

2 Trabalhos Relacionados

Ao considerar várias etapas do processo de criação de uma aplicação hipermídia, especialmente a de projeto, na qual a técnica de *storyboard* melhor se encaixa, durante a análise de outros trabalhos foi preciso considerar também as ferramentas cujo foco está concentrado em diferentes partes desse processo de criação. Este capítulo lista, entre essas ferramentas, aquelas que foram mais relevantes para este trabalho, procurando destacar suas limitações, soluções e como elas influenciaram em algumas decisões de projeto.

Para deixar mais claro quais tipos de características e funcionalidades foram investigados para o desenvolvimento do trabalho, as ferramentas listadas neste capítulo são divididas em: ferramentas de projeto, ferramentas de *storytelling* e ferramentas de autoria multimídia. Inicialmente, são descritas as ferramentas de projeto, cujo foco está no planejamento e na prototipação, e que demonstram uma preocupação maior com o leiaute e navegação das aplicações. Em seguida, são listadas as ferramentas de *storytelling*, cujo interesse está na criação de narrativas interativas e não lineares. E por fim, são apresentadas as ferramentas de autoria multimídia, incluindo aquelas voltadas à TV digital interativa.

É importante ressaltar que a divisão adotada não é exclusiva, ou seja, uma ferramenta poderia estar em mais de um grupo. No entanto, elas foram mantidas em apenas um, considerando aquela característica que foi mais representativa para o trabalho.

2.1 Ferramentas de projeto

Apesar de não ser um requisito deste trabalho, as ferramentas apresentadas nesta seção são ferramentas baseadas em caneta (*pen-based*). A motivação para análise desse tipo de ferramenta é investigar quais características tornam essas ferramentas mais atrativas do que a prototipação por lápis e papel. Como este trabalho se propõe a utilizar a técnica de *storyboard*, essa análise é importante.

Basta lembrar que a técnica de *storyboard* vem sendo adotada por anos e, até hoje, o lápis e papel vem sendo utilizado na criação de *storyboards*.

A **SILK** (*Sketching Interfaces Like Crazy*) (Landay, 1996) é uma ferramenta para o esboço de interface. A SILK mantém o foco nos estágios iniciais do projeto de interface, permitindo explorar diferentes ideias de projeto com um esforço mínimo de trabalho. A ferramenta procura preservar propriedades importantes da prototipação por lápis e papel, como: rapidez e flexibilidade. No entanto, a SILK permite que os esboços produzidos na ferramenta sejam interativos.

Através de um editor de *storyboard* (Figura 2.1) é possível especificar como a tela deve mudar em resposta às ações tomadas pelo usuário. Dessa forma, tanto o projetista quanto os usuários podem avaliar a navegação da ferramenta no estágio de rascunho da interface.

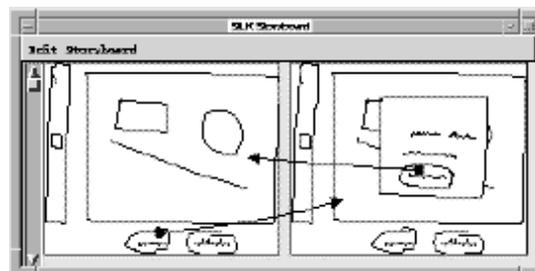


Figura 2.1 – Editor de *storyboard* do SILK (Landay, 1996)

A Figura 2.1 mostra um exemplo simples de como essa navegação é especificada. O esboço na Figura 2.1 define que uma caixa de diálogo deve aparecer (lado direito) quando o usuário clicar em um dos botões da tela (lado esquerdo). E que, logo depois que o usuário apertar no botão da caixa de diálogo (lado direito), a aplicação deve voltar para tela anterior (lado esquerdo).

Além da interatividade, a SILK explora outros pontos que não podem ser considerados na prototipação por lápis e papel. O reúso é o principal deles. Constantemente o projetista tem que redesenhar elementos que não mudam. A SILK permite utilizar partes de esboços antigos, evitando que projetista tenha que começar a partir de uma tela em branco, o que ocorreria se estivesse trabalhando com o lápis e papel. Isso impacta diretamente a quantidade de versões que podem ser exploradas em um projeto, o que talvez resulte em um produto de melhor qualidade.

