*] o usuário poderá criar uma região clicando e arrastando o mouse sobre outra região ou sobre a região base (2).

A linha diagonal representa que a região tem altura e larguras iguais e, só aparece, neste caso.

Para apagar, basta selecionar a borracha [*delete region*] e clicar sobre a linha de uma região.

Tabela 16: Composer (reprojeto) – Relação entre interações e o efeito ou passos referentes a manipulação de regiões

Nesta versão, o usuário pode também copiar uma região e colar a região copiada para o palco. A nova região terá o *id* automaticamente criado e terá os

mesmos atributos da região cópia, podendo ser alterados acionando a referente opção.

Como foi mostrado na descrição da visão de leiaute, o usuário pode inspecionar os atributos de uma região, selecionado uma região do palco e inspecionando os atributos no painel região. Da mesma forma ele pode inspecionar os descritores relacionados a região, assim como os nós.

Manipular descritores

Na visão de leiaute, é possível visualizar a lista de descritores relacionada a uma região. A Figura 94 mostra a tela de criação/edição de um descritor.

The screenshot shows the 'Create descriptor...' dialog box. It has a blue title bar with a close button. The main area is divided into four panes: 'Existing descriptors' (top left) with a list of descriptors and 'new'/'delete' buttons; 'Properties' (top right) for the selected descriptor, showing 'descriptor id', 'region', and 'nodes'; 'Parameters' (bottom left) with a list of parameters and 'new'/'delete' buttons; and another 'Properties' (bottom right) for the selected parameter, showing 'param' and 'value'. A 'Code preview' section at the bottom shows the XML representation of the descriptor and parameter. At the very bottom, there are 'OK', 'Cancel', and 'Help' buttons, and a yellow warning icon with the text: 'No region is associated to this descriptor, although there is at least one visual media using it.'

Figura 94: Composer (reprojeto) – Criação de descritores (modo compacto)

Nesta mesma tela o usuário pode criar um descritor novo, clicando na ação [new] e editar os descritores existentes. A área de propriedades do descritor é sempre a mesma. O usuário deve obrigatoriamente informar um *id* para o descritor e deve selecionar uma região e um ou mais nós relacionados. No exemplo, o

usuário não definiu uma região para o descritor, apesar de ter definido o nó de mídia. Este é um exemplo que não gera um erro, pois o próprio usuário pode ter decidido por não visualizar o vídeo, mas apenas tocar o seu áudio. Nesse caso, um *warnig* irá aparecer no rodapé da janela, bem próximo aos botões de finalização dela. O [x] ao lado do descritor *dmenu* da lista de descritores indica que este descritor não está sendo usado, e por isto pode ser removido.

A caixa *Parameters* é composta de uma área de adição/edição/remoção das propriedades dos parâmetros e uma lista de parâmetros. Ao selecionar um parâmetro da lista, automaticamente os valores passam para o formulário e o rótulo do botão de ação muda para [*Edit*]. O usuário pode remover um ou mais parâmetros através da ação [*delete*]. Caso queira criar um novo, basta clicar na ação [*new*] e preencher o formulário de propriedades. O botão terá rótulo [*Add*].

Cada tipo de parâmetro terá propriedades diferentes. A Figura 94 mostra propriedades para o parâmetro *soundLevel*. A Figura 95 mostra a definição de um parâmetro do tipo *fit*. Novamente optamos por usar imagens que demonstram o comportamento da opção a ser selecionada pelo usuário. Observe que, na lista de parâmetros, deixamos claro qual o tipo do parâmetro e o valor, para servir como referência ao usuário. Exemplo: parâmetro do tipo: *soundLevel* / valor: 0.9.

A caixa [*Preview*] mostra o pedaço do código NCL referente às alterações que estão sendo feitas.

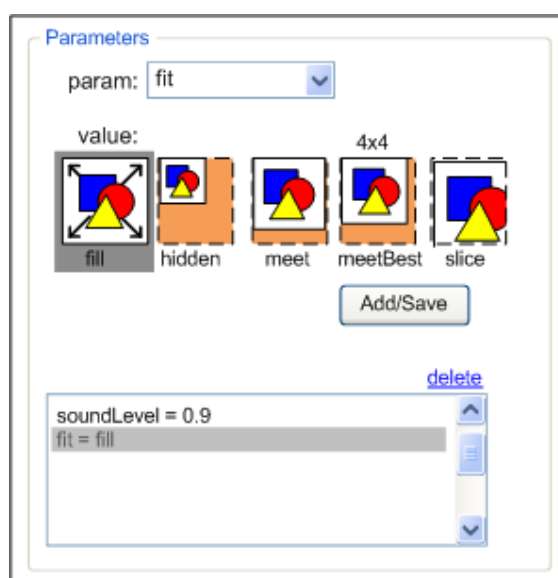


Figura 95: Composer (reprojeto) – Criação de descritores (parâmetro fit)

