

5 Implementação

Neste capítulo são apresentados os detalhes sobre a implementação da ferramenta. São discutidas as tecnologias envolvidas, assim como as limitações e problemas encontrados durante o desenvolvimento. Ao final, são discutidas algumas produções de aplicações que foram realizadas com a ferramenta, como forma de testar a implementação.

5.1 Plataforma

A linguagem utilizada para o desenvolvimento da ferramenta foi o C++ com *framework* Qt (Nokia Inc., 2011). O Qt é uma aplicação multiplataforma e um *framework* para criação de interfaces gráficas com APIs para C++. Através do Qt é possível criar códigos legíveis e de fácil manutenção.

O *framework* Qt em conjunto com várias outras ferramentas integradas com o próprio Qt formam a plataforma utilizada para o desenvolvimento deste projeto, denominada Qt SDK (*Software Development Kit*). Dentre essas ferramentas, destaca-se o Qt Creator, que é o ambiente de desenvolvimento da plataforma Qt. Este ambiente inclui editores C++ e *JavaScript*, ferramentas para o *design* de interfaces gráficas, gerenciamento de projeto, depuração de código, controle de versão, entre outras.

Além de características como robustez, desempenho, portabilidade e outras discutidas em (Lima & Soares, 2011), a escolha do Qt como plataforma também vai permitir uma fácil integração com a ferramenta NCL Composer, que foi desenvolvida sobre a mesma plataforma.

5.2 Arquitetura

A arquitetura utilizada para o desenvolvimento da ferramenta segue o padrão de arquitetura MVC (*model-view-controller*) (Gamma et al., 1997), com o objetivo de isolar os módulos da aplicação, permitindo desenvolver e testar cada um desses módulos separadamente. O MVC divide o projeto em três grandes

módulos ou componentes: o modelo, a visão e o controlador. O modelo é responsável por gerenciar a informação e notificar os observadores das mudanças nos dados. A visão é onde são definidas as interfaces de interação com o usuário. O controlador é responsável por receber as entradas de dados, validar e tomar as ações apropriadas sobre o modelo de dados.

Como discutido no Capítulo 3, a ISB Designer utiliza várias visões para organizar e facilitar a criação das aplicações. Para trabalhar com várias visões, no módulo de visão da arquitetura foi definido um ponto de extensão, em que cada uma dessas visões pudesse ser facilmente integrada. Além desse, outro ponto de extensão no módulo do controlador foi definido, para que a ferramenta pudesse ser facilmente integrada em outro *software*. A Figura 5.1 apresenta a arquitetura em grandes blocos da ISB Designer.

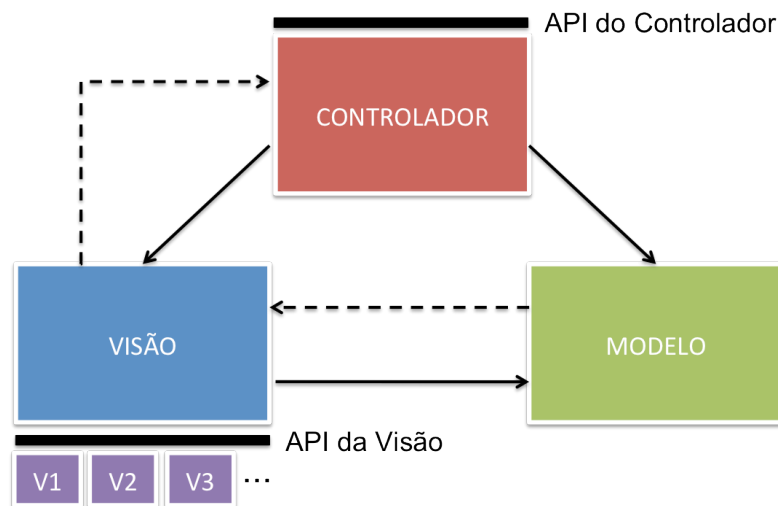


Figura 5.1 – Arquitetura da ISB Designer

Na implementação atual, todo o processo de comunicação entre os módulos é realizado pelo mecanismo de comunicação entre objetos do Qt, denominado *signal/slots*. Os *signal/slots* são abstrações especiais de métodos de uma classe C++. No mecanismo de *signal/slots*, os *signals* de um objeto são conectados a *slots* de outro, deixando para o Qt a responsabilidade de manter a lista dessas relações e realizar chamadas apropriadas quando um *signal* for emitido. O Qt permite inclusive que relações sejam criadas em tempo de execução, o que dá uma maior flexibilidade na especificação da comunicação entre os objetos.

