

**UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**



**FERRAMENTA PARA GERAÇÃO DE UMA BASE DE DADOS XML UTILIZADA
NA AVALIAÇÃO DAS FERRAMENTAS DE VISUALIZAÇÃO DE INFORMAÇÕES
HIERÁRQUICAS**

Rodrigo Soares Rodrigues

Pelotas, 2006.

RODRIGO SOARES RODRIGUES

**FERRAMENTA PARA GERAÇÃO DE UMA BASE DE DADOS XML UTILIZADA
NA AVALIAÇÃO DAS FERRAMENTAS DE VISUALIZAÇÃO DE INFORMAÇÕES
HIERÁRQUICAS**

Trabalho Acadêmico apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Prof^a. Adj. Eliane da Silva Alcoforado Diniz, Msc.

Co-Orientadora: Prof^a. Carla Maria Dal Sasso Freitas, PhD.

Pelotas, 2006.

Banca examinadora:

Profa. Ana Marilza Pernas Fleischmann, M.Sc.

Prof. Leonardo Fossati Silveira, M.Sc.

Aproveito para dedicar este trabalho à
minha mãe, fonte inspiradora e motivadora
da minha vida.

Agradecimentos

Agradeço aos meus pais, pelo apoio nos momentos mais difíceis.

Agradeço a todos os professores com que tive contato durante todos esses anos, principalmente os que ainda continuam no curso, lutando por melhorias do mesmo, com o objetivo de colocar o curso no melhor caminho possível.

Agradeço à minha orientadora, prof.^a Eliane, pela confiança em mim depositada e pelo compartilhamento de suas idéias e sugestões, nas quais foram essenciais tanto na definição quanto na realização e conclusão deste trabalho.

Agradeço à prof.^a Carla, co-orientadora deste trabalho, pelo constante interesse em ajudar apesar da distância.

Agradeço aos meus colegas de curso, parceiros nos momentos bons momentos e nos ruins também.

Resumo

RODRIGUES, Rodrigo Soares. **Geração de uma base de dados XML para auxiliar na avaliação das ferramentas de visualização de informações hierárquicas.** 2006. 70f. Trabalho de Conclusão de Curso. Graduação em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Grandes volumes de dados e uma diversidade de informações acabam por criar sérias dificuldades a usuários interessados em explorar, interpretar e compreender esses dados na busca de conhecimento. Decorrente desse fato, pesquisadores da área de visualização possuem uma grande preocupação em projetar e desenvolver ferramentas que possibilitem um acesso fácil e rápido a essa informação, visando minimizar o esforço requerido na obtenção e no entendimento da mesma. Apesar de nos últimos anos terem surgido uma quantidade razoável de técnicas de visualização de informações, verifica-se uma grande dificuldade no manuseio das ferramentas que as implementam. Por isso, existe atualmente um grande esforço para avaliar essas ferramentas. Porém, os resultados obtidos não podem servir como base de comparação entre diferentes avaliações, uma vez que cada avaliador utiliza elementos distintos no processo de avaliação, tais como critérios, conjunto de tarefas e conjuntos de dados. Visando eliminar parte deste problema, propõe-se elaborar uma ferramenta que gere bases de dados em linguagem XML, servindo como uma fonte única de informação para se testar as diversas ferramentas de visualização de informações hierárquicas. Complementarmente, será apresentado um estudo sobre as principais técnicas de visualização hierárquicas existentes.

Palavras-chaves: Visualização de informações, análise de dados, XML, avaliação, hierarquias.

Abstract

RODRIGUES, Rodrigo Soares. **Geração de uma base de dados XML para auxiliar na avaliação das ferramentas de visualização de informações hierárquicas.** 2006. 70f. Trabalho de Conclusão de Curso. Graduação em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Great volumes of data and a diversity of information create a series of difficulties for users accessing these data bases, seeking to explore, interpret and understand the data in a search for knowledge and insights. Therefore, researchers in the visualization area have a great concern related to the project and development of tools and applications to make possible easy and fast accesses to information, aiming to minimize the effort to understand of the data base and acquire new information. Although in the last years a reasonable amount of techniques for information visualization have appeared, there is a great difficulty to interact and operate the applications implementing these techniques, creating a great interest in evaluating these techniques and tools. Although all the effort directed to this purpose, the results of different evaluations cannot serve as a comparison base, because each usability expert uses distinct elements in the evaluation process, such as the criteria, the task set and data base. Aiming to eliminate one of these problems, this work attempt to elaborate a tool to generate databases in the XML language, that serve as a global source of data in the testing of different tools for hierarchic information visualization. Complementarily, taxonomy of the main existing techniques for hierarchic visualization is presented.

Word-keys: Visualization of information, Data analysis, XML, hierarchy, evaluation.

Lista de Figuras

FIGURA 1 Modelo de visualização de Haber e McNabb	26
FIGURA 2 Modelo de referência de visualização	26
FIGURA 3 <i>Chernoff Faces</i>	28
FIGURA 4 Esquema da <i>Chernoff Faces</i>	28
FIGURA 5 Representação esquemática da <i>Bifocal Displays</i> : Distorção na direção X (a) e Distorção na direção Y (b)	29
FIGURA 6 Representação esquemática da <i>Perspective Wall</i>	30
FIGURA 7 A ferramenta <i>Perspective Wall</i>	30
FIGURA 8 <i>Table Lens</i>	31
FIGURA 9 Diagrama <i>top-down</i>	32
FIGURA 10 Listas delimitadas	33
FIGURA 11 Listas indentadas	33
FIGURA 12 Conjuntos aninhados	34
FIGURA 13 <i>Treemaps</i>	35
FIGURA 14 Ferramenta <i>Information Slices</i>	36
FIGURA 15 Esquema básico da técnica <i>Cone Tree</i>	37
FIGURA 16 Ferramenta <i>Cone Tree</i>	38
FIGURA 17 Ferramenta <i>SpaceTree</i>	38
FIGURA 18 Representação esquemática da <i>Hyperbolic Tree</i>	39
FIGURA 19 <i>Bifocal Tree</i>	40
FIGURA 20 Estrutura de um documento XML	42
FIGURA 21 Documento ML	44
FIGURA 22 Parte de uma DTD para o documento XML da Fig. 20	44

FIGURA 23 Documento XML	45
FIGURA 24 Visão geral da ferramenta	47
FIGURA 25 Visão geral da ferramenta, destacando as duas áreas	47
FIGURA 26 Visão geral da ferramenta destacando a Caixa de Texto	48
FIGURA 27 Inserção do nodo raiz	48
FIGURA 28 Nodo inserido na área de visualização dos dados	49
FIGURA 29 Nodo UFPEL sendo inserido como filho de UNIVERSIDADES.....	49
FIGURA 30 Localização do nodo filho inserido na árvore	50
FIGURA 31 Inserção do nodo UFRGS.....	50
FIGURA 32 Visualização do novo nodo inserido na árvore.....	51
FIGURA 33 Apresenta o nodo DINFO, inicialmente filho de UFPEL e irmão do IFM	52
FIGURA 34 Localização do nodo DINFO após ser movido para filho de IFM.....	52
FIGURA 35 Nodo IFM sendo renomeado.....	53
FIGURA 36 Nodo renomeado para INSTITUTO DE FÍSICA E MATEMÁTICA	53
FIGURA 37 Nodo MECATRÔNICA visualizado na árvore de dados.....	54
FIGURA 38 Nodo MECATRÔNICA selecionado para sua remoção	54
FIGURA 39 Botão para salvar árvore construída	55
FIGURA 40 Escolha do nome do arquivo a ser salvo em disco	55
FIGURA 41 Botão de saída da ferramenta	56
FIGURA 42 Mensagem de confirmação de saída da ferramenta	56
FIGURA 43 Estrutura do arquivo exemplo.xml gerado pela FGAXML	57
FIGURA 44 Visualização da árvore gerada utilizando a ferramenta <i>Treemaps</i> 4.1 ...	58
FIGURA 45 Visualização da base de dados gerada pela FGAXML, através da ferramenta <i>SpaceTree</i>	58

Lista de Tabelas

TABELA 1 Classificação de informações.....	20
TABELA 2 Classes de representações visuais.....	22

Lista de Abreviaturas e Siglas

1D - *Uma Dimensão*

2D - *Duas Dimensões*

3D - *Três Dimensões*

ASCII - *American Standard Code)*

DTD - Document Type Definitions

FGAXML - Ferramenta Geradora de Árvores no Formato XML

InfoVis - *Information Visualization*

n-D - *n-Dimensões*

VisCA - *Information Visualization Conceptual Framework, Techniques and Applications*