Manipular nós de mídia

A criação de um nó de mídia poderá ser feita através do menu ou através do menu *pop-up* que aparece ao usar o botão direito do mouse na visão estrutural. Essas ações resultam na criação de um nó como *placeholder*, ou seja, será fornecido um *id* automático e será do tipo *image* como *default*. Esta funcionalidade foi inserida para agilizar o processo de criação.

Na janela de propriedades do Composer, o usuário poderá escolher um arquivo de imagem, um de vídeo, um de áudio e um de texto como *default* para quando um nó não tenha sido completamente especificado (caso do *placeholder*) o usuário possa testar seu programa sem problemas. Estas configurações também podem entrar na janela de configurações de opções do sistema.

Sempre após a criação de um nó, o usuário pode alterar suas configurações clicando duas vezes sobre o desenho do nó, clicando com o botão direito do mouse e selecionando a referente opção no menu *pop-up*, ou através das caixas suspensas da visão estrutural.

Além desta forma, o usuário pode selecionar um nó, acionar o comando [*copy*] e depois o comando [*paste*] no palco. Um novo nó será criado com as mesmas propriedades do nó copiado, com exceção do *id* que será sugerido outro.

Outra forma alternativa de criação é arrastar um arquivo (de imagem, de vídeo, de áudio ou texto) para o palco. Da mesma forma é criado um *placeholder* para o nó, porém, o sistema poderá recuperar o tipo de mídia e já associar isto à sua configuração. Além disto, como sugestão de *id*, será fornecido o nome do arquivo. A Figura 96 mostra esta interação.

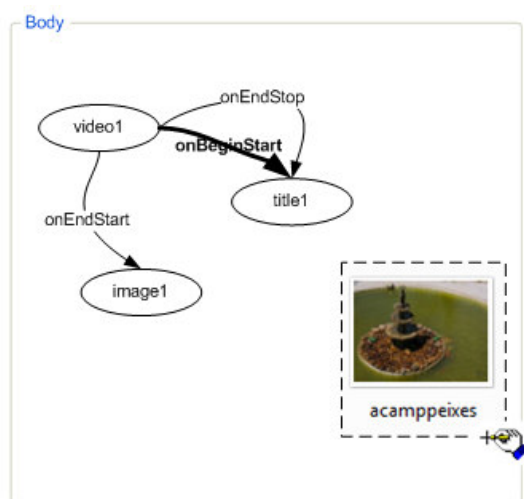


Figura 96: Composer (reprojeto) – Criação de um nó de mídia através de manipulação direta (arquivo sobre espaço vazio)

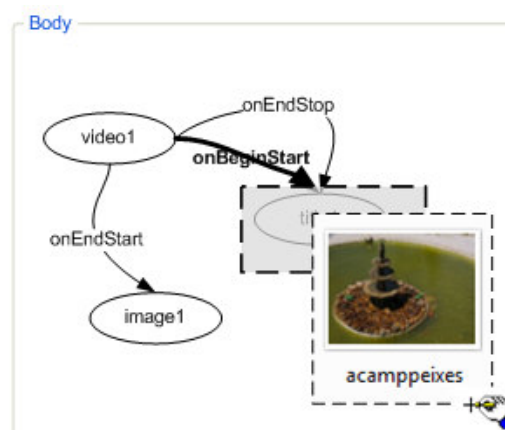


Figura 97: Composer (reprojeto) – Criação de um nó de mídia através de manipulação direta (arquivo sobre nó)

Caso o nó seja arrastado para um espaço vazio, um novo nó será criado de acordo com a explicação anterior. Caso o arquivo seja arrastado para cima de um nó, o nó receberá marcação diferenciada para demonstrar que recebeu o foco (Figura 97).

Caso o nó que recebe o arquivo já possua um arquivo associado, aparecerá uma janela de informação, pedindo a confirmação para sobrescrever ou não o arquivo.