Outro ponto importante da ferramenta é a “memória de projeto”. Rascunhos são comumente anotados, entre outras razões, para tentar deixar mais claro o

motivo por trás de determinada decisão ou para indexar e organizar o projeto. O SILK dispõe de um mecanismo de busca que permite que as anotações realizadas nos esboços possam ser facilmente achadas. Dessa forma, é possível agilizar ainda mais o processo de criação.

Em uma etapa final, a SILK permite ainda que os rascunhos sejam substituídos por objetos gráficos reais (*widgets*). A maior parte desse processo é automatizado, necessitando de assistência somente em alguns detalhes da interface (e.g., *labels*, cores, etc.). Depois desta etapa, a aplicação resultante pode ser entregue aos programadores para integrar o código gerado com o restante do sistema.

Todas essas funcionalidades caracterizam a SILK como uma ferramenta que dá suporte a todo o ciclo de projeto – indo da etapa inicial e criativa, ao desenvolvimento do protótipo, teste e implementação final da interface.

No entanto, quando consideramos o projeto de aplicações multimídia, a SILK ainda é limitada. O editor de *storyboard* da SILK suporta apenas transição entre telas, desconsiderando, por exemplo, requisitos como o sincronismo temporal e a utilização de mídias contínuas, que fazem parte de uma apresentação multimídia.

Mesmo com essa limitação com relação às aplicações multimídia, a SILK permitiu identificar alguns dos requisitos básicos deste trabalho: o reúso e a navegação. O reúso no sentido de evitar retrabalho, e a navegação no sentido de tornar o esboço inicial interativo, tornado possível avaliar a navegação da aplicação ainda nos estágios iniciais de projeto.

A **DEMAIS** (*DEsigning Multimedia Applications with Interactive Storyboards*) (Bailey et al., 2001) possui motivações semelhante às da SILK. A DEMAIS também está interessada nos estágios iniciais do processo de criação, porém, ao contrário da SILK, mantém o foco nas aplicações multimídia. Assim, ela procura ir além das funcionalidades presentes na SILK, permitindo, por exemplo, definir elos temporais, interações mais expressivas e a utilização de mídias contínuas no projeto da aplicação.

Para isso, a DEMAIS dispõe de vários componentes. Esses componentes são: *Storyboard Editor*, *Narration Editor*, *Content Manager* e *Multi-View Editor*.

O *Storyboard Editor* (Figura 2.2) permite ao projetista desenhar livremente o esboço da interface de sua aplicação. Ao contrário da SILK, possibilita ainda

que sejam definidos relacionamentos internos ao *storyboard*. Esses relacionamentos são definidos conectando as entidades do esboço através de linhas e um conjunto padronizado de ícones para determinar o comportamento desses relacionamentos. De modo geral, esses ícones apresentam um formato circular, para especificar o evento que dispara uma relação, e um formato quadrado, para fazer referência à ação que será tomada quando a relação for disparada.

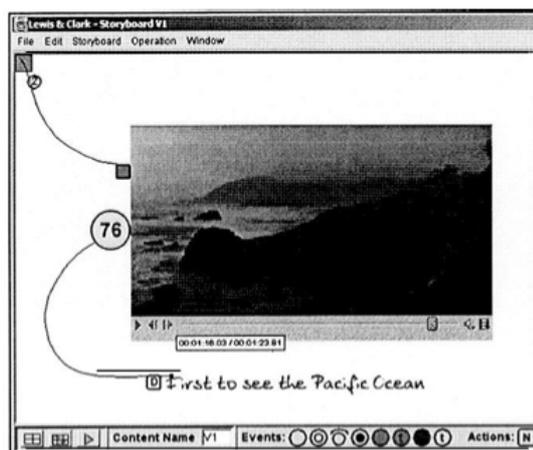


Figura 2.2 – Storyboard Editor do DEMAIS (Bailey et al., 2001)