W3C - *World Wide Web Consortium*

XML - *Extensible Markup Language*

XSD - *XML Schema Definition*

Sumário

1 Introdução	12
1.1 Motivação e Justificativa	15
1.2 Objetivos	16
1.3 Organização do Trabalho	17
2 Visualização de Informações	18
2.1 Classificação de Informações	18
2.2 Representações Visuais.....	21
2.3 Mecanismos de Interação	23
2.4 Modelo de Referência de Visualização	25
2.5 Ferramentas de Visualização de Informações	27
2.5.1 Visualização de Características e Valores Diversos	27
2.5.2 Visualização de Estruturas e Relacionamentos	28
3 Visualização de Estruturas Hierárquicas	32
3.1 Técnicas Baseadas em Preenchimento de Área	34
3.1.1 <i>Treemaps</i>	34
3.1.2 <i>Information Slices</i>	36
3.2 Técnicas Baseadas em Diagramas de Nós e Arestas	36
3.2.1 <i>Cone Tree</i>	37
3.2.2 <i>SpaceTree</i>	38
3.2.3 <i>Hyperbolic Tree</i>	38
3.2.4 <i>Bifocal Tree</i>	38
4 Ferramenta Geradora de Árvores no Formato XML	41
4.1 XML	41
4.1.1 Estrutura de um documento XML.....	42
4.1.2 Característica da linguagem XML	43
4.1.2.1 Metalinguagem DTD	43
4.1.2.2 Metalinguagem XML <i>Schema</i>	45

4.2 Descrição Operacional da F GAXML	46
4.3 Apresentação da Base de Dados no Formato XML	57
5 Conclusão.....	60

Referências

Anexos

1 Introdução

A visualização de informações permite representar dados através de ambientes gráficos, de modo que o usuário possa obter, através de sua percepção visual, uma melhor análise e entendimento das informações. A visualização de informações utiliza técnicas que combinam conceitos de computação gráfica, interfaces homem-computador e mineração de dados (FREITAS et al., 2001).

A escolha de qual técnica deve ser utilizada na construção de um sistema de visualização é uma das principais considerações a serem feitas. Essa escolha depende do tipo da informação desejada, das tarefas a serem realizadas pelo usuário e da maneira como a informação será disponibilizada. Devido à diversidade de técnicas existentes e de diferentes ferramentas que implementam uma mesma técnica, há uma dificuldade na escolha da melhor ferramenta entre as disponíveis. Estudar parâmetros para escolher a melhor ferramenta para um cenário (melhor adaptada à base de dados, às tarefas e ao perfil do usuário) serviu como motivação para este trabalho.

Nesse trabalho serão descritas ferramentas de visualização de informações que representaram avanços expressivos na área de visualização, um estudo das mesmas possibilitará que as suas características sejam apresentadas e as suas principais limitações sejam destacadas. Porém, antes que as ferramentas sejam disponibilizadas para o usuário final, se faz necessário uma avaliação, onde os especialistas na área poderão encontrar os problemas no uso dessas ferramentas e, assim, poderão sugerir alterações para torná-las mais adequadas às necessidades dos usuários. Todavia, esses especialistas utilizam conjuntos de dados distintos, o que dificulta a comparação entre as ferramentas ou entre as tarefas realizadas entre diferentes ferramentas de uma mesma técnica. Este problema foi um dos agentes

motivadores para a geração de uma base de dados contendo informações que possam ser utilizadas no processo de avaliação. Para a obtenção dessa base de dados será construída uma ferramenta, que possibilitará a criação de varias base de dados com tamanhos variados.

1.1 Motivação e Justificativa

Devido ao avanço tecnológico do hardware, o qual permitiu um aumento significativo de poder de processamento e armazenamento de dados, e ao crescimento e popularização da *World Wide Web* - WWW, que proporcionou o surgimento de novas tecnologias de comunicação, pode-se afirmar que nunca antes na História houve um volume de dados e uma quantidade de informação disponível tão grande como agora (KEIM, 2002).

Com o acesso facilitado a essa quantidade de informação, explorar e analisar esse grande volume de dados acaba tornando-se uma tarefa árdua, principalmente, quando há a necessidade de filtrar informações irrelevantes para os objetivos da tarefa que está sendo realizada (KEIM, 2002). Uma abordagem usada em sistemas de recuperação de informações, para contornar as dificuldades de selecionar informações relevantes dentre os resultados das buscas em grandes bases de dados é utilizar técnicas de visualização de informações, através das quais os usuários obtêm uma representação visual que, se por um lado abstrai detalhes de um conjunto de dados, por outro propicia uma organização desse conjunto, de acordo com os seus próprios critérios (FREITAS et al., 2001).

Ferramentas de visualização de informações são baseadas em uma representação visual e mecanismos de interação que auxiliam o usuário na melhor compreensão dos dados representados. As ferramentas podem utilizar representações visuais 1D, 2D ou 3D, não sendo necessariamente igual à dimensão do domínio da informação (SHNEIDERMAN, 1996).

Ainda que haja uma grande variedade de modelos e ferramentas para visualização de informações, cada aplicação requer um estudo particular, a fim de determinar se a ferramenta selecionada é adequada às necessidades do usuário e é utilizável pelo seu perfil (conhecimento em informática, habilidades motoras e capacidades cognitivas) (SHNEIDERMAN, 1996). Geralmente, este estudo é dirigido pelo tipo de informação que deve ser representado e pelo processo de análise do

usuário, que deve ser auxiliado pela ferramenta. Após outros trabalhos, como por exemplo, o projeto de pesquisa *Information Visualization Conceptual Framework, Techniques and Applications - VisCA*, classificarem os tipos de dados e as técnicas de visualização de informações, o desafio agora é o desenvolvimento de uma ferramenta capaz de gerar uma base de dados comum, que possa ser utilizada na avaliação de diferentes ferramentas de visualização. Esta base de dados é essencial para a realização de testes empíricos realizados com os usuários ou com os especialistas. Isso possibilitará que problemas oriundos da construção das ferramentas sejam detectados, fornecendo subsídios aos construtores de modo a realizar alterações antes que as mesmas sejam disponibilizadas aos usuários finais. Por outro lado, quando os testes forem realizados com os usuários finais, outros tipos de problemas poderão ser detectados, entre os quais o prazer subjetivo, que permitirá avaliar o grau de usabilidade da ferramenta em questão.

Entretanto, um dos maiores problemas encontrado pelos avaliadores é a variedade disponível e o tipo dos conjuntos de dados possíveis para a avaliação, fazendo com que cada avaliador crie o seu próprio conjunto de dados, eliminando qualquer parâmetro formal na avaliação. Este agravante foi um agente motivador para a elaboração de uma ferramenta, a qual se propõe a construir bases de dados de tamanhos variados, representadas através da *eXtensible Markup Language - XML* possibilitando, ao menos no que se refere ao tipo de dados utilizados, haja uniformização em um dos importantes parâmetros do processo de avaliação de ferramentas de visualização. Essa uniformização permitirá aos avaliadores se concentrarem na elaboração do seu método de avaliação, em vez de se preocuparem com a geração das árvores que compõe suas bases de dados.

1.2 Objetivos

O principal objetivo deste trabalho é a elaboração de uma ferramenta que gere árvores de dados apoiada pela linguagem XML. Criando dessa maneira, bases de dados comuns, necessárias no processo de avaliação das ferramentas de visualização de informações. Isso minimizará no que se refere à base de dados utilizada, distorções dos resultados obtidos no processo de avaliação.

1.3 Organização do trabalho

A exposição dos aspectos relacionados à área de visualização de informações, será abordada no capítulo 2. Onde, inicialmente, será apresentada uma classificação do tipo de informações em estudo. Também, serão apresentados modelos de referência de visualização que são aplicados às técnicas de visualização, prosseguindo com algumas definições dessas técnicas.

O capítulo 3 apresenta com maior ênfase as técnicas de visualização hierárquicas devido ao foco do escopo deste trabalho.

No capítulo 4, será apresentada a ferramenta para construção de base de dados. Onde serão descritas as características gerais da XML, assim como os motivos que levaram à sua utilização para representar as bases de dados geradas pela ferramenta desenvolvida. O modelo operacional do funcionamento da ferramenta, assim como, uma descrição da base de dados gerada pela mesma, também, serão abordados nesse capítulo.

No capítulo 5, são apresentadas as conclusões, ressaltando os resultados obtidos durante a realização deste trabalho e, finalizando, serão apresentadas as propostas de trabalhos futuros.