A seguir são descritos os detalhes de cada módulo.

5.2.1 Modelo

A camada de modelo é responsável por manter as estruturas de dados que representam as principais entidades da aplicação. O modelo da ISB Designer define cinco entidades básicas: as sequências, as dimensões, os elos, os painéis, e os objetos.

As **sequências** no modelo interno da ferramenta são estruturas de dados que representam o conceito de sequências, discutido no Capítulo 3. Assim, as sequências definem a parte linear de uma narrativa. Nesse modelo, uma narrativa pode ser formada por mais de uma sequência, que podem ser dispostas de forma linear ou não-linear. Quando as sequências são dispostas de forma linear, só existe uma ordem ou alternativa de prosseguir na história. Caso contrário, quando as sequências são dispostas de forma não-linear, uma narrativa define caminhos diferentes para prosseguir na história.

Toda sequência é formada por uma ou mais **dimensões**. A dimensão está diretamente relacionada ao conceito de distribuição da narrativa: em que parte uma aplicação pode ser distribuída por diferentes meios de exibição. Assim, pode-se dizer que cada dimensão está associada a um dispositivo de exibição. As dimensões são formadas por um conjunto de painéis dispostos sequencialmente.

Os **elos** são responsáveis pela navegação entre as sequências. Os elos podem ser de três tipos: sequencial, interativo e adaptativo. Os elos do tipo sequencial ligam duas sequências de forma natural, sem nenhuma condição associada a ele. Normalmente são utilizados para definir o caminho natural da narrativa. Os elos do tipo interativo têm uma condição de interação associada a eles. Assim, na execução da aplicação final, por exemplo, a narrativa só seguirá por aquele caminho caso o telespectador interaja com o botão do controle remoto especificado pelo elo. Os elos adaptativos têm uma condição de avaliação associada a eles.

Os **painéis** representam um *snapshot* da tela. São formados por um conjunto de objetos e contêm informações como: tempo de início e fim, e anotações realizadas pelos usuários. Cada **objeto** está relacionado com o objeto de mídia que vai fazer parte da apresentação.

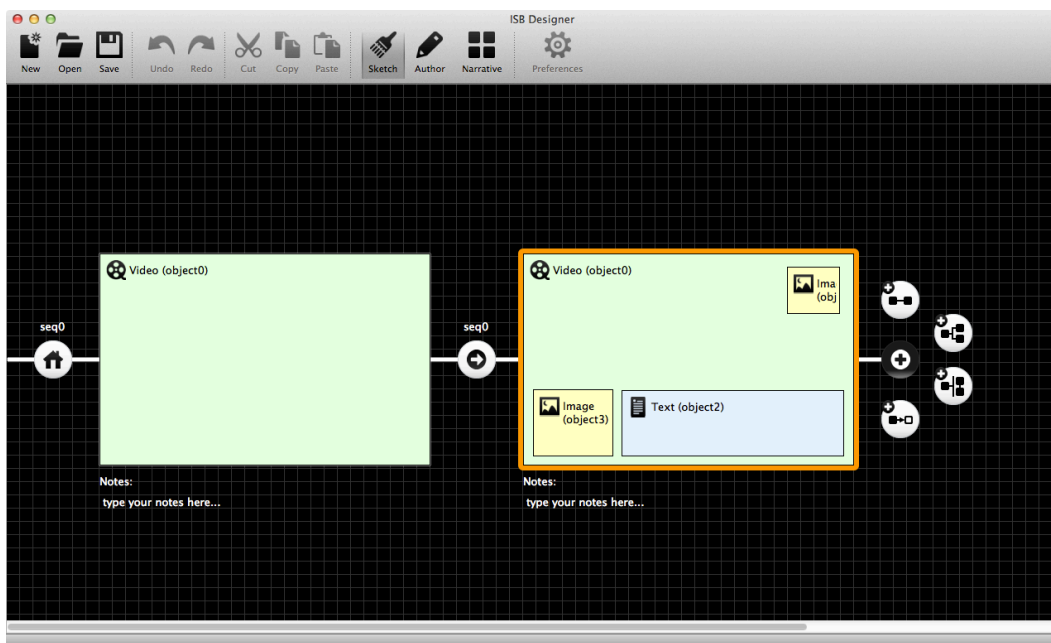
Esses elementos e sua organização definem basicamente o modelo interno da ISB Designer. Este módulo é responsável também por realizar a tradução do modelo interno para outros modelos ou linguagens. Na versão atual, somente o conversor para linguagem NCL foi implementado.