O esboço da Figura 2.2, por exemplo, define dois relacionamentos. O primeiro especifica que um vídeo deve ser iniciado automaticamente, assim que a apresentação for iniciada. Na DEMAIS, todo *storyboard* possui uma entidade que representa o início da apresentação, essa entidade se localiza no canto superior esquerdo e possui o formato de um quadrado. Com esta entidade, é possível definir relacionamentos para determinar que ações deve ser tomadas no início da apresentação, o que, no exemplo da Figura 2.2, é iniciar o vídeo. O segundo relacionamento definido na Figura 2.2 especifica que um texto deve ser exibido assim que o vídeo estiver no segundo 76. A condição de disparo da relação, o 76 segundo do vídeo, é definida através dos controles do vídeo, posicionando a barra de progresso no instante desejado. Assim o projetista sabe inclusive em que cena do vídeo determinada ação será tomada.

Com *Narration Editor* o projetista pode escrever um texto que será sintetizado, possibilitando que a apresentação multimídia seja narrada. No texto, podem ser definidos pontos de sincronismo, de modo que possam ser estabelecidos relacionamentos com partes do texto. O projetista pode inclusive gravar a própria voz e sincronizar o áudio com os pontos de sincronismos

definidos no texto, descartando o sintetizador. O *Content Manager* tem como principal funcionalidade o reúso de conteúdo e a visualização de toda a produção do projeto. O *Content Manager* permite recortar, copiar, colar um *storyboard* ou uma narração, além de possibilitar ao projetista alternar entre o conteúdo que vai ser editado.

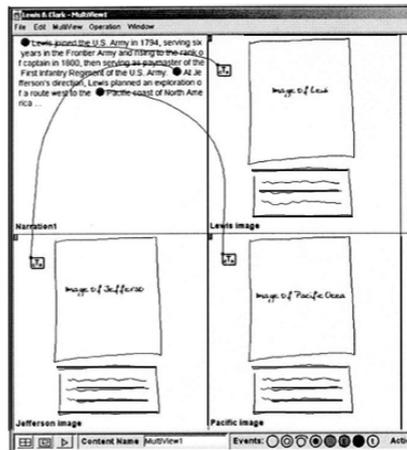


Figura 2.3 – Multi-View Editor do DEMAIS (Bailey et al., 2001)

O último componente, o *Multi-View Editor* (Figura 2.3), permite definir relacionamentos entre *storyboards*, entre narrações e entre ambos. Como no *Storyboard Editor*, os relacionamentos nesse componente também são especificados através de linhas e os ícones padronizados da DEMAIS.

Apesar de manter o foco no estágio inicial do projeto da aplicação, quando o projetista trabalha em uma aplicação envolvendo vídeo, por exemplo, ele acaba perdendo um pouco da flexibilidade do processo de criação. Ao trabalhar com vídeo, a DEMAIS solicita que esse conteúdo seja especificado, fazendo com que o projetista tenha que se comprometer com a produção de um vídeo ainda na etapa de rascunho da aplicação. Caso o projeto desse conteúdo fosse realizado em conjunto com a aplicação, ou seja, o vídeo pudesse também ser especificado na forma de rascunho, não haveria essa perda de flexibilidade e relações bem mais interessantes poderiam ser pensadas.

2.2 Ferramentas de *storytelling* interativo

O principal interesse na análise de ferramentas de *storytelling* está na análise dos tipos de estruturas que essas ferramentas utilizam para auxiliar o usuário na criação de narrativas não lineares. Em especial, essas ferramentas representam

narrativas complexas, em que a quantidade de caminhos pelos quais a história pode seguir é grande. O interesse por essas narrativas mais complexas existe, mas provavelmente não serão tão comuns no ambiente da TV digital interativa.

Quando consideramos a produção de um conteúdo não linear para TV digital, é preciso ter em mente que características como a qualidade das atuações, uma direção consistente, um roteiro profundo, ainda serão os fatores determinantes para a satisfação do telespectador. A possibilidade do telespectador decidir o andamento da história será um complemento, e que deve ser utilizado com cautela para não comprometer a coerência da história. Além disso, uma quantidade muito grande de interrupções ou interações pode não ser tão prazeroso. Por esse motivo, é possível que narrativas complexas não serão tão comuns no ambiente da TV digital interativa.