2 Visualização de Informações

2.1 Classificação de Informações

Informações correspondem a atributos que podem ser classificados através de diferentes critérios. A identificação das características desses atributos é de fundamental importância quando se deseja realizar a escolha de uma determinada ferramenta de visualização. Os atributos podem enquadrar a entidade dentro de uma classe; um atributo desse tipo pode ser classificado como: característica; categoria; atributo nominal ou ordinal (WARE, 2000). Freitas e Wagner (1995), na tab. 1 apresentou alguns critérios que podem ser utilizados na caracterização dos atributos, tais como:

- a) a classe de informação em que eles estão contidos – categoria representa toda a informação que possui uma característica isolada e independente de qualquer função. Pode-se citar como exemplos, o peso de indivíduos de uma determinada população ou a cor de seus olhos ou ainda a fauna que predomina em uma região qualquer. Um dado de natureza escalar apresenta um conjunto de valores amostrados de uma relação definida no domínio do atributo, ou seja, valores que estão envolvidos no domínio de uma determinada entidade, como por exemplo, a temperatura e a pressão de um gás em um recipiente. Um dado do tipo vetorial pode representar grandezas vetoriais, ou seja, grandezas que apresentam uma orientação espacial (direção e sentido). Como exemplos, a representação de uma grandeza física associada a um fluido, ou o deslocamento entre dois pontos onde há a necessidade de especificar a direção e o sentido desse deslocamento. Atributos do tipo tensorial podem ser citados como a deformação em torno de um

ponto e são empregados em áreas de aplicações de dinâmica de fluidos computacionais. Informações do tipo relacionamento representam ligações entre duas ou mais entidades, como é o caso de um link em uma página na *web*, ou a ligação de nodos filhos com seus respectivos pais na estrutura de uma árvore;

- b) tipo de dado, no sentido clássico de tipo primitivo - os tipos de valores estão relacionados ao conceito de tipo de dado que ocorre nas linguagens de programação. Ao se declarar o tipo de uma variável, por exemplo, é delimitado o conjunto de valores que ela pode assumir e as operações que pode ser efetuada sobre ela. Dados podem assumir valores alfanuméricos, numéricos ou podem ser compostos em valores simbólicos. Dados numéricos podem ser representados em uma numeração inteira (negativa, zero ou positiva) ou dentro do conjunto dos números Reais (números normalmente representados por uma parte inteira e uma parte fracionária);
- c) a natureza do domínio - indica se o dado pode estar definido em um domínio discreto, contínuo ou contínuo-discretizado. Num domínio discreto os dados ficam limitados a um conjunto finito de elementos, como por exemplo, marcas de automóveis de uma determinada montadora. No domínio contínuo pode ser definido por infinitos pontos em um mapa, por exemplo, quando uma forma geográfica de relevo está sob estudo, a sua altura é definida por um conjunto de dados representados por pontos em um domínio contínuo. Uma amostragem no tempo (domínio contínuo), realizada em um período regular (diário, mensal ou anual), é um exemplo de um domínio contínuo-discretizado;
- d) a dimensão do domínio - determina a dimensão do espaço onde o objeto está definido. O objeto pode estar definido no espaço unidimensional - 1D, bidimensional - 2D, tridimensional - 3D ou n-dimensional – n-D. Exemplo de domínio 1D seria alguma característica de uma entidade observada ao longo de um período de tempo. Domínios 2D e 3D representam valores observados para uma determinada forma de relevo tanto em um plano como no espaço 3D,

respectivamente. Quanto a um exemplo de dados que são representados em um domínio n-D tem-se o de aplicações que geram informações multivaloradas, como dados populacionais, onde cada indivíduo da população possuiria informações aninhadas.

Tabela 1 - Classificação de informações.

Critério	Classe	Exemplo
Classe de informação	Categoria Escalar Vetorial Tensorial Relacionamento	Gênero Temperatura Grandezas físicas associadas à dinâmica de fluidos Link num hiperdocumento
Tipo de valores	Alfanumérico Numérico (inteiro, real) Simbólico	Gênero Temperatura Link num hiperdocumento
Natureza do domínio	Discreto Contínuo Contínuo-discretizado	Marcas de automóveis Superfície de um terreno Anos (tempo discretizado)
Dimensão do domínio	1D 2D 3D n-D	Fenômeno ocorrendo no tempo Superfície de um terreno Volumes de dados médicos Dados de uma população

Fonte – FREITAS; WAGNER, 1995.

Em Luzzardi (2001), pode-se observar uma outra classificação para os dados de acordo com a estrutura representada e de acordo com o domínio.

Quanto à estrutura pode-se considerar que as informações e os dados encontram-se representados e organizados em uma das seguintes formas:

- a) listas e tabelas - estruturas que permitem representar um conjunto de dados de forma a preservar a relação de ordem linear ou total entre eles. Como por exemplos, dados multidimensionais, relacionais e estatísticos;
- b) árvores - são estruturas de dados que caracterizam uma relação entre os dados que a compõem. A relação existente entre os dados (nodos) é uma relação de hierarquia ou de composição, onde os dados são hierarquicamente subordinados entre si. Como exemplo tem-se a estrutura de um diretório de arquivos, árvores genealógicas, diagramas organizacionais, manuais e catálogos de bibliotecas;
- c) grafos - são representações de relações mais gerais entre os elementos dos dados. Pode ser representado como um conjunto de nodos

conectados por linhas (arestas). Exemplos de grafos: estruturas de páginas na Internet, redes de computadores, sistemas de tempo real (redes de Petri), diagramas de fluxo de dados.

Quanto ao domínio em que as informações estão definidas, elas podem:

- a) estar associadas a posições no espaço, por exemplo bidimensional ou tridimensional;
- b) ser temporais, ou seja, relacionadas com uma variação de tempo;
- c) estar associadas a entidades;
- d) estar associadas aos relacionamentos entre entidades.

2.2 Representações Visuais

Representações visuais correspondem aos elementos visuais de figuras ou a imagens compostas por estes elementos, que representam graficamente um conjunto de dados e auxiliam o usuário a perceber as características desses dados. São objetos visuais necessários nas tarefas de exploração e análise das informações (GERSHON, 1997).

A utilização de metáforas visuais aumenta o processo de percepção do usuário ao adaptar em uma nova situação processos cognitivos já empregados em outras visualizações. Metáfora visual utiliza uma representação visual e mecanismos de interação para manipular a imagem. O uso de objetos visuais caracteriza a aplicação de uma metáfora visual, como parte da composição de uma técnica de visualização (GERSHON, 1997).

Algumas ferramentas utilizam-se de metáforas visuais para a representação dos dados, tais como, a *Perspective Wall* (MACKNLAY; ROBERTSON; CARD, 1991), assim como, o *File System Navigator* da *Silicon Graphics* (TESLER; HEILMEIER, 1992). As representações visuais podem ser classificadas de acordo com a tab. 2, onde alguns exemplos de sua utilização são apresentados.

As representações podem ser unidimensionais, bidimensionais ou tridimensionais, conforme a dimensão do espaço onde os objetos geométricos estejam aplicados. Representações unidimensionais são pouco comuns, mas podem ser utilizadas, por exemplo, como uma linha para descrever o tempo (LUZZARDI, 2001).

Tabela 2 - Classes de representações visuais

Classe	Tipo	Utilização
Gráficos 2D, 3D	De pontos Circulares De linhas De barras De superfície (para 3D)	Representação da distribuição dos elementos no espaço domínio ou representação da dependência/correlação entre atributos.
Ícones Grifos Objetos geométricos	Elementos geométricos 2D ou 3D diversos	Representação de entidade num contexto ou representação de grupos de atributos de diversos tipos.
Mapas	De pseudo-cores De linhas De superfícies De ícones, símbolos diversos	Representação de campos escalares ou de categorias. Representação de linhas de contorno de regiões, isovalores Idem, no espaço 3D. Representação de grupos de atributos (categoriais, escalares, vetoriais, tensorais).
Diagramas	Nodos e arestas	Representação de relacionamentos diversos: É-um, É-parte-de, Comunicação, Sequência, Referencia, etc.

Fonte – FREITAS; WAGNER, 1995.

Além das ferramentas de visualização de informações citadas anteriormente, outras, segundo Luzzardi (2001), também utilizam outros tipos de metáforas visuais:

- a) organização de informação na forma de listas ou tabelas - na ferramenta *Chernoff Faces* (CHERNOFF, 1973), informações são codificadas para uma representação da face humana;
- b) árvores - *CityScape* (GERSHON; EICK, 1995) representa as informações como uma cidade com prédios e ruas;
- c) listas, tabelas, árvores e grafos - *Fisheye* (FURNAS, 1986, 1999) exhibe as informações aplicando uma distorção na imagem, no que se convencionou chamar de visão foco+contexto.

2.3 Mecanismos de Interação

A utilização de uma representação visual estática, nem sempre é suficiente para que o usuário possa entender e compreender grandes volumes de dados. Para facilitar esse entendimento, são disponibilizadas funções que permitem a exploração desses, através da utilização de operações em níveis diferentes de abstração. Estas operações (ações) alteram a representação visual, de modo que novos aspectos do conjunto de dados possam ser observados. Assim, a representação pode ser alterada para mostrar apenas a região de interesse selecionada pelo usuário. Quando a região de interesse do usuário ocupa o campo de visão principal e o restante do conjunto de dados é mantido em uma área separada, chama-se de visão geral+detalhe. No entanto, quando a região de interesse é exibida em conjunto com o restante dos dados, utiliza-se a denominação foco+contexto (FREITAS; WAGNER, 1995).

Existe alguma confusão em relação ao entendimento sobre tarefas de usuários e mecanismos de interação. Sendo, pois, necessário que se tenha uma clara visão desses conceitos. Entende-se por tarefas de usuários as operações de alto nível de abstração que os usuários podem realizar sobre uma representação visual (manipulação da informação, de símbolos e de objetos). Os mecanismos de interação são operações de baixo nível de abstração que permitem aos usuários realizarem as suas tarefas diretamente sobre uma representação visual, por exemplo, o deslocamento horizontal/vertical de uma *scroll bar* em uma tela (LUZZARDI, 2001).

