

6 Conclusão

6.1 Trabalhos Relacionados

Dentre as funcionalidades fornecidas pela interface gerada pelo framework, em destaque está a possibilidade do zoom livre. Disponibilizar esta funcionalidade com tamanha simplicidade de implementação só foi possível devido à biblioteca Piccolo. O pioneirismo da utilização desta biblioteca para exibição e navegação em objetos ocorre em outros trabalhos. O Niagara [19] é um organizador de notas pessoais, classificadas por assuntos e grupos, que se utiliza da navegação por zoom disponibilizada pelo Piccolo. A figura 16 ilustra uma tela do Niagara que exibe à esquerda a árvore hierárquica dos grupos existentes e à direita, na ZUI, os grupos e itens (lembretes).

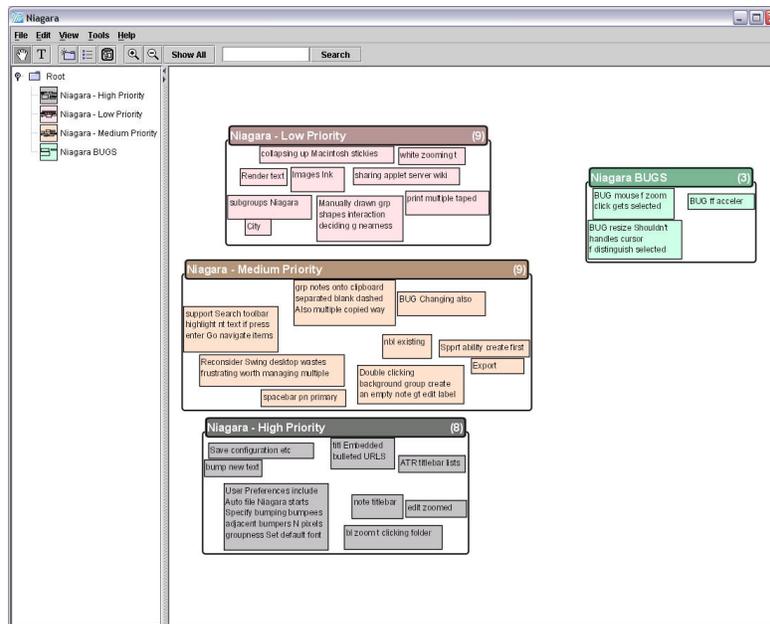


Figura 16 – Niagara - Visão de grupos de notas.

Este software é o que mais se aproxima ao resultado esperado com a utilização do framework devido à sua classificação dos conjuntos e a forma de visualização. Utilizando-se deste framework o trabalho para se construir um

software desta categoria seria reduzido apenas ao tempo de se conceber a classe de objetos que representa as notas pessoais.

Dois outros softwares utilizam a força gráfica do Piccolo. O SAPHARI (Semi-Automatic PHoto Annotation and Recognition Interface) [20] é um software que organiza automaticamente fotos em diferentes grupos, através do reconhecimento de características visuais. Por exemplo, ele divide em grupos as fotos de paisagens, de pessoas, por concentração de cores, entre outros. E a partir de uma foto pode-se chegar a fotos similares gerando uma navegação intuitiva.

O projeto PhotoMesa [21], que possui o mesmo esquema de navegação nos conjuntos, é um software mais popular. Neste existe também o conceito de manipulação direta sobre os objetos, no caso imagens, onde é possível aplicar alterações e arrasta-los. As figuras seguintes ilustram a navegação no PhotoMesa. A interface gerada pelo framework descrito nesta dissertação busca seguir o mesmo esquema desta navegação fornecida por estes.

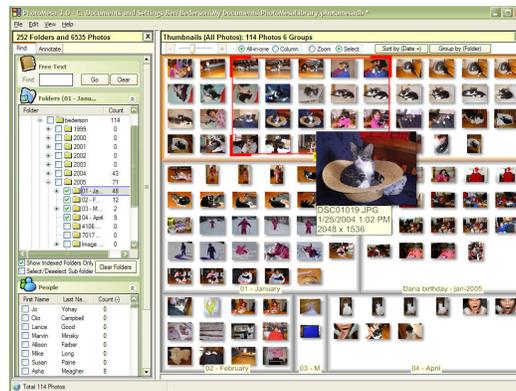


Figura 17 – PhotoMesa - Zoom Out exibindo 114 fotos em 6 grupos.

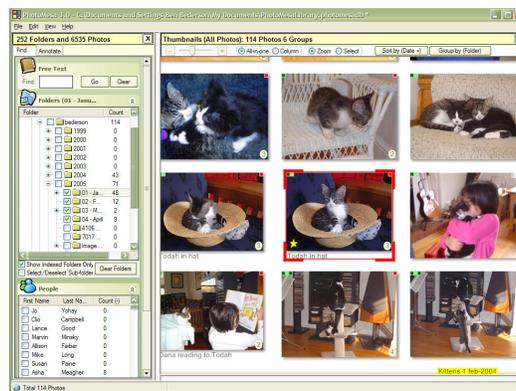


Figura 18 – PhotoMesa - Zoom In em um grupo de fotos.