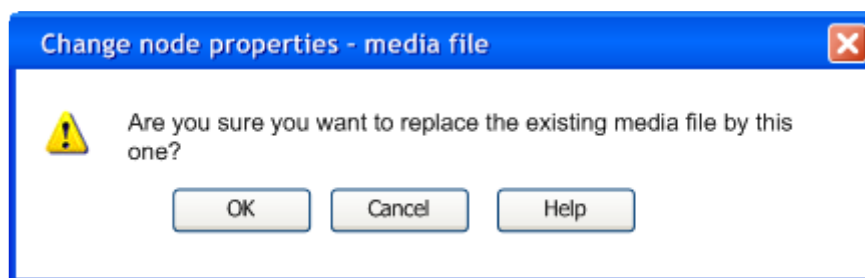


Figura 98: Composer (reprojeto) – Janela de confirmação para sobrescrever arquivo

Depois, o usuário deverá escolher entre manter o *id* original ou mudar para um de sua escolha (Figura 99). Como sugestão, é dado o nome do arquivo.



Figura 99: Composer (reprojeto) – Janela para alteração de *id*

A janela de propriedade de um nó pode ser visualizada na Figura 100. O tipo do nó é automaticamente detectado pelo sistema de acordo com o preenchimento do campo file.

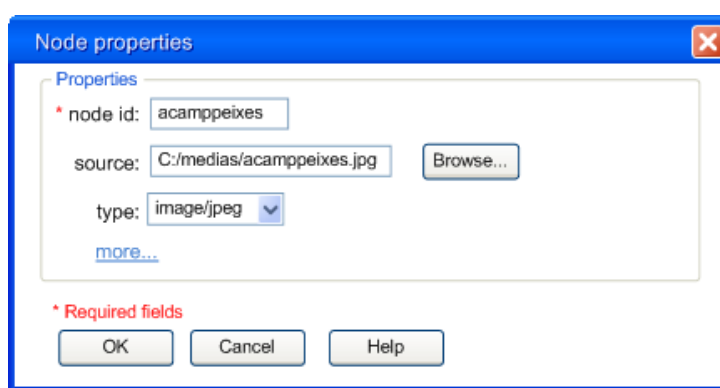
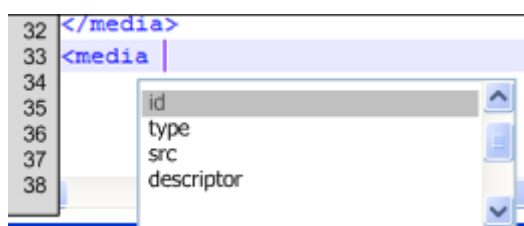


Figura 100: Composer (reprojeto) – Janela de propriedades de um nó

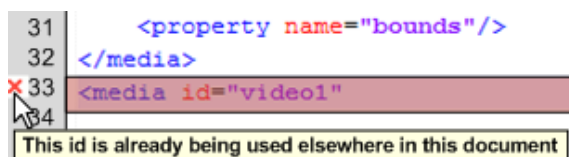
Sempre que o usuário modificar a associação de um arquivo de mídia e o próprio tipo de mídia, uma mensagem deve ser mostrada, avisando sobre possíveis inconsistências e sugerir correções, tal como modificar a duração previamente especificada para um imagem que foi substituída por um vídeo para a duração real do vídeo.

Ao modificar cada atributo, também devem haver avisos sobre possíveis inconsistências e apresentações de sugestão de correções. Esse aviso deve ser apresentado de forma sutil, de maneira que possa ser ignorado pelo usuário sem que isso requeira qualquer ação adicional. Por exemplo, ao definir uma duração para uma mídia contínua, deve ser apresentado um alerta para o usuário que a mídia será cortada ou que o último quadro ficará visível por n segundos a mais do que a duração intrínseca da mídia.

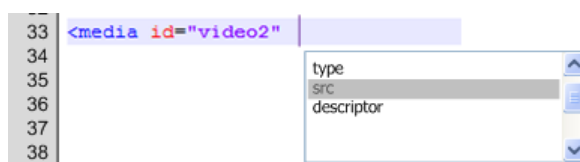
A seguir demonstramos como seria a criação de um nó de mídia na visão textual. Este processo poderá ser estendido para a criação dos outros elementos, cada um com as respectivas propriedades.