Freitas e Wagner (1995) identificaram sete tarefas básicas que deveriam ser suportadas por uma ferramenta de exploração visual:

- a) navegação - exploração do espaço onde os objetos que representam a informação estão contidos;
- b) seleção - de vários objetos ou de parte de um objeto;
- c) consulta - recuperação de informações em atributos de um objeto;
- d) preparação - modificação de atributos e inicialização de parâmetros;
- e) processamento - simulação, cálculos, monitoramento, controle e registro da evolução;
- f) anotação - comentários e texto;
- g) armazenamento - de imagens da visualização.

Shneiderman (1996) prevê sete tarefas de alto nível para uma aplicação de visualização de informações:

- a) visão geral - todos os conjuntos de dados devem estar visíveis para o usuário de modo que ele possa obter uma visão geral do universo que contém esses dados. Os objetos apresentados devem possuir um número limitado de parâmetros e de relacionamento para que o usuário não fique perdido, devido às limitações impostas pelos equipamentos gráficos e pelo sistema perceptível humano. Os atributos gráficos mais comuns que compõe os objetos visualmente são as cores, a localização, tipos de símbolos e tamanho dos símbolos. A percepção humana certamente limita a representação, pois existem ainda questões sobre a quantidade de informação que o usuário pode perceber e cognitivamente processar. A obtenção dessa visão geral do conjunto de dados é benéfica para o usuário, pois ajuda na localização e diminui o tempo de pesquisa;
- b) zooming - realiza a combinação de filtragem com um aumento no nível de detalhes dos objetos. Esta tarefa geralmente é utilizada pelos usuários quando há a necessidade de se obter um detalhamento maior dos dados ou para focalizar um determinado conjunto de dados;
- c) filtragem - permite a redução da quantidade de elementos do conjunto de dados. A redução é obtida através da eliminação de itens de acordo com seus atributos;
- d) visão geral+detalhe - o usuário pode visualizar detalhes de um ou mais dados específicos e manter a visão geral do espaço em que esses e todos os outros dados estão contidos. Visão geral+detalhe mantém a visualização geral dos dados enquanto esse detalhamento é apresentado ao usuário em uma janela auxiliar, sem prejudicar a orientação do usuário e a exploração dos dados. As formas de uma apresentação deste tipo podem ser mostradas uma de cada vez (*time multiplexing*) ou podem ser expostas ao mesmo tempo em partes diferentes da tela (*space multiplexing*) (CARD; MACKINLAY; SHNEIDERMAN, 1999);

- e) relações - necessidade do usuário em conhecer as relações de determinada informação que está sendo focalizada. Essa informação pode conter vínculos com outras informações, eventos ou atributos;
- f) histórico - suporte para desfazer ou repetir operações;
- g) extração - permite ao usuário extrair sub-coleções de informações e parâmetros de consulta que resultam em uma informação desejada, armazenando os resultados fora da ferramenta.

Tanto as tarefas definidas por Freitas e Wagner (1995) como as tarefas de usuário definidas por Shneiderman (1996) são tarefas abstratas e gerais e pode-se notar uma similaridade entre elas.

Tendo em vista que as técnicas de visualização de informações são destinadas primordialmente a apoiar a exploração do conjunto de informações representadas visualmente, é essencial utilizar os mecanismos de interação (LUZZARDI, 2001). Estas técnicas permitem ao usuário:

- a) selecionar novo foco de interesse sem perder o contexto;
- b) filtrar informações desnecessárias;
- c) agrupar informações relacionadas;
- d) realizar operações de *zooming* sobre determinadas informações;
- e) realizar diversas consultas buscando uma informação em particular (recuperação de informações).

2.4 Modelo de Referência de Visualização

Da necessidade de identificar os componentes essenciais na composição de uma técnica de visualização ou no desenvolvimento de uma nova, é necessária a utilização de um modelo de referência de visualização. Um *pipeline* simplificado foi sugerido por Haber e McNabb (1990) (Fig. 1). Primeiramente, os dados sofrem uma filtragem, destinada a reduzir o conjunto de dados a ser exibido. Em seguida, é realizado um mapeamento para alguma representação geométrica. E finalmente, a imagem é gerada (*rendering*).



Figura 1 - Modelo de visualização de Haber e McNabb.
Fonte: HABBER; McNABB, 1990.

Uma visualização é descrita por Card, Mackinlay e Shneiderman (1999) como uma seqüência de mapeamentos ajustáveis dos dados para uma representação visual, de modo a possibilitar a interação do usuário com o espaço de informação. Essa interação sucessiva é chamada por Card, Mackinlay e Shneiderman (1999) de “cristalização do conhecimento”. A Fig. 2 expressa a seqüência de mapeamentos do modelo de visualização de Card, Mackinlay e Shneiderman (1999).

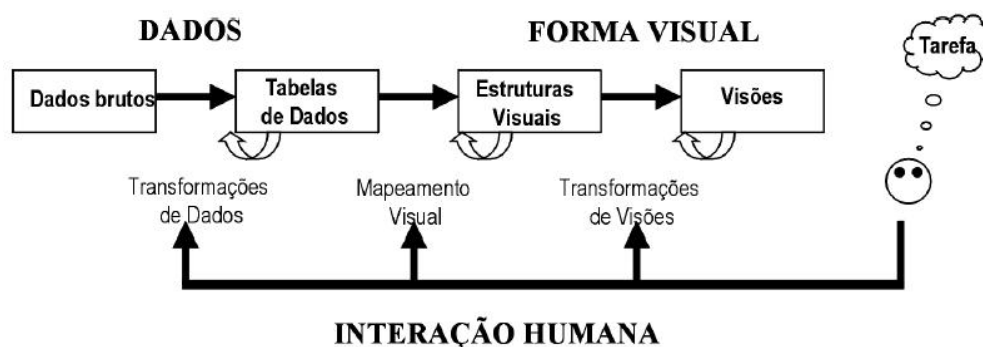


Figura 2 - Modelo de referência de visualização.
Fonte: CARD; MACKINLAY; SHNEIDERMAN, 1999.

Como pode ser visto na Fig. 2, os dados brutos são transformados em tabelas, após terem sido coletados ou gerados por algum processo externo. O uso de tabelas é uma simplificação, pois os dados podem ser representados em qualquer tipo de estruturas de dados, dependendo exclusivamente da aplicação e do conjunto de dados do usuário. Novas transformações podem ser aplicadas além das transformações iniciais, tanto para acrescentar novos dados ao conjunto inicial ou na reorganização do conjunto de dados.

As tabelas de dados, que contêm as relações presentes no conjunto de dados brutos, são mapeadas para estruturas visuais, representando os significados das informações e dos dados por meio de objetos gráficos. Finalmente, estas estruturas visuais são agrupadas e exibidas em imagens ou visões. As operações e

interações do usuário sobre essas representações são chamadas de transformações visuais.

2.5 Ferramentas de Visualização de Informações

As ferramentas de visualização de informações utilizam técnicas que possibilitam o uso de representações ou metáforas visuais para exibir graficamente dados, os quais normalmente não possuem uma representação gráfica que seja direta, óbvia e natural. Com o propósito de melhorar a representação do conjunto de informações, os desenvolvedores de ferramentas de visualização de informações fazem uso das técnicas que procuram utilizar diferentes objetos geométricos para auxiliar na percepção visual dos usuários (LUZZARDI, 2001).

Diversos autores (SHNEIDERMAN, 1996; CARD; MACKINLAY; SHNEIDERMAN, 1999; CHI; RIEDL, 1998) procuraram classificar as técnicas de acordo com: à visualização de características e valores, ou quanto à visualização de estruturas e relacionamentos. Este último, sendo subdividido em estruturas hierárquicas e grafos e estruturas lineares.

2.5.1 Visualização de Características e Valores Diversos

A visualização de características e valores diversos é exibida em gráficos ou mapas dos mais variados tipos, desde os tradicionais gráficos de pontos e linhas até conjuntos de ícones ou símbolos. Ícones são utilizados para identificar uma entidade ou elemento que fazem parte de um contexto. Quando um objeto geométrico tem a forma e outros atributos visuais determinados pelos valores dos atributos da entidade, é conveniente utilizar a denominação glifo (*glyphs*) para descrever esse objeto (CHERNOFF, 1973). Um exemplo da utilização de glifos é o conjunto de faces de Chernoff (CHERNOFF, 1973). Como pode ser observado na Fig. 3, cada região da face representa um atributo e variações nessas regiões indicam as diferenças entre os dados visualizados.

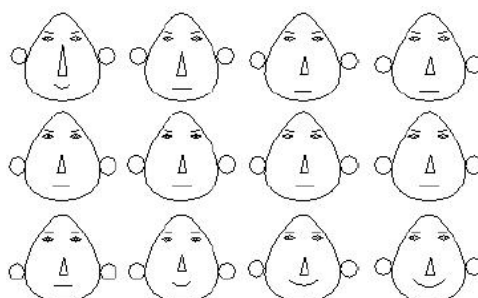


Figura 3 - *Chernoff Faces*.
Fonte: LUZZARDI, 2001.