No entanto, a representação da estrutura da narrativa ainda é algo necessário para permitir ao autor, no mínimo, organizar melhor a sua história. E a análise desses casos limites certamente trará alguma informação de como representar essa estrutura de forma eficiente.

A **StoryTec** (Gobel et al., 2008) é uma plataforma para autoria e experimentação de histórias interativas e não lineares. Essa plataforma é composta de um ambiente de autoria e um de execução. O módulo de autoria é baseado em uma arquitetura de *plugin*, e sendo composto por cinco componentes visuais: *Story Editor*, *Stage Editor*, *Action Editor*, *Property Editor* e *Asset Manager*.

A nossa análise vai se restringir ao *Story Editor*, já que este é o componente que lida com a estrutura da narrativa, foco de nossa análise. Detalhes sobre os outros componentes podem ser vistos em (Gobel et al., 2008). O *Story Editor* é editor da estrutura da narrativa. Na StoryTec, uma narrativa é estruturada por um grafo hierarquicamente organizado composto por duas camadas: as cenas e as cenas complexas. As cenas complexas definem tipicamente uma localização no mundo da história e podem conter uma ou mais cenas. As cenas podem definir um estado específico da história, uma parte lógica da história ou os elementos ativos da história (e.g. personagens virtuais, objetos interativos, etc.). As cenas só podem existir dentro de uma cena complexa.

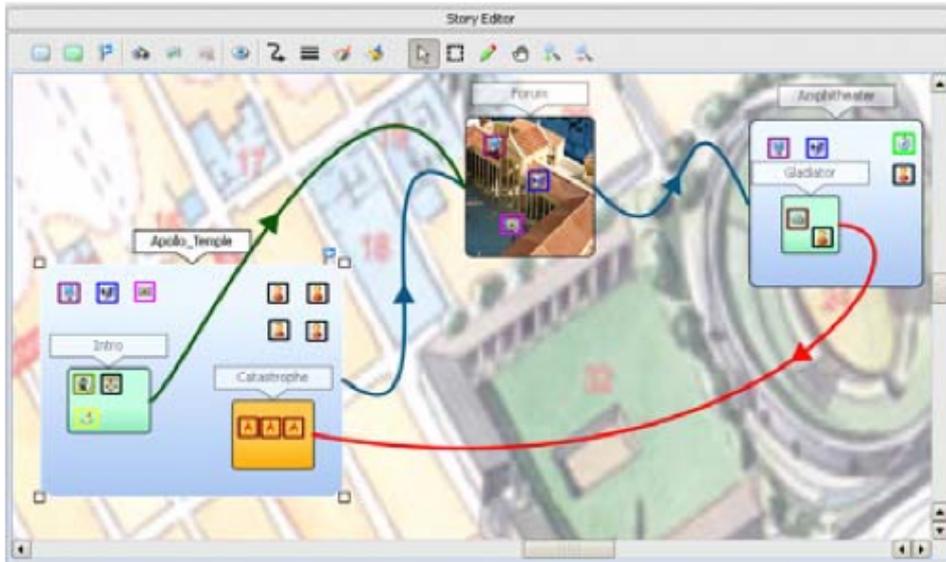


Figura 2.4 – Story Editor do StoryTec (Gobel et al., 2008)

O *Story Editor* (Figura 2.4) é responsável por representar essa estrutura, contendo além de cenas e cenas complexas, as transições entre esses elementos. Para lidar com estruturas complexas, o *Story Editor* dispõe de um *zoom* semântico, adotando diferentes representações para os elementos da estrutura, conforme o nível de detalhe especificado pelo *zoom*. A escolha pela representação em forma de grafo partiu de testes de usabilidade que foram efetuados com esse componente. Nessa avaliação, os participantes envolvidos chegaram à conclusão que esse tipo de representação é altamente recomendado para criação de histórias interativas.