O código NCL gerado a partir do conversor NCL ainda mantém uma organização bem próxima do modelo interno da ferramenta. Isso faz com que o código NCL seja completamente desestruturado.

5.2.2 Visões

Cada visão foi implementada em um submódulo diferente e integrada à ferramenta através das interfaces da API da visão⁴. Muitos dos detalhes do funcionamento das visões discutidos nos Capítulos 3 e 4 foram mantidos. Então, somente os detalhes que ainda não foram discutidos serão apresentados nesta seção.

A Figura 5.2 e 5.3 apresenta a visão de rascunho. Na visão de rascunho as regiões onde será apresentado cada objeto de mídia podem ser definidas sem muitos detalhes (Figura 5.2) ou com um rascunho do objeto que será apresentado (Figura 5.3).



⁴ Detalhes sobre a API de Visão disponível em <http://www.telemidia.puc-rio.br/~edcaraujo/isbdesigner/>.

Figura 5.2 – Visão de rascunho da ISB Designer

A especificação de um rascunho para uma região pode ser realizada através de um *menu* contextual ou simplesmente pelo *drag-n-drop* do rascunho para dentro da região. Outro detalhe, é que o identificador da sequência fica visível o tempo todo, para que o usuário se localize entre as sequências.

A Figura 5.3 mostra um exemplo de distribuição da narrativa. A sequência da Figura 5.3 é composta por duas dimensões, um relacionada ao conteúdo que será exibido na TV e a outra relacionada ao conteúdo que será exibido em um telefone celular.