A Fig. 4 apresenta uma representação esquemática da *Chernoff Faces*, onde é possível observar as regiões pertencentes a cada face que sofrem alterações para representar diferentes tipos de dados.



Figura 4 - Esquema da *Chernoff Faces*.

Fonte: Adaptada de CHERNOFF, 1973.

2.5.2 Visualização de Estruturas e Relacionamentos

A Visualização de estruturas e relacionamento é subdividida em estruturas hierárquicas e grafos e estruturas lineares. Devido à grande quantidade de informação organizada de forma hierárquica (por exemplo, em sistemas de arquivos e sites da *web*), muitas das técnicas de visualização de informações têm sido desenvolvidas para auxiliar na visualização e interação de informações organizadas nessa forma. Decorrente desse fato e devido à relevância desse tipo de organização

para este trabalho, haverá uma ênfase maior sobre as técnicas de visualização hierárquicas no capítulo seguinte.

As estruturas lineares são capazes de descrever uma organização através da enumeração de dados para melhor visualização da informação, podendo, por exemplo, representar relações em um hiperdocumento. As ferramentas que serão apresentadas a seguir utilizam esse tipo de estrutura para apresentar os itens e objetos que representam à informação (documentos, figuras, gráficos, etc.). Devido a grande diversidade de técnicas que estão inseridas nessa classificação, foram selecionadas algumas técnicas, pois devido as suas características, elas representam a grande maioria dessa categoria.

A técnica Bifocal Display, de Spence e Apperley (1982), apresenta a informação em três áreas distintas no espaço de visualização. A sua área central é destinada para os dados que estão no foco (em destaque) da imagem e as outras informações de contexto geral são apresentadas à direita e à esquerda da região detalhada. A área central possui um tamanho maior em relação às áreas laterais e, por isso, é exibida de forma distorcida tanto na direção X (Fig. 5a) quanto na direção Y (Fig. 5b).

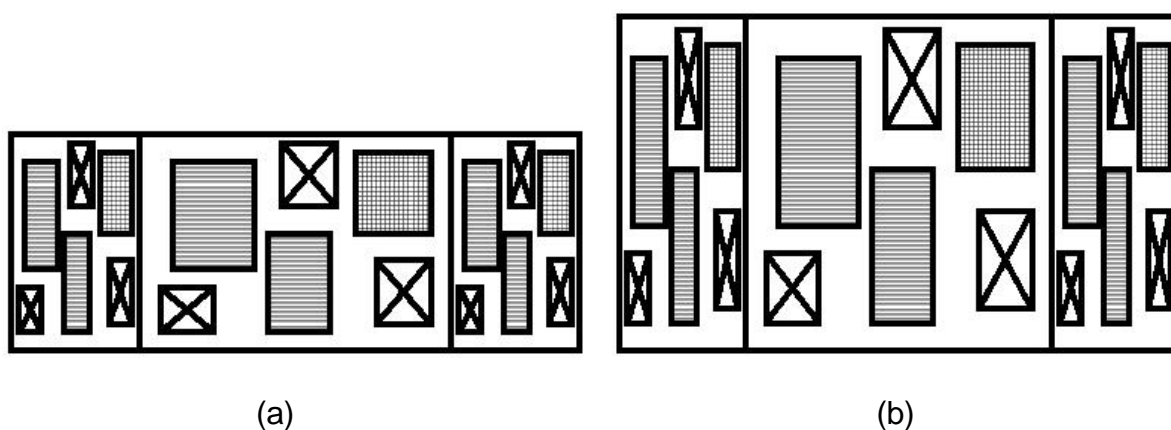


Figura 5 – Representação esquemática da *Bifocal Displays*: Distorção na direção X (a) e Distorção na direção Y (b).

Fonte: Adaptada de SPENCE; APPERLEY, 1982.

O mesmo conceito foi utilizado de forma diferente na construção da ferramenta de visualização *Perspective Wall* (MACKINLAY; ROBERTSON; CARD, 1991), onde o campo visual da informação é composto por uma “parede”, também posicionando a informação contextual nas suas laterais, utilizando linearmente os

espaços para organizar a estrutura visual da informação. Diferente da técnica *Bifocal Display*, vista anteriormente, que utiliza exclusivamente o layout 2D, na *Perspective Wall* a informação é gerada em um layout 3D, fornecendo assim uma utilização eficiente do espaço e a possibilidade de aplicar uma transição suave entre os elementos da visualização (Fig. 6 e Fig. 7). O uso dessa técnica permite que contexto mantenha-se visível, possibilitando a localização das informações, sendo essa uma importante vantagem apresentada pela técnica.

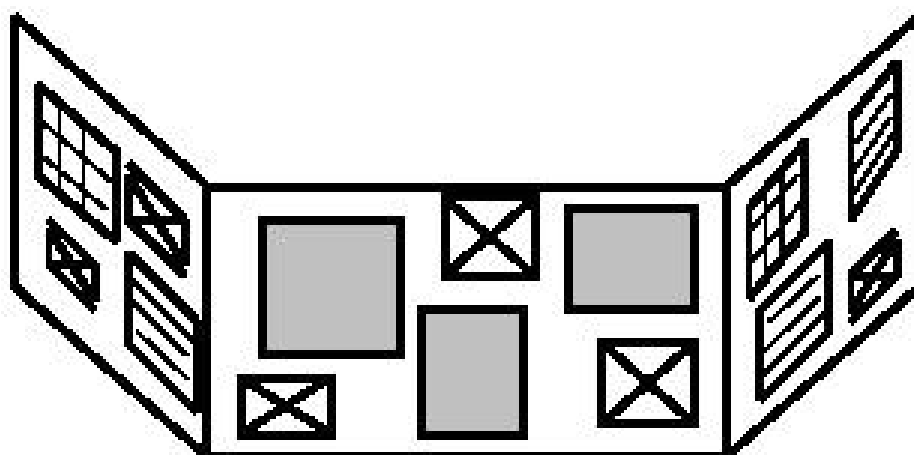


Figura 6 - Representação esquemática da *Perspective Wall*.
Fonte: MACKINLAY; ROBERTSON; CARD, 1991.



Figura 7 – A ferramenta *Perspective Wall*.
Fonte: MACKINLAY; ROBERTSON; CARD, 1991.

A *Table Lens* (RAO; CARD, 1994) (Fig. 8) é uma técnica de visualização que utiliza o mecanismo foco+contexto na manipulação e visualização de grandes tabelas. Através da distorção da visão nas direções, de forma independente, X e Y, consegue-se obter uma visualização do conjunto de dados completo e também

interagir com itens específicos, de acordo com o nível de interesse do usuário. A ferramenta apresenta operações de *zoom* (alteração da área de foco), *adjust* (mudança do conteúdo de visualização sem alterar o tamanho da área focal), *slide* (mudança na posição da área focal dentro do contexto) e *adjust+zoom* (aumenta o número de células, sem alterar o tamanho original das mesmas, na área focal).

	Year	Quarter	Product	Channel	Region	Salesperson	Units	Revenue	Profits
128	1993	2	ForeCode Pro	Direct Sales	Southwest	Kevin Polen	1029	139898	171581
444	1993	4	ForeCode Pro	VAR	West	Tom Tuttle	302	122310	51371
445	1993	4	ForeCode Pro	VAR	West	Ann Thomas	302	122310	51371
446	1993	3	ForeMost B...	Direct Sales	Midwest	Bal Vistone	301	2.9595e+006	929338
447	1993	3	ForeMost B...	VAR	South	Gary Copper	301	2.708e+006	948150

Figura 8 - *Table Lens*.

Fonte: RAO; CARD, 1994.

3 Visualização de Estruturas Hierárquicas

Estruturas de informações organizadas em hierarquias, na forma de árvores genealógicas e organogramas, geralmente são representados por diagramas compostos por nodos e arestas. Vários *layouts* têm sido utilizados para representar árvores. Alguns dos mais conhecidos são: o diagrama *top-down*, a lista delimitada, a lista indentada e os conjuntos aninhados (semelhantes ao diagrama de Venn usado para representar conjuntos) (CAVA, 2002). Um *layout* do tipo *top-down*, que pode ser observado na Fig. 9, fornece uma solução efetiva. Porém, a relação entre altura e profundidade da árvore pode trazer sérias dificuldades na visualização da estrutura hierárquica, pois quanto maior a profundidade da árvore, maior será a dificuldade de visualização do nível mais distante da raiz. Ou seja, enquanto a profundidade cresce de maneira linear, o espaço exigido para exibição do nível mais baixo cresce de forma exponencial.

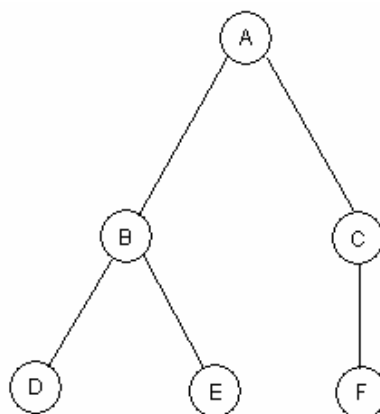


Figura 9 - Diagrama *top-down*.
Fonte: VELOSO et al., 1986.