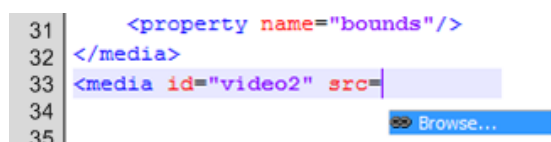
- (1) Usuário digita: “<media” + espaço.
- (2) Uma caixa de seleção irá se abrir.
- (3) O usuário escolhe a opção *id*.



- (4) O sistema cria uma marca (x) ao lado da linha que contém um erro.
- (5) Passando o mouse sobre a marca, o usuário informa o erro.



- (6) O usuário continua digitando o código. Sempre que dá um espaço, o sistema abre uma janela com as possíveis opções a serem preenchidas ainda.



- (7) Quando escolhe a opção *src* o sistema pede que o usuário escolha a localização do arquivo.
- (8) Clicando na opção [*Browse...*] uma janela com o gerenciador de arquivos irá se abrir.



- (9) Ao escolher a opção [*descriptor*] o sistema busca quais *ids* de descritores relacionados aos nós já foram inseridos e fornece uma lista para que o usuário escolha um.

Tabela 17: Composer (reprojeto) – Sequência de operações para criação de um nó de mídia na visão textual

Gerenciar âncoras

Procuramos apresentar na janela de propriedades de um nó todas as informações referente a ele. Optamos por informar além da lista de âncoras, a possibilidade de adicionar/deletar/editar cada âncora nesta mesma janela (Figura 101).

Node properties

Properties

* node id:

source:

type:

Content Anchors

[new](#) [delete](#)

Interval1 - 0 - 9
Interval2 - 10 - 21

Properties

* anchor id:

begin: seconds

end: seconds

Property Anchors

[new](#) [delete](#)

<input type="checkbox"/> top
<input checked="" type="checkbox"/> width
<input checked="" type="checkbox"/> soundLevel

Properties

name:

Descriptor

[new...](#)

Refer

Code preview

```
<media type="video" id="video1" src="media/video1.mpg"
descriptor="dVideo1">
<area id="interval1" begin="0s" end="10s"/>
<area id="interval2" begin="10s" end="20s"/>
</media>
```

* Required fields

Error Message: This id already exists.

Figura 101: Composer (reprojeto) – Propriedades de um nó

Caso o usuário queira adicionar uma âncora, basta clicar na opção [*new*] e preencher no formulário as propriedades da âncora. Caso queira editar uma âncora, basta selecioná-la na lista e o formulário irá carregar suas propriedades. Na própria lista as informações relevante da âncora aparecem. O exemplo mostra uma âncora de *id Interval1* que começa no instante 0s e termina no instante 9s (Interval1 – 0 – 9).

Na caixa de âncoras de propriedades aparecerão todos os possíveis valores, e o usuário poderá marcar/desmarcar as âncoras necessárias ou adicionar novas.

Ao modificar as âncoras, a caixa de código irá apresentar a linha do código referente a alteração. A ação [*New...*] da caixa de descritores irá abrir a janela de criação de descritores mostrada anteriormente. A caixa *Refer* permite selecionar outros nós para serem referenciados.

A figura mostra, ainda, a apresentação de uma mensagem de erro no final da janela, indicando que o *id* escolhido já está sendo usado.

Gerenciar contextos

A janela de criação ou edição de contextos é a mesma. O usuário deve informar o *id* para a criação e eventualmente a referência a um outro contexto previamente definido. O sistema lista as portas, nós de mídia, elos e outros contextos que possam ser inseridos dentro do contexto a ser criado/editado. O usuário pode selecionar os que irão fazer parte dele. As portas são mostradas de acordo com a relação *id > component*; os nós de mídia de acordo com a relação *id > descriptor*.