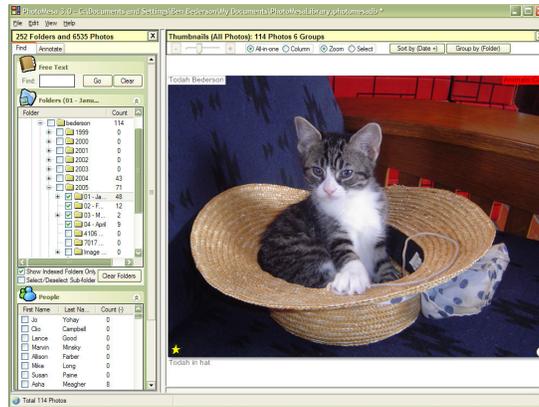


Figura 19 – PhotoMesa - Zoom In em uma única foto.

O MFTV (Multi-Faceted Tree) [22] foi criado com o objetivo de visualizar grandes quantidades de dados numa hierarquia de árvore multifacetada, utilizando-se também do zoom infinito fornecido pelo Piccolo. Além da estrutura de facetas, que pode ser similarmente exibida em objetos, conjuntos e subconjuntos, este projeto é o que melhor utiliza a funcionalidade de zoom da biblioteca Piccolo e que segue fielmente o mantra “Overview first, zoom and filter, then details on demand”. As figuras a seguir ilustram dois momentos dentro do MFTV.

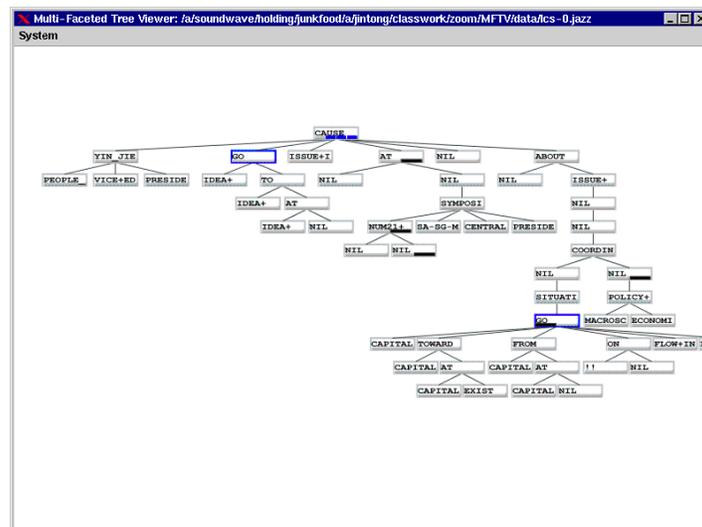


Figura 20 – MFTV - Visão geral da árvore.

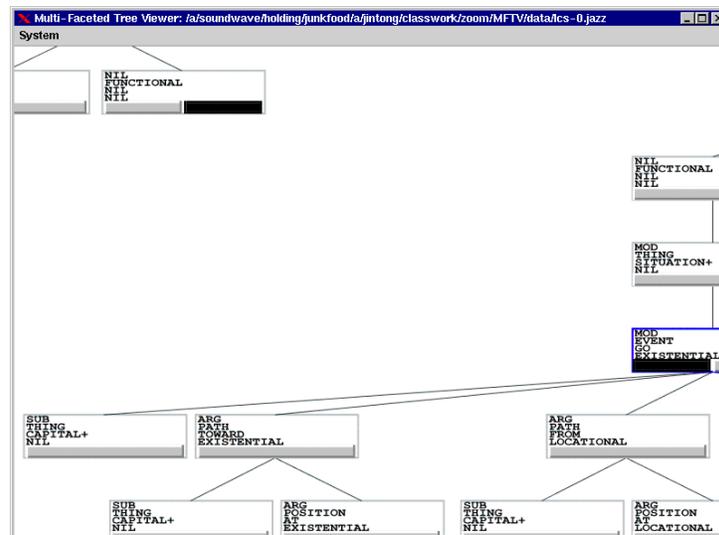


Figura 21 – MFTV - Após o zoom in mais detalhes são exibidos.

Seguindo a linha de facetas e navegação, o Flamenco [23] é uma interface de busca de informação categorizada. Ele tem como objetivo fornecer ao usuário uma navegação por grande quantidade de informações de forma ordenada, permitindo o refinamento ou expansão de uma consulta. Assim como o Flamenco, este projeto tem como grande finalidade possibilitar a navegação e categorização de informação. A diferença é que o Flamenco é uma ferramenta específica para navegação. Sua preocupação é exibir a informação através de buscas e links, sempre dentro de uma linha de interações. A navegação facetada não está implícita no projeto. Não existe uma organização dinâmica dos grupos, resultante de consultas e links. Este framework pode sim, simular essa navegação através da pré-organização dos conjuntos e objetos onde é possível criar uma hierarquia facetada ou da forma que foi implementada no caso de uso do capítulo 5 onde a partir de um objeto é possível saber a quais facetas este pertence e exibir todos os outros itens desta faceta.