As listas delimitadas (exemplo na Fig. 10) mostram todos os caminhos entre a raiz e cada um dos nodos listados, separados através de parênteses, indicando claramente os relacionamentos pai/filho existente entre todos os nodos. Esta forma

de representação de árvore faz com que o usuário tenha que percorrer toda a árvore a fim de conseguir uma construção mental de toda a hierarquia, tornando inviável essa aplicação quando a hierarquia em questão é muito grande, aplicando uma sobrecarga cognitiva no usuário (CAVA, 2002).

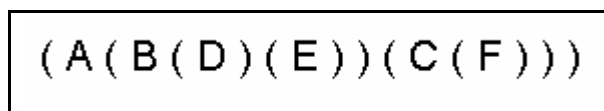


Figura 10 - Listas delimitadas.
Fonte: VELOSO et al., 1986.

Listas indentadas (Fig. 11) exibem uma lista de filhos para cada nodo, onde a hierarquia pai/filho(s) é representada através da variação no tamanho de linhas horizontais. Mesmo que essa representação resolva o problema de exibição da topologia da árvore encontrado nas listas delimitadas, ela introduz sérias dificuldades quando as árvores crescem em tamanho, pois acarreta em uma quantidade de linhas paralelas ao longo da margem esquerda. Em árvores muito grandes, poderia gerar uma confusão visual ao usuário, perdendo a visão geral e não compreendendo as relações pai/filhos.

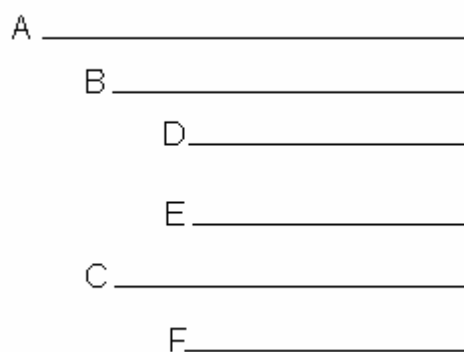


Figura 11 - Listas indentadas.
Fonte: VELOSO et al., 1986.

Em conjuntos aninhados, a árvore é exibida na forma de agrupamentos de conjuntos. Os nodos são representados como um subconjunto pertencente ao conjunto do seu respectivo pai, como mostra a Fig. 12. A grande vantagem do uso dessa representação está no aproveitamento máximo do espaço da tela. Contudo,

quanto maior o tamanho da estrutura hierárquica, mais difícil é a visualização dos conjuntos e subconjuntos na tela, devido às limitações da área de amostragem, que restringe a visualização da topologia da árvore e as conexões entre os nodos.

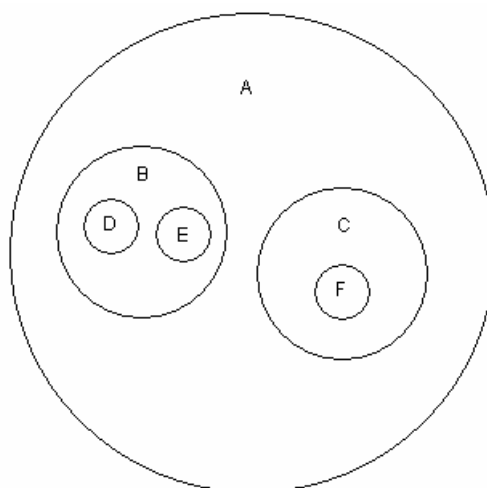


Figura 12 - Conjuntos aninhados.
Fonte: VELOSO et al., 1986.

A partir dessas representações clássicas, foram desenvolvidas técnicas de visualização com o objetivo de auxiliar os usuários a melhor compreender as informações dispostas hierarquicamente. Essas técnicas estão divididas em duas classes: as que utilizam uma abordagem do preenchimento da área e as que utilizam diagramas de nodos e arestas.

3.1 Técnicas Baseadas em Preenchimento de Área

Nessa classe o espaço disponível para a representação da estrutura é subdividido recursivamente, de forma que cada subárea represente um nodo da hierarquia. A seguir serão apresentadas algumas ferramentas que utilizam técnicas pertencentes a esse grupo.

3.1.1 *Treemaps*

É uma técnica proposta por Shneiderman (1992), com o objetivo de visualizar os tamanhos de arquivos de estruturas de diretórios para observar a utilização do espaço de armazenamento em disco. Utiliza o preenchimento do espaço visual bidimensional, com a divisão da área retangular da amostragem em sucessivos retângulos para representar os nodos de maneira distinta, como pode ser

visto na Fig. 13, que apresenta a ferramenta desenvolvida pelo mesmo criador da técnica, também chamada de *Treemaps*. O tamanho da área ocupada por cada nó é proporcional a qualquer um de seus atributos, como por exemplo, o seu tamanho.

Em casos especiais, a área de um determinado nó pode não ser exibida se o seu peso for muito pequeno ou se houver grande densidade de nodos representados. Várias divisões na área retangular da visualização podem gerar um retângulo com uma área menor que um *pixel*, resultando em uma oclusão visual ou perda de visibilidade.

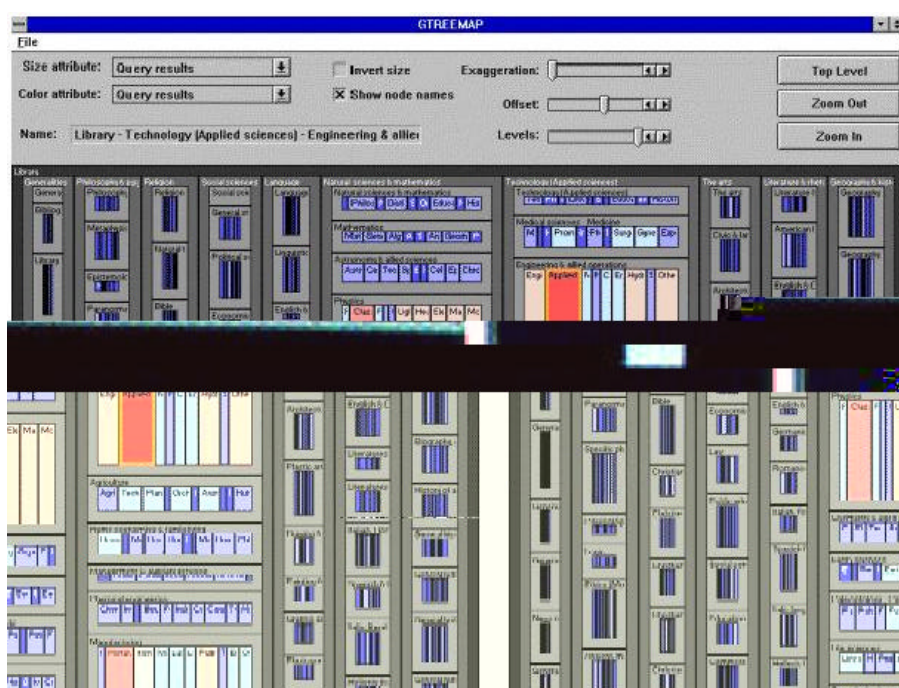


Figura 13 - *Treemaps*.

Fonte: SHNEIDERMAN, 1992.

Algumas facilidades encontradas na ferramenta *Treemaps*, tais como: a modificação do mapeamento dos atributos para o tamanho e cor dos retângulos; a realização de uma filtragem de nodos, ocultando os nodos indesejáveis; a expansão de um nível de cada vez ou de um nodo em particular; um mecanismo de *zoom* utilizado para ampliar um determinado nodo, dentre outras, permitem ao usuário destacar os dados de seu interesse.

3.1.2 *Information Slices*

A técnica de visualização *Information Slices* (ANDREWS; HEIDEGGER, 1998) foi desenvolvida com o propósito de permitir a visualização de estruturas de diretórios e arquivos. Ela utiliza um ou mais discos semicirculares para representar a hierarquia dos arquivos. Cada fatia desses discos representa um nodo (diretório ou arquivos), com o espaço de cada fatia sendo diretamente proporcional ao tamanho em *bytes* ocupado. A Fig. 14 apresenta uma ferramenta de visualização de arquivos utilizando a técnica *Information Slices*.

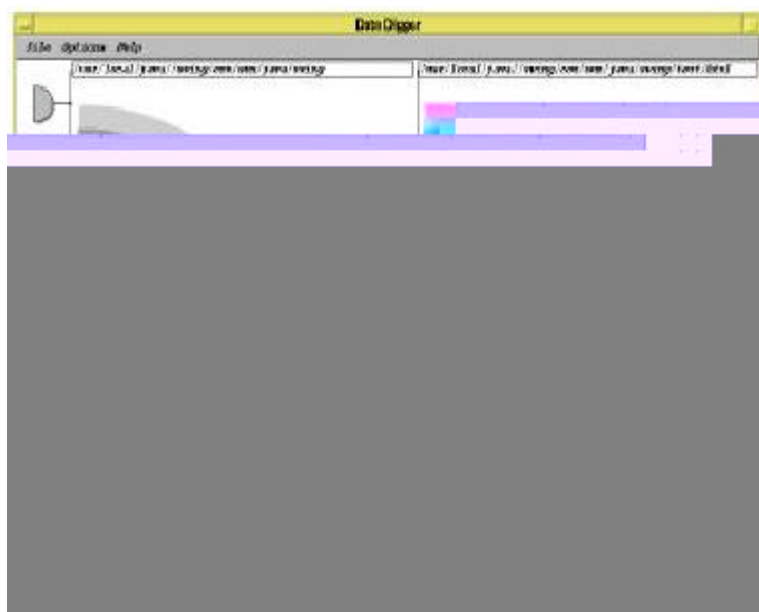


Figura 14 – Ferramenta *Information Slices*.
Fonte: ANDREWS; HEIDEGGER, 1998.