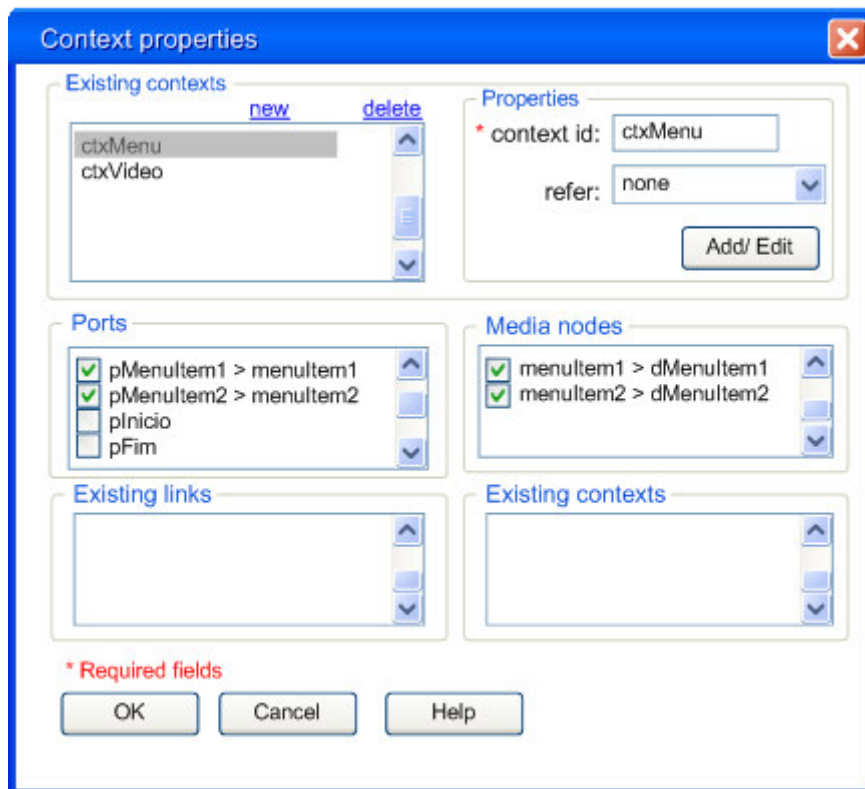


Figura 102: Composer (reprojeto) – Propriedades de um contexto

Gerenciar elos

A Figura 103 mostra a tela de criação de um elo. Após informar um *id* único para o elo, o usuário deve escolher um conector. Abrirá um painel com as diversas partes do conector a serem preenchidas. Na figura, o usuário escolheu um conector *onBeginStartN*. Na região 1 o usuário pode adicionar (ou remover) vários *linkParam*. Na região 2 o usuário configura o conector de acordo com os papéis.

No papel em destaque, o usuário adicionou um *bindParam* através do botão [+ *bindParam*]. Como este conector pode ter mais de um papel *start*, aparece uma opção [*add bind using role*]. Neste caso, a única opção de seleção de papel é *start*, porém poderíamos ter mais opções em outros casos. A ordem dos *bind* pode ser alterada porque este conector possui *qualifier – seq*. Assim como nas outras janelas de propriedades, o usuário pode ver o código que está sendo editado.

Figura 103: Composer (reprojeto) – Propriedades dos links

Gerenciar conectores

A Figura 104 mostra a tela de propriedades dos conectores. Nela, é possível criar um novo conector ou selecionar um (podendo selecionar uma base da lista de bases acessíveis pelo documento) – região 1 da figura.

A janela mostra outras cinco regiões. Na região 2 o usuário escolhe papéis para adicionar no painel de papéis selecionados. A região 3 mostra um exemplo de um conector que contém dois papéis: *onBegin* e *start*. Ao selecionar um papel na região 3, as definições do mesmo aparecem na região 4. As informações variam de acordo com o papel escolhido e podem ser alteradas. A região 5 mostra a lista de parâmetros associados ao conector. Esta lista pode ser modificada, renomeando, adicionando ou removendo um parâmetro. A região 6 mostra a

descrição textual do conector. Os valores entre “<” e “>” mostram *placeholders* de elementos que sofrerão as ações referente aos papéis.