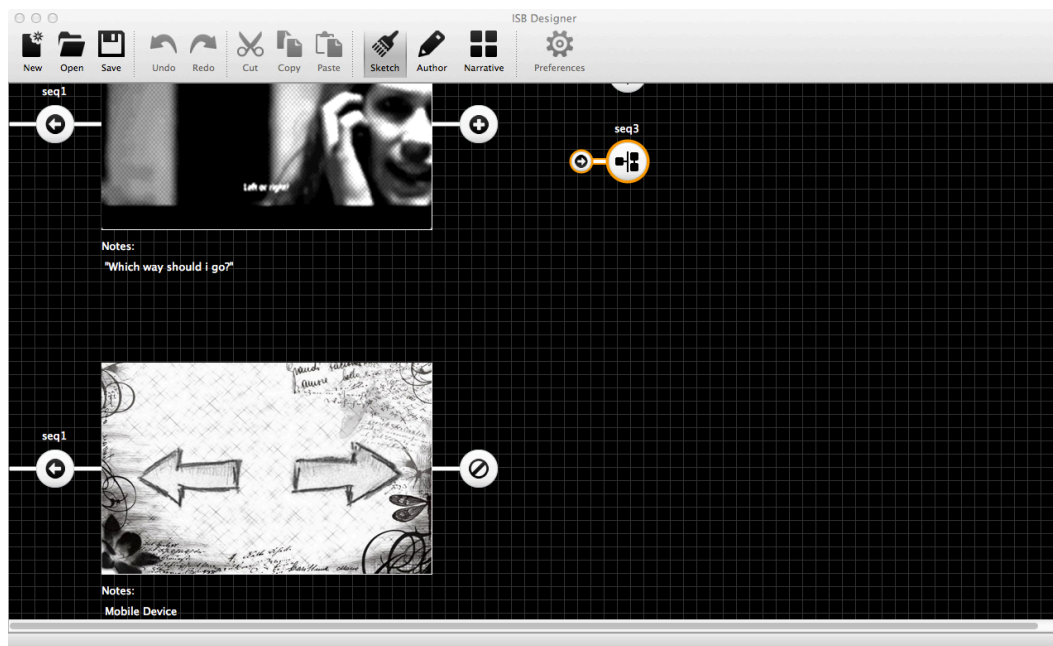


Figura 5.3 – Visão de rascunho da ISB Designer na edição de uma sequencia distribuída

A Figura 5.4 apresenta a visão de autoria. Da mesma forma como acontece na visão de rascunho, na visão de autoria o conteúdo final pode ser especificado através do *drag-n-drop* desse conteúdo para dentro das regiões.

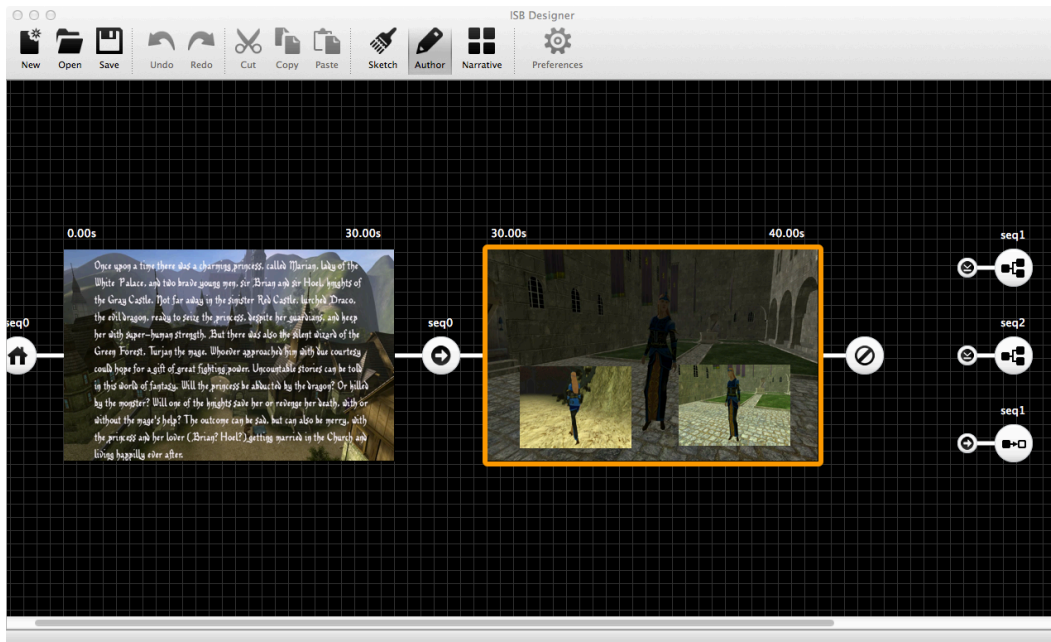


Figura 5.4 - Visão de autoria da ISB Designer

A Figura 5.5 apresenta a visão de narrativa, em que é possível observar a estrutura da narrativa em termos de sequências. Na versão atual, a visão apresenta a estrutura da narrativa como uma árvore. Quando ocorre uma referência, um ícone diferenciado é utilizado. A visão de narrativa é utilizada somente para navegação, nessa versão, as sequências não podem ser manipuladas pelo usuário.

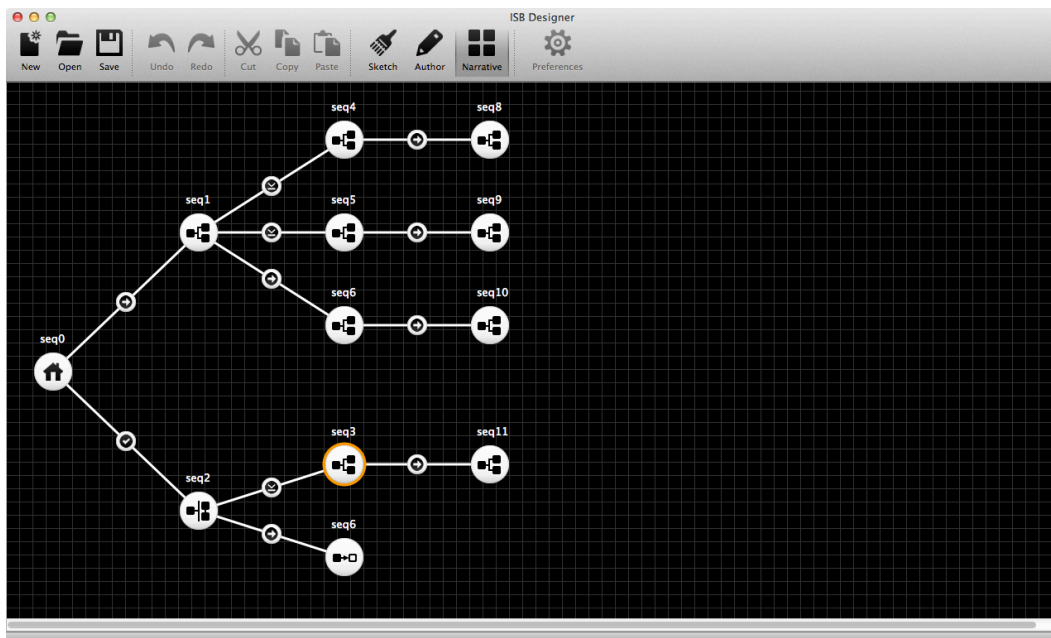


Figura 5.5 – Visão de narrativa da ISB Designer

5.2.3 Controlador

O módulo do controlador é responsável por avaliar e realizar as alterações no modelo. Através do controlador é possível ter acesso, mesmo que de forma indireta, a todas as funcionalidades da ferramenta. Isso permite que através da API do controlador⁵, outros *softwares* tenham total controle sobre a ferramenta.

5.3 Teste

Para testar a implementação e avaliar se a sincronização entre as visões estava sendo realizada de forma correta, foram realizadas duas produções⁶. A primeira é uma adaptação do filme interativo *Last Call*. É a mesma aplicação utilizada na avaliação com os usuários. Lembrando, nesse filme a personagem principal solicita, em vários pontos da história, a ajuda dos telespectadores para decidir qual caminho deveria seguir. Na versão adaptada para o teste, a aplicação contém apenas um ponto de decisão, em que telespectador tem a oportunidade de decidir por qual caminho a personagem deveria seguir através do uso de seu controle remoto.

A segunda produção foi baseada em um conto de fadas gerado a partir do Logtell (Logtell, 2011). O Logtell é um sistema de *storytelling* interativo capaz de gerar enredos automaticamente, baseado em uma especificação lógica de um gênero literário. Está aplicação contém dois pontos de decisão. No primeiro o telespectador pode decidir se a princesa deve ou não sair do castelo. No segundo, se o cavaleiro deve ou não lutar com seu inimigo. Uma característica especial dessa aplicação é que, independente da primeira decisão o telespectador, deverá decidir o destino do cavaleiro. Isso possibilitou testar a funcionalidade de reuso de sequências da ferramenta.

A duas produções foram realizadas com sucesso, permitiram concluir, que, de maneira geral, a implementação das funcionalidades mais básicas da ferramenta estão funcionando corretamente. No entanto, em uma fase de aprimoramento da segunda produção, identificou-se que especificar mudança de

⁵ Detalhes sobre a API do Controlador disponíveis em <http://www.telemedia.puc-rio.br/~edcaraujo/isbdesigner/>.

⁶ Os vídeos dessas produções estão disponíveis em <http://www.telemedia.puc-rio.br/~edcaraujo/isbdesigner/videos.html>.

foco é uma atividade muito complexa de ser realizada. Isso porque toda interação na ISB Designer resulta em uma nova sequência, e neste caso, ao contrário, deveria ocorrer dentro de um mesmo painel.