A navegação é realizada por meio de um mecanismo de seleção, que permite que um nodo no disco da esquerda tenha seus detalhes exibidos no disco da direita, com seus respectivos filhos distribuídos em uma fatia de 180 graus. Cores diferenciam nodos quanto ao seu tipo. Os atributos dos nodos selecionados são exibidos em uma barra de status auxiliar localizada na parte inferior da tela.

3.2 Técnicas Baseadas em Diagramas de Nodos e Arestas

Nessa abordagem os nodos que compõem a estrutura são representados por figuras geométricas e seus relacionamentos por linhas. Algumas dessas técnicas têm usufruído do espaço tridimensional para representar estruturas hierárquicas,

fundamentadas na idéia de que a terceira dimensão fornece um espaço adicional para visualização de grandes estruturas. Decorrente desse fato pode ocorrer perda de informações relevantes com a sobreposição de elementos. Com isso, novos mecanismos de interação devem ser disponibilizados para que os usuários possam explorar as informações, tais como: alterar interativamente a posição da câmera (ponto e ângulo de observação), podendo mover-se na visualização para revelar objetos ocultos (fora da visualização) ou sobrepostos por outros objetos. A seguir, são descritas algumas técnicas e ferramentas de visualização pertencentes a essa classe.

3.2.1 Cone Tree

A técnica *Cone Tree* (ROBERTSON; CARD; MACKINLAY, 1991) utiliza um espaço tridimensional para representar as informações da árvore na forma geométrica de cones. Os nodos são representados por retângulos. A raiz da árvore está localizada no topo da estrutura de um cone tridimensional. Os filhos de cada nodo estão dispostos na base circular do cone. A Fig. 15 mostra o esquema básico da estrutura, enquanto que a Fig. 16 apresenta uma representação de uma estrutura de arquivos, utilizando sombreamento e transparência na composição visual dos cones. Recursos de rotação, animação e *zoom* tornam o acesso à informação fácil e com transição suave e controlada pelo usuário, mantendo a orientação de quem está explorando. Maiores detalhes dessa técnica poderá ser visto na referência citada nessa seção.

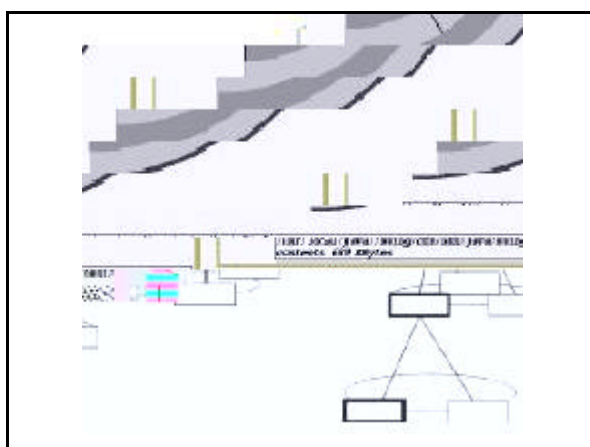


Figura 15 - Esquema básico da técnica *Cone Tree*.
Fonte: ROBERTSON; CARD; MACKINLAY, 1991.



Figura 16 - Ferramenta *Cone Tree*.
Fonte: ROBERTSON; CARD; MACKINLAY, 1991.

3.2.2 *SpaceTree*

Ferramenta, desenvolvida nos laboratórios de pesquisa da Universidade de Maryland onde também foi desenvolvida a ferramenta *Treemaps* 4.1. Conforme é apresentado na Fig. 17, a *SpaceTree* (SPACETREE, 2006) representa a hierarquia utilizando o digrama clássico de nodos e arestas, em uma orientação horizontal do *layout top-down* em um único sentido (esquerda para direita). Dependendo do tamanho da árvore de dados, os nodos que não estão selecionados possuem a sua representação ajustada, contraído a visualização dos mesmos, com o objetivo de melhorar a visualização do nodo selecionado.

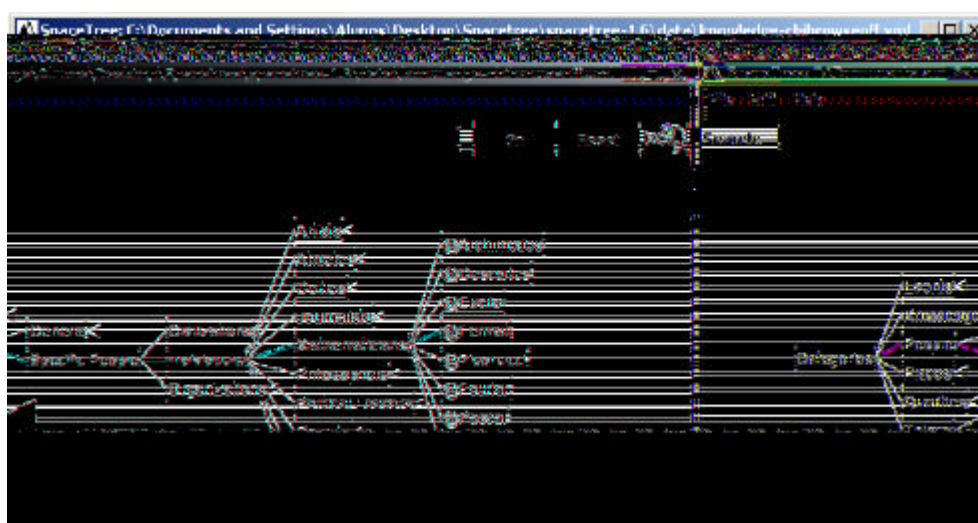


Figura 17 – Ferramenta *SpaceTree*.
Fonte: Adaptada de SPACETREE, 2006.

3.2.3 Hyperbolic Tree

A técnica *Hyperbolic Tree* (LAMPING; RAO; PIROLI, 1995) representa hierarquias através de um *layout* radial definido em um plano hiperbólico. A Fig. 18a mostra a transição de um nodo quando o mesmo é selecionado (na parte inferior direita da tela). Sendo movimentado automaticamente pela técnica para o foco da visualização (Fig. 18b e 18c). A informação que está em destaque é diferenciada das demais através do realce na intensidade de seu contorno, tanto do nodo raiz como de seus filhos.

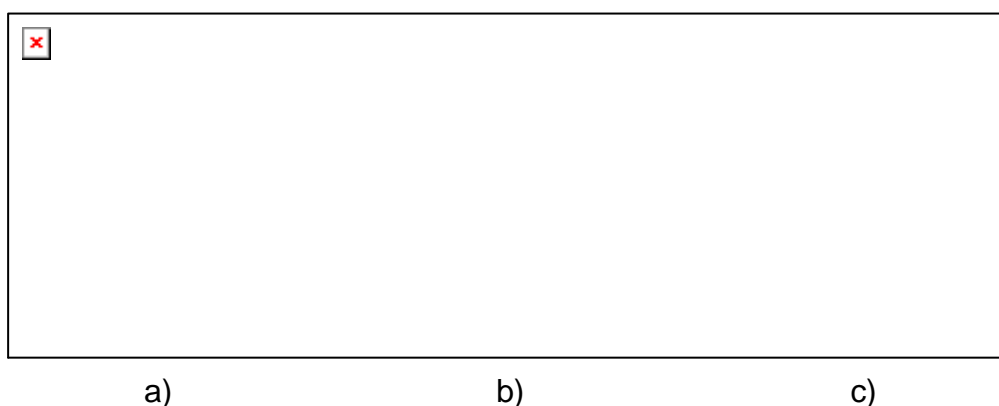


Figura 18 - Representação esquemática da *Hyperbolic Tree*.
Fonte: LAMPING; RAO; PIROLI, 1995.

3.2.4 Bifocal Tree

Na técnica *Bifocal Tree* (CAVA, 2002), a hierarquia é visualizada com o auxílio de duas áreas conexas, sendo uma para mostrar o contexto do nodo selecionado pelo usuário e a outra para mostrar os detalhes, conforme é exibido na Fig. 19. Inicialmente nenhum nodo está presente na área de detalhe. Assim que um nodo é selecionado, ele passa a ser o foco da visualização e imediatamente seus detalhes são visualizados na área de detalhes, à direita. A área de contexto, à esquerda, visualiza o nodo pai do nodo selecionado pelo usuário no centro de sua estrutura. A área de detalhes disponibiliza um espaço maior para visualizar e expandir a subárvore do nodo selecionado, mantendo a visão geral da árvore na área do contexto.