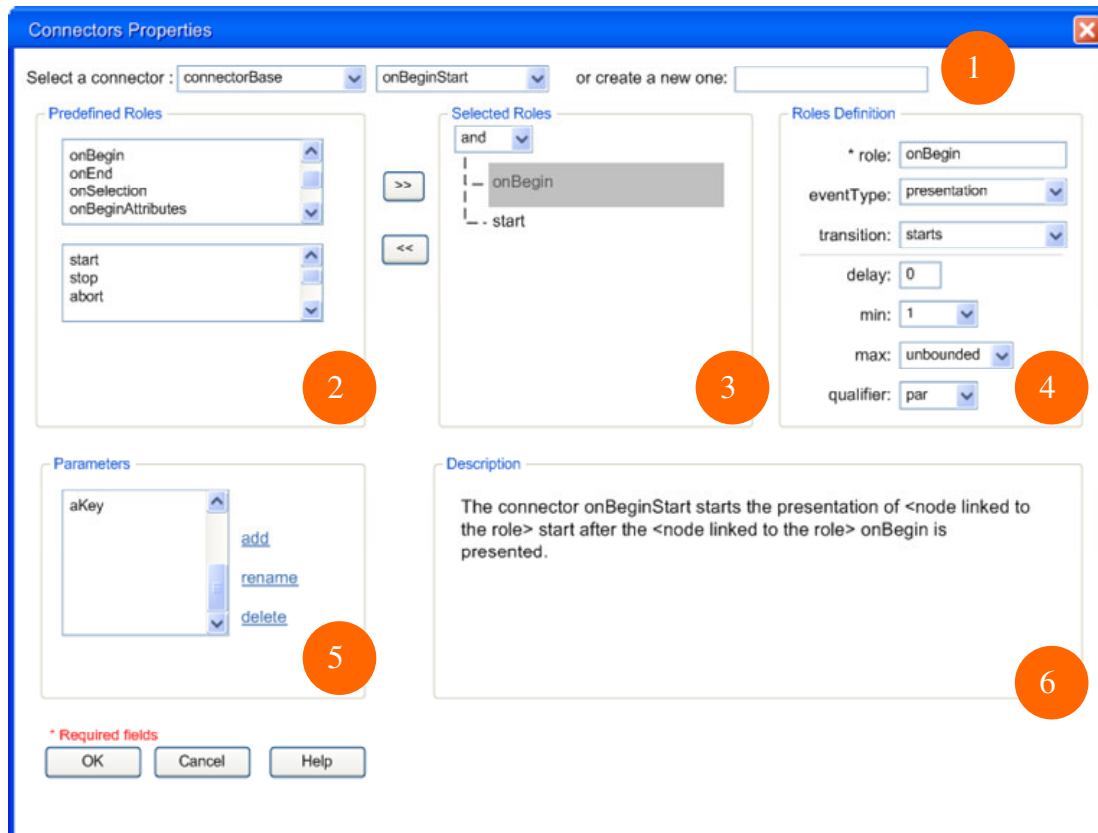


Figura 104: Composer (reprojeto) – Propriedades dos conectores

Gerenciar regras

As regras podem ser criadas escolhendo a referente opção do menu. A partir desta opção, o usuário deverá preencher os parâmetros da regra como mostra a Figura 105. Ele deverá fornecer um *id* a qual propriedade a regra está se referindo, qual o comparador a ser usado (o sistema carrega automaticamente a lista de possíveis comparadores) e uma opção para inserção do valor da regra.

Há um opção para inserir uma nova regra nesta mesma janela. A partir da segunda regra adicionada, aparecerá um botão para exclusão da regra, deixando sempre pelo menos uma regra (Figura 106).

Create rule...

Rule

* rule id:

refer to: (var)

comparator:

value:

[add one more](#)

Code preview

```
<ruleBase>
  <rule id="rSubtitleOn"
    var="subtitle" comparator="eq" value="on" />
</ruleBase>
```

* Required fileds

Figura 105: Composer (reprojeto) – Propriedades de uma regra

Create rule...

Rule

* rule id:

refer to: (var)

comparator:

value:

* rule id:

refer to: (var)

comparator:

value:

[add one more](#)

Code preview

```
<ruleBase>
  <rule id="rSubtitleOn"
    var="subtitle" comparator="eq" value="on" />
  <rule id="rRule1"
    var="insertValue" comparator="eq" value="insertValue" />
</ruleBase>
```

* Required fileds

Figura 106: Composer (reprojeto) – Propriedades de uma regra, duas regras sendo criadas ao mesmo tempo

Os mecanismos de manipulação direta e programação via menu (como no caso da criação de conectores e elos, são similares aos apresentados por diversos sistemas de EUD revisto da literatura, porém não se enquadram na classificação e fluxograma apresentado no capítulo 3. Por isto, não serão tratados como adaptação, e sim como auxílio a criação de programas. O mesmo pode ser considerado sobre o *preview* do código. Ele pode ser caracterizado como aprendizado por desvelamento, uma vez que mostra ao usuário o preenchimento do código na medida em que ele executa uma tarefa. Apesar disto, não consideramos isto como sendo uma adaptação, e sim um recurso de design de apoio a uma atividade de programação.

O Composer foi criado inicialmente utilizando o idioma inglês. Porém, como nasceu no Brasil e faz parte de um projeto que almeja a criação de programas para a TV Digital brasileira, é necessário que possa ser traduzido para o português e, eventualmente, para outras línguas. Desta forma, podemos considerar que existem diferenças ou preferências lingüísticas entre os usuários. Assim, seria interessante propor uma adaptação do idioma através da parametrização, no diálogo de alteração das opções do sistema.