Alguns outros trabalhos seguem a mesma linha de desenvolvimento deste projeto e servem de exemplo para trabalhos futuros. O USIXML (User Interface eXtensible Markup Language) [24] tem como principal característica a geração de interfaces a partir de uma descrição XML. O OpenLaszlo [25] cria as mesmas interfaces também a partir de descrições XML e de Javascript, gerando interfaces ainda mais avançadas com diversos recursos de animação. Destes trabalhos, existem algumas similaridades deste projeto onde a persistência das informações

dos conjuntos e objetos é feita através de uma descrição XML, assim como a semântica de interações feitas pelo usuário. A recuperação dessa informação, a renderização da hierarquia dos conjuntos e objetos e a utilização da semântica descrita para as interações, podem ser vistas como uma geração de interface.

Essas duas plataformas vão um pouco além, possuindo descrições XML bem amplas, contendo todas as informações necessárias para gerar as interfaces como posicionamento de objetos, o conteúdo dos mesmos, interações possíveis do usuário, resposta de ações e até mesmo a descrição de algumas animações. Essas interfaces são especialmente voltadas para internet o que inspiram o desenvolvimento de módulos WEB deste framework.

6.2 Contribuições

As principais contribuições deste trabalho são:

- Especificação de uma arquitetura de implementação e desenvolvimento de um framework para criação de aplicações ZUI ainda pouco exploradas para navegação e manipulação direta em grupos e objetos.
- Arquitetura de desenvolvimento que leva em consideração tanto o usuário desenvolvedor quanto o usuário final permitindo a implementação de funcionalidades que são configuráveis por este último.
- Possibilidade de mapeamento de modelos organizacionais complexos para uma interface de representação visual que permite entender tais modelos graficamente e manipulá-los diretamente.
- Geração de uma interface de representação visual que segue o consagrado mantra da busca visual de informação que possibilita o usuário final em um primeiro momento visualizar todos os objetos em questão, alterar o ponto de visão através do zoom, filtrá-los e finalmente requisitar dados específicos.
- Funcionalidades configuráveis dentro de certo equilíbrio de possibilidades, riscos e complexidade para o usuário final. Se por um lado o usuário tem liberdade de configurar diversos aspectos do

funcionamento da aplicação, por outro, as possibilidades são restringidas pelo usuário desenvolvedor conforme o objetivo de sua aplicação.

- Modelo de implementação onde as aplicações geradas não possuem uma semântica fixa para interações. O usuário desenvolvedor pode disponibilizar para o usuário final a sua semântica, uma ou várias, para que estas sejam selecionadas.

6.3 Trabalhos Futuros

- Aprimoramento da codificação dos métodos dos grupos e itens acessados pela aplicação. Alguns métodos podem ser implementados de forma estática, não havendo a necessidade de múltiplas chamadas e aproveitando fazendo proveito de vetores variáveis da aplicação.
- Criação de uma interface visual para a alteração das configurações feitas pelo usuário final, o que nesta versão é feito através de um arquivo XML.
- Desenvolvimento do zoom semântico. Utilizando uma ferramenta que permite com tão facilidade a utilização do zoom em uma interface, mais desses recursos devem ser explorados. O zoom poderia ter um significado diferente além da sensação visual de proximidade. Por exemplo, o zoom poderia mudar o estado dos objetos, exibindo um ícone e com a aproximação, um resumo de um texto e então o texto completo. No caso de músicas o zoom poderia alterar o volume do som. Um pouco mais além seria permitir ao usuário desenvolvedor implementar a sua semântica para esta interação o que tornaria o zoom com semântica customizável.
- Desenvolvimento de filtro de transformação. Esse filtro seria como a lente de filtragem, mas permitiria ao usuário visualizar transformações sobre o objeto dinamicamente. As alterações seriam implementáveis pelo desenvolvedor e uma gama de opções seria oferecida ao usuário.
- Desenvolvimento de novas possíveis interações e operações sobre a interface. Opções de semântica para a operação de drag&drop de um grupo sobre um item e resultados vetoriais para operações em aplicadas em diversos itens como, por exemplo, concatenar textos e músicas.

- Desenvolvimento de um método de interação por colisão de objetos.
- Desenvolvimento de um modelo similar para geração de interfaces voltadas para WEB. Permitir acesso remoto a aplicação e/ou acessar e manipular objetos remotos.
- Especificação e desenvolvimento de modelos de interação sobre a interface para dispositivos que permitem diferentes tipos de interação por parte do usuário como, por exemplo, uma interface multi-toque, reconhecimento gestual entre outros.