A **Story Canvas** (Skrorupski & Mateas, 2010) é uma ferramenta de autoria para criação de narrativas interativas e generativas, sendo essas últimas, aplicações a partir das quais várias narrativas interativas podem ser geradas. No *Story Canvas*, a estrutura da narrativa é definida por um grafo composto de *goals* e *fragments*. O autor começa a compor a sua história definindo o *goal* principal. Na Figura 2.5 os *goals* são representados pelas linhas tracejadas. Dentro de cada *goal* podem ser definidos *fragments*. Os *fragments* definem o conteúdo da história. Associada a cada um deles, existe uma pré-condição para que esse conteúdo seja considerado na história. A partir de um *fragment* podem ser definidos *subgoals*, para garantir a coerência da história ou permitir a interação do usuário. Por exemplo, se no conteúdo de um *fragment* é a aparição de um novo personagem aparece na história, um *subgoal* seria: introduzir esse novo personagem na história. Somente depois que esse *subgoal* fosse cumprido a

narrativa poderia seguir por aquele caminho. O conteúdo e as pré-condições de cada *fragment* são definidos em outra visão.

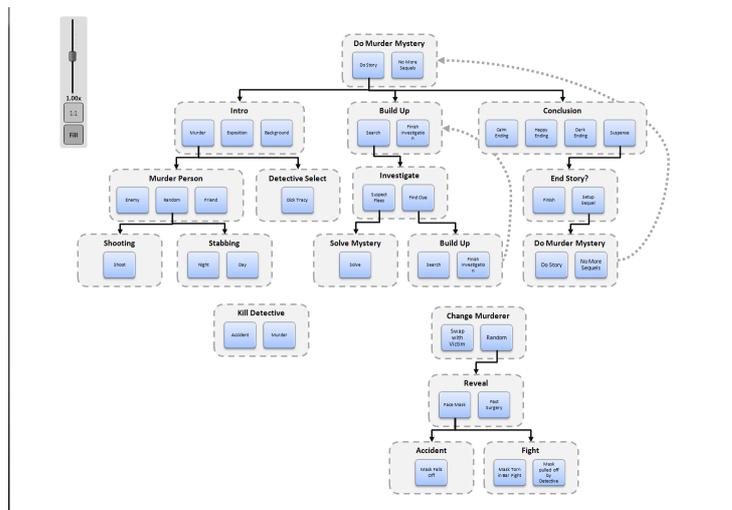


Figura 2.5 – A interface da estrutura da história do Story Canvas (Skrorupski & Mateas, 2010)

Através dessa estrutura de *goals*, *subgoals*, *fragments* e suas pré-condições associadas, a Story Canvas permite geração de histórias interativas. Para lidar com narrativas complexas, a Story Canvas também utiliza um *zoom* sobre estrutura.

De modo geral, não importa o conceito por trás da estrutura de uma narrativa interativa, a estrutura em grafos parece ser a mais aceitável para representação da estrutura da história. E, no caso das estruturas complexas, o simples *zoom* parece ser o suficiente para ajudar o usuário a lidar com essa estrutura.

2.3 Ferramentas de autoria multimídia

No caso das ferramentas de autoria multimídia, a análise procurou investigar a flexibilidade e rapidez na criação de uma aplicação multimídia, as abstrações utilizadas para lidar com a complexidade das tecnologias envolvidas, e como essas ferramentas dão suporte ao projeto do conteúdo multimídia.

Adobe Flash (Adobe Flash, 2011) é uma plataforma de autoria amplamente utilizada para criação de conteúdo para Web. Essa ferramenta é composta de duas visões principais: a visão espacial, chamada de *stage*, e uma visão temporal. Em conjunto, as duas visões funcionam como uma poderosa ferramenta de edição espaço/temporal.

A especificação da interatividade é realizada através da codificação de *scripts ActionScript* (ActionScript, 2011). Mesmo a interatividade mais simples precisa ser codificada, o que é consideravelmente complicado para autores não programadores.

O **LimSee2** (LimSee2, 2011) é uma ferramenta de autoria para documentos SMIL (W3C, 2008). O LimSee2 utiliza a abordagem de múltiplas visões integradas, para a composição de documentos hipermídia. A ferramenta possui cinco visões: a estrutural, a temporal, a espacial, a de propriedade e a textual. Mesmo utilizando abstrações gráficas e outros recursos, a maioria dessas visões ainda carregam uma carga muito grande dos conceitos da linguagem alvo. Isso exige que o usuário tenha ao menos um conhecimento básico sobre linguagem, para criar uma aplicação de qualidade.