Segundo questões de preferências dos usuários, podemos definir adaptações no sentido de alteração do léxico do sistema, ou seja, da apresentação. O usuário pode definir se deseja visualizar ou não: ícones e/ou os nomes das mídias e os *ids* dos elos ou dos conectores usados por eles. O fluxograma a seguir ilustra como a adaptação do léxico pode ser empregada segundo as preferências do usuário.

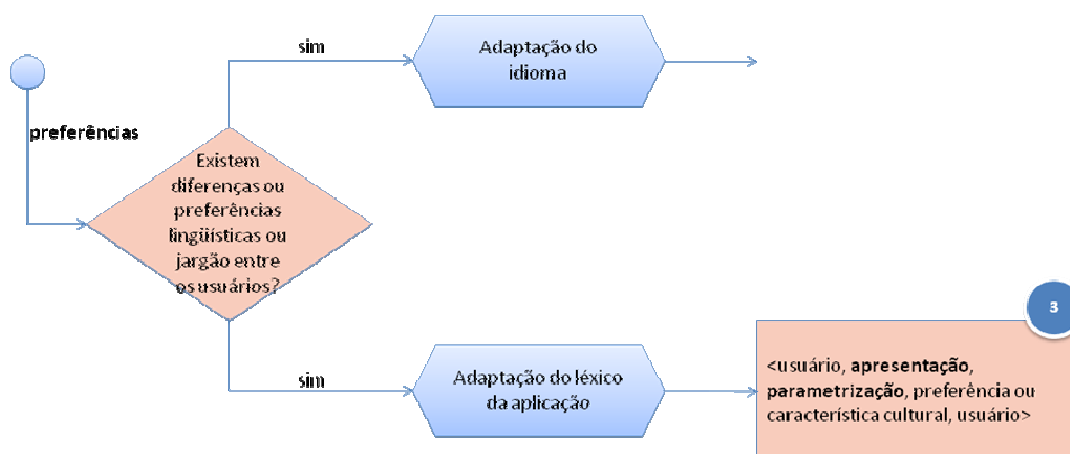


Figura 107: Fluxograma da adaptação do léxico do sistema

Outra funcionalidade interessante a ser incorporada no Composer é a de distinção de perfis de uso. O projeto poderia ser configurado para aceitar determinadas alterações relacionadas aos perfis. Por exemplo, roteiristas podem mudar toda a estrutura do programa e os produtores de mídia poderiam apenas substituir as mídias fazendo alterações apenas nas mídias existentes. Assim, temos um mapeamento para a realização de uma tarefa de acordo com o usuário.