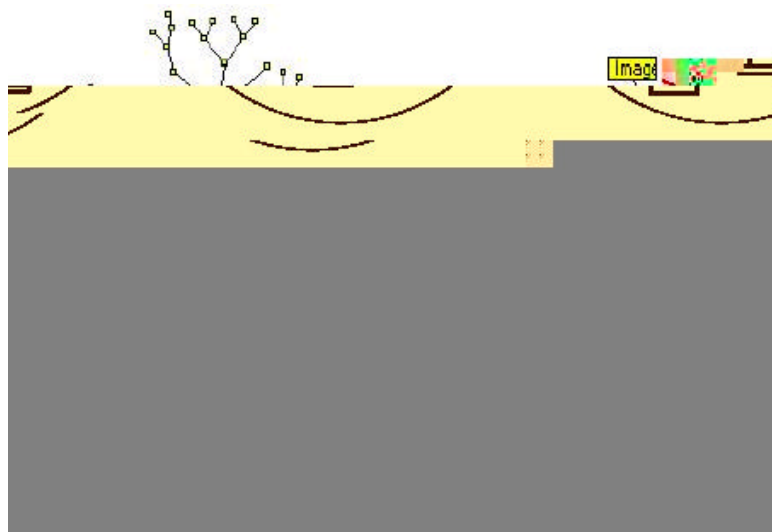


Figura 19 - *Bifocal Tree*.
Fonte: CAVA, 2002.

A apresentação de algumas ferramentas de visualização de informações, apresentadas no capítulo 2 e nesse capítulo, teve como objetivo principal ressaltar as diversidades existentes entre o universo de ferramentas disponíveis, tanto em nível comercial quanto em nível acadêmico. A vantagem de se ter uma gama considerável de ferramentas para esse propósito, se traduz na possibilidade do usuário encontrar uma que melhor se adeqüei as suas necessidades. Embora, as ferramentas de visualização de informações apresentem avanços expressivos na área de visualização, se faz necessária uma avaliação, onde os especialistas na área poderão encontrar os problemas no uso dessas ferramentas e, assim, poderão sugerir alterações para torná-las mais adequada às necessidades dos usuários. Todavia, esses especialistas utilizam conjuntos de dados distintos, o que dificulta a comparação entre as ferramentas ou entre as tarefas realizadas entre diferentes ferramentas de uma mesma técnica. Para minimizar esse problema, se faz necessário a utilização de uma base de dados, que contenha informações que possam ser utilizadas no processo de avaliação. Para a obtenção dessa base será construída uma ferramenta, que utilizará elementos que possibilitará a homogeneidade das bases utilizadas no processo de avaliação. Essa ferramenta será abordada no capítulo seguinte.

4 Ferramenta Geradora de Árvores no Formato XML

A construção da Ferramenta Geradora de Árvores no Formato XML – FGAXML, que se propõe a construir bases de dados de tamanhos variados, representadas através da linguagem XML, possibilita que haja uniformização em um dos importantes parâmetros (dados) do processo de avaliação de ferramentas de visualização. Essa uniformização evitará distorções nos resultados obtidos no processo de avaliação e permitirá aos avaliadores se concentrarem na elaboração do seu método de avaliação, em vez de se preocuparem com a geração das árvores que compõe suas bases de dados.

Antes de ser feita a descrição da FGAXML, será feita uma introdução a XML, devido ao fato, dessa linguagem ser utilizada na representação da base dados gerada por essa ferramenta.

4.1 XML

O XML é uma linguagem de marcação de dados que provê uma formatação em *tags*¹ para descrever dados estruturados. Ela foi desenvolvida para ser utilizada em marcações de documentos, possibilitando declarações mais precisas do conteúdo (ANDERSON, 2001). A linguagem XML, com todas as tecnologias agregadas, representa uma maneira distinta de descrever dados. Sendo assim, o XML tem um papel importantíssimo neste mundo atual, que tende à compatibilidade e integração entre os sistemas, já que ela é uma tecnologia que permite compartilhar informações de maneira flexível, confiável e simples de se entender (RAY, 2001).

Tal linguagem é considerada um instrumento de padronização, já que ela utiliza texto puro dentro de marcações, isto é, as *tags*, que podem ser definidas

¹ *Tags* – são campos com limitadores de início e fim

facilmente pelo usuário. Outro fator que determinou a linguagem XML como uma linguagem de padronização é o fato de possuir compatibilidade com mais de um conjunto de caracteres, o que caracteriza uma vantagem em relação às outras linguagens que trabalham apenas com padrões americanos, como, por exemplo, o *American Standard Code - ASCII*.

Outra utilização encontrada para a XML é o armazenamento de informações, já que ela possui características para estruturação de dados aninhados, sendo capaz de representar uma arquitetura no modelo de dados hierárquicos. As características de linguagem de marcação padrão e a possibilidade de armazenamento informações estruturadas foram os agentes motivadores que levaram à utilização da linguagem XML no presente trabalho.

4.1.1 Estrutura de um documento XML

Para ser capaz de conter informação, um documento precisa ser bem-definido, possuindo um tamanho finito, isso é obtido através do uso de *tags* iniciais e *tags* finais (LIGHT, 1999). Um exemplo de uma estrutura básica de um documento XML é apresentado na Fig. 20.

```
<?xml version="1.0" encoding="ISO-8859-1"?>2
<cadastro>
  <usuario>
    <nome>Rodrigo</nome>
    <email>rodrigo@xml.com.br</email>
    <idade>18</idade>
  </usuario>
  <usuario>
    <nome>César</nome>
    <email>cesar@xml.com.br</email>
    <idade>16</idade>
  </usuario>
</cadastro>
```

Figura 20 - Estrutura de um documento XML.

A estrutura de uma árvore armazenada em XML sempre deve ter um elemento principal chamado de root tag. Ele é importante para aninhar e organizar

² O documento XML utiliza a codificação ASCII. É importante ressaltar que para utilizar elementos da língua portuguesa, deve-se inserir o seguinte código: <?xml version="1.0" encoding="ISO-8859-1"?>.

as demais marcações e também por não permitir que os dados fiquem organizados livremente pelo documento, tornando-o mal formado. No exemplo anterior, <cadastro> é a root tag do documento. No resto do documento deve-se escrever *tags* de marcação definidas pelo próprio usuário, as quais são escritas de forma aninhada. No espaço delimitado pelas *tags* de início e fim armazenam-se os dados relativos a essa *tag*. Obrigatoriamente uma *tag* deverá ser fechada exatamente com o mesmo nome em que foi aberta, precedido por uma “/”. Deve-se observar também a grafia, pois as marcações XML fazem distinção entre letras maiúsculas e minúsculas. Na Fig. 20, por exemplo, <usuario> e </usuario> são *tags* do documento.

4.1.2 Características da linguagem XML

A linguagem XML provê um padrão capaz de codificar conteúdo para uma grande variedade de aplicações, desde as mais simples até as mais complexas. Um documento XML pode conter muitos tipos de informação e inclusive definir outras linguagens (metalinguagem). Por exemplo, ao escrever aplicações para *palms*, poderia ser utilizada uma linguagem para aplicações *Wireless* (sem fio), que poderia se chamar PML. Da mesma maneira, desenvolvedores de ferramentas de visualização poderiam criar linguagens próprias para representar suas bases de dados e utilizá-las no processo de avaliação das referidas ferramentas.

Existem duas metalinguagens que definem as linguagens que podem ser obtidas a partir da XML: *Document Type Definitions* - DTD e a XML Schema.

4.1.2.1 Metalinguagem DTD

A DTD é uma gramática para o documento XML e sua importância está relacionada à possibilidade do próprio usuário definir suas marcações. O exemplo apresentado na Fig. 21 representa um documento XML composto de um número de elementos “pessoa”. A seguir, a Fig. 22, demonstra uma possível representação desse documento utilizando um arquivo DTD.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<bd>
    <peessoa>
        <nome>Rodrigo</nome>
        <idade>25</idade>
        <email>rodrigo+xml.com.br</email>
    </peessoa>
    <peessoa> ... </peessoa>
</bd>

```

Figura 21 - Documento XML.

```

<DOCTYPE bd
...
    <!ELEMENT bd (peessoa*)>
    <!ELEMENT pessoa (nome, idade, email)>
    <!ELEMENT nome (#PCDATA)>
    <!ELEMENT idade (#PCDATA)>
    <!ELEMENT email (#PCDATA)>
...
>

```

Figura 22 - Parte de uma DTD para o documento XML da Fig. 20.

Na Fig. 22, A primeira linha indica que o elemento raiz é “bd”. As próximas cinco linhas são declarações de marcações, que mostram que “bd” pode conter um número arbitrário (representado pelo asterisco) de elementos “pessoa”, que contém os elementos “nome”, “idade”, “*email*”, os quais contêm apenas caracteres “data”. A expressão “pessoa*” é uma expressão regular, significando qualquer número de elementos pessoa.

A XML não requer obrigatoriamente um arquivo DTD para cada