O LimSee2 coexiste com o **LimSee3** (Deltour & Roisin, 2006). Este último tem como objetivo dar suporte a arquétipos de documento SMIL para facilitar a geração de documentos em formatos padronizados. Esse paradigma é baseado em famílias de documentos adaptáveis para diferentes categorias de autores e aplicações.

O **Icareus iTV Suite Author** (Icareus, 2011) é uma ferramenta de autoria para desenvolvimento de aplicações para o *middleware* MHP. A ferramenta utiliza a abstração de cenas. Cada cena possui um conteúdo próprio que será apresentado no momento da execução da aplicação. O que determina a mudança de cena é a interatividade. O público alvo da ferramenta é bem semelhante ao deste trabalho. No entanto, o foco dessa ferramenta está na interatividade, deixando de lado o sincronismo e a possibilidade de estabelecer uma semântica maior entre a interação e o conteúdo do vídeo principal.

O **ShapeShifting TV** (Ursu et al., 2008) é um paradigma para criação de narrativas interativas para TV. O ShapeShifting TV possui motivações bem semelhantes com a desse trabalho. O ShapeShifting TV está interessado na criação de aplicações em que o telespectador possa interferir no conteúdo principal da aplicação.

O ambiente de autoria desse paradigma é bem completo. Possui visões para edição de ontologias e anotações dos objetos de mídia. Neste paradigma existe a possibilidade de fazer relacionamentos semânticos entre os objetos de mídia, daí a necessidade de uma visão para anotação de mídia e edição de ontologias. Possui

também um conjunto de visões para trabalhar com a estrutura da aplicação e a pré-visualização da narrativa.

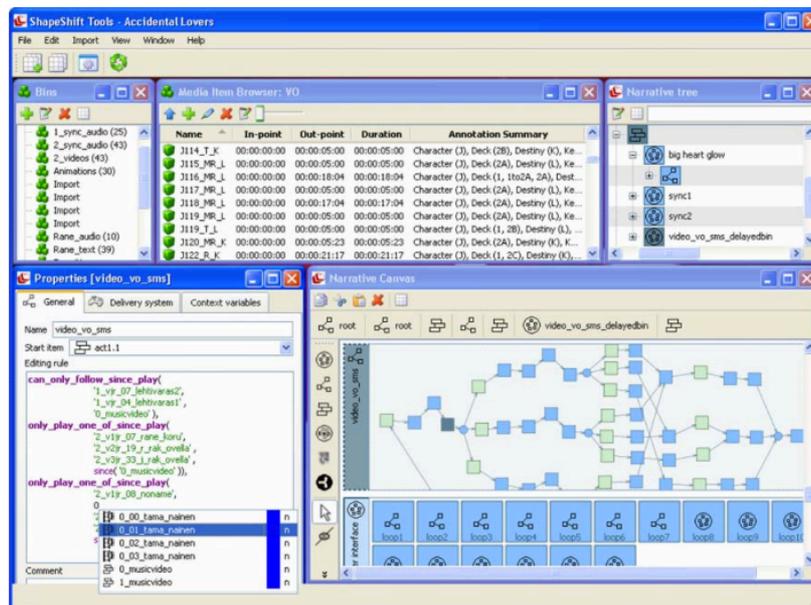


Figura 2.6 – Ambiente de autoria do ShapeShifting TV (Ursu et al., 2008)

A Figura 2.6 apresenta parte dessas visões, em especial as duas visões para edição da estrutura da narrativa. O principal problema do ShapeShifting TV está relacionado ao projeto da aplicação. Até mesmo a autoria de um exemplo simples. A ferramenta exige do usuário uma quantidade muito grande de informação para que uma aplicação seja criada. Isso faz com que o ambiente de autoria do ShapeShifting TV não seja adequado para que o autor possa explorar várias idéias de projeto.

O **StoryToCode** (Neto & Santos, 2009) define um modelo para especificação de aplicações para TV digital interativa. Ele procura formalizar os elementos presentes no *storyboard* através de uma linguagem formal, para que seja possível a geração de aplicação nas mais diferentes linguagens. A dificuldade do StoryToCode ocorre no processo de extração das informações relevantes do *storyboard*. Essa extração é realizada somente por engenheiros de *software* e de forma manual, criando uma barreira para a geração da aplicação final pelo *designer*.

A **NCLite** (*NCL Integrated Tailoring Environment*) (Encarnação & Barbosa, 2010) é uma ferramenta de autoria que permite criar aplicações NCL abstraindo os detalhes da linguagem. A NCLite utiliza um conceito de cenas

próprio, mas a sua abordagem se assemelha com a do Icareus iTV Suite Autor. A grande diferença entre os dois trabalhos é o suporte à edição das características temporais oferecido pela NCLite.

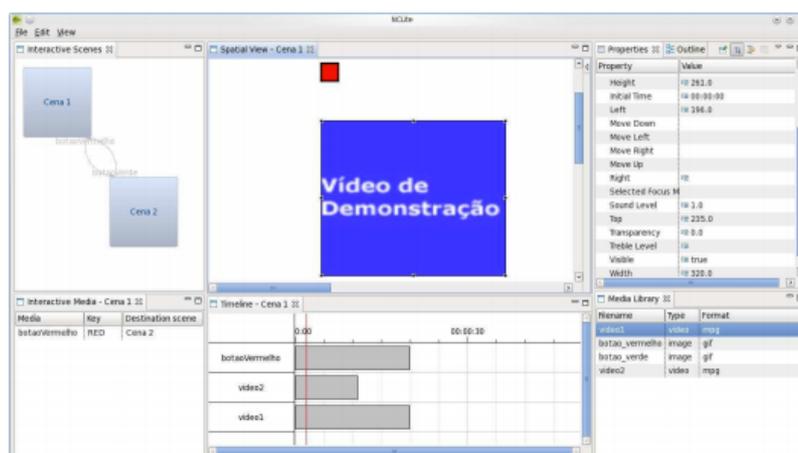


Figura 2.8 – Interface da NCLite (Encarnação & Barbosa, 2010)

A NCLite tem o foco na interatividade de forma que ele possa ser utilizado em conjunto como uma aplicação de edição de vídeo. A crítica à NCLite é exatamente em cima dessa última característica. A NCLite separa a especificação da interatividade da produção do conteúdo audiovisual, o que, como já vimos, possivelmente dificulta a produção de aplicações com uma interatividade mais interessante.

A primeira versão do **NCL Composer** (Guimarães & Soares, 2007) se apóia na abordagem de múltiplas visões integradas para a composição do documentos hipermídia, nesse caso, na linguagem NCL. A ferramenta foi inicialmente concebida tendo como público-alvo não programadores com pouco ou nenhum conhecimento em NCL.

Cada visão do NCL Composer dá ao autor uma perspectiva diferente e específica sobre o documento. A Figura 2.9 apresenta as quatro visões da ferramenta: a visão estrutural, a temporal, a de layout e a textual.

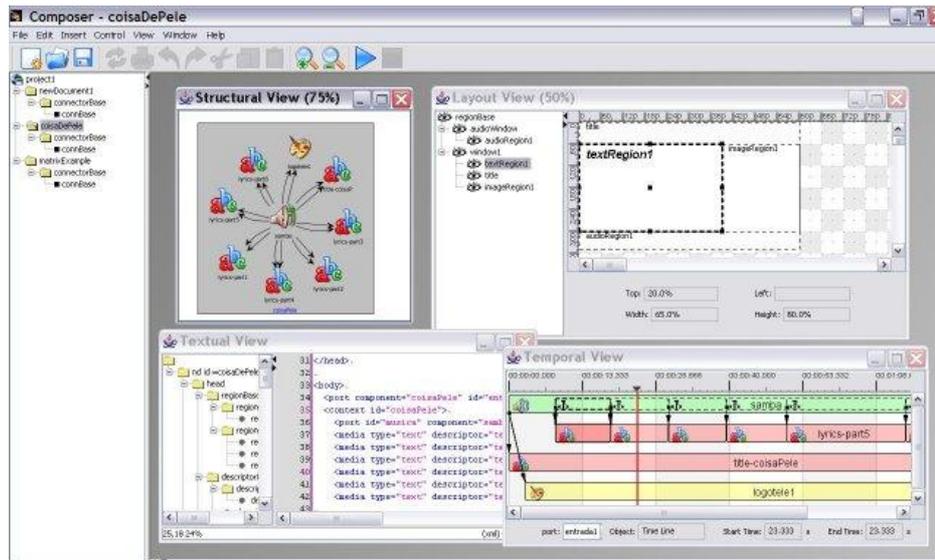


Figura 2.9 – Interface do antigo NCL Composer (Guimarães & Soares, 2007)

A nova versão do NCL Composer (Lima & Soares, 2011) é baseada no padrão arquitetural de micro-núcleo e extensões. Ele foca nos requisitos não funcionais para uma ferramenta de autoria de documentos hipermídia (e.g. customabilidade, extensibilidade, eficiência, portabilidade, etc.). A sua arquitetura extensível e personalizável permite à ferramenta atender a diferentes ambientes de desenvolvimento e a diferentes perfis de usuários, bastando para isso apenas instalar as extensões apropriadas.

Atualmente a nova versão do NCL Composer possui sete extensões: a visão de leiaute, a visão estrutural, a visão de *outline*, a visão de propriedade, o *plugin* de pré-visualização de objetos de mídia, a visão textual e um *plugin* de validação do documento NCL. A Figura 2.10 apresenta a interface do novo NCL Composer.

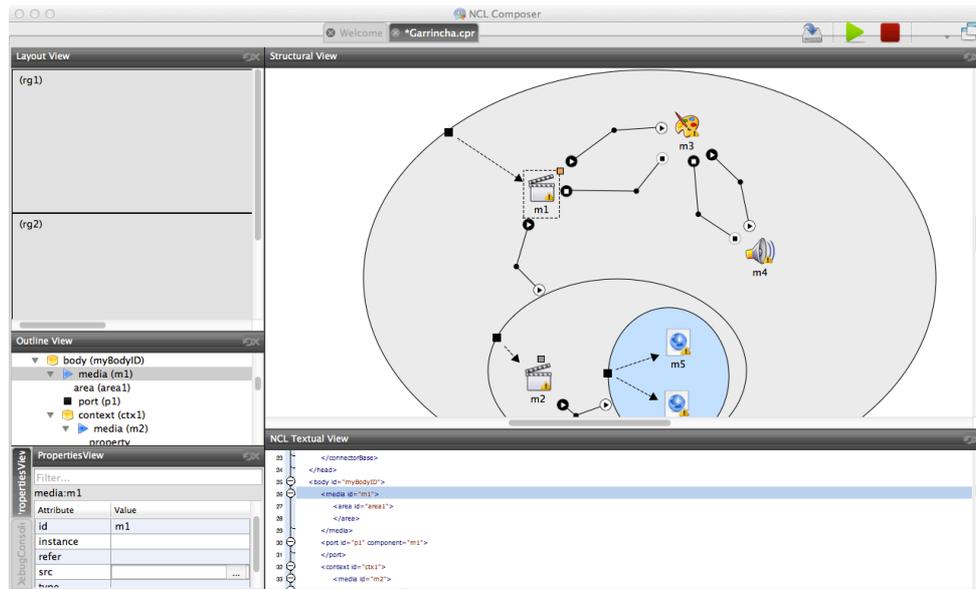


Figura 2.10 – Interface do novo NCL Composer

Os problemas do NCL Composer, tanto na versão anterior como na versão mais recente, é que as visões ainda estão próximas dos conceitos relacionados à linguagem NCL. Isso exige um conhecimento básico da linguagem para poder lidar com as abstrações. Porém, por conta da capacidade de extensão do novo NCL Composer esse problema pode ser solucionado com a simples integração de abstração de mais alto nível em sua arquitetura.