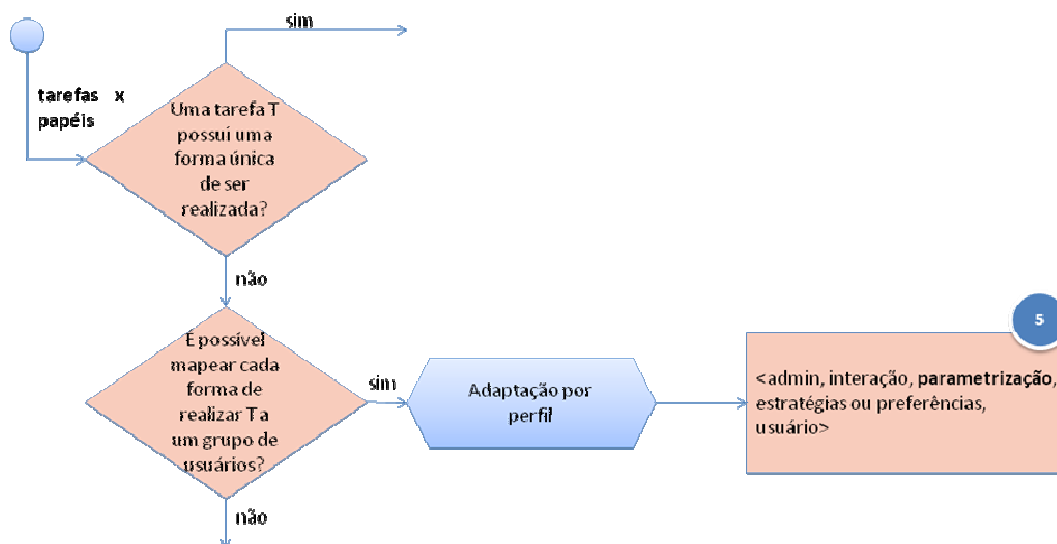


Figura 108: Fluxograma da adaptação por perfil

Uma outra forma de configuração de parâmetros de uma tarefa, além do destacado sobre a configuração do local de armazenamento dos arquivos e pastas de mídias descritos anteriormente, é a definição da forma de visualizar um programa. Ou seja, o usuário pode definir que a área de visualização do programa deve ser em tela inteira ou em uma janela. Isto é uma forma de alteração do comportamento do sistema, segundo as preferências dos usuários.

No decorrer das explicações dos esboços de tela falamos sobre a utilização de *wizards* como forma de integração externa. Porém, podemos sugerir também como forma de auxiliar o programador na tarefa de criação do programa NCL, o uso de moldes de programas (ou parte de programas) frequentes. Teríamos nesse caso uma composição declarativa, onde o usuário iria compondo o programa adicionando partes prontas. Isto advém da resposta sobre o sistema ter um conjunto de funcionalidades bem definidas, por exemplo, ter uma associação entre duas mídias com um elo e permitir que, ao adicionar este bloco ao programa, o usuário possa alterar a mídia e o tipo de conector.

Algumas tarefas no Composer podem ser consideradas frequentes e de pouca variação, como por exemplo, empacotar um documento NCL copiando os conectores para o próprio arquivo NCL. Para tarefas desse tipo, seria interessante disponibilizar mecanismos de gravação de macro, cujas funcionalidades possam ser facilmente reconhecidas pelo usuário. Isto quer dizer que não estamos sugerindo a criação de uma nova linguagem de interpretação do sistema. Deve ser algo simples que não exija do usuário a necessidade de aprendizado de uma nova linguagem caso queria fazer pequenas alterações na macro gravada, como ocorre nas macros do Adobe Photoshop®. O fluxograma a seguir ilustra este processo.

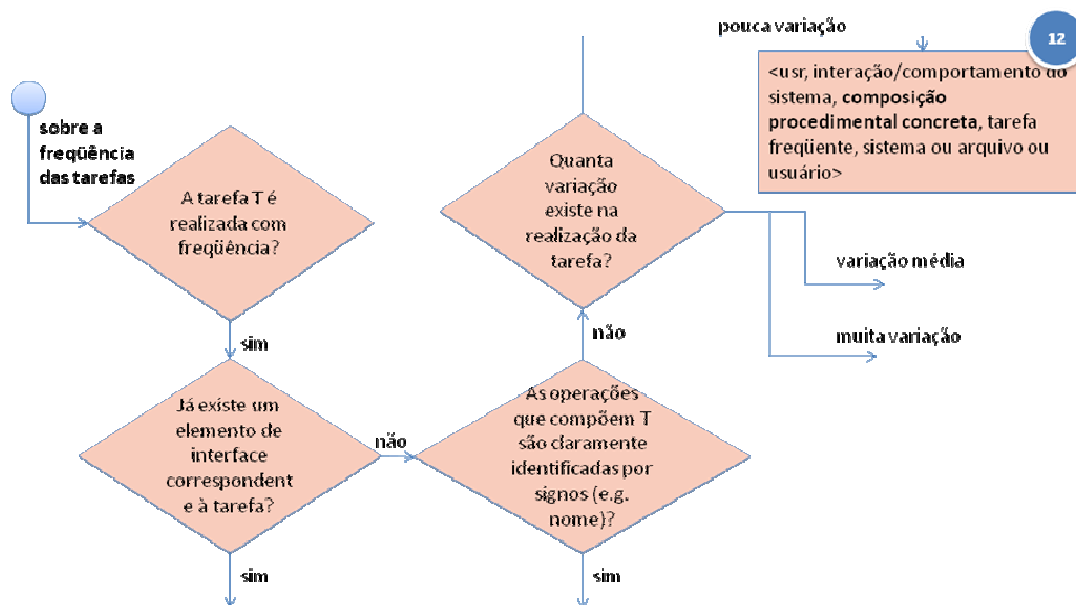


Figura 109: Fluxograma para empacotar documento copiando bases

É importante considerar também o fornecimento de mecanismos de parametrização para a criação de teclas de atalho, itens de menu ou botões para barras de ferramentas que possam ser associados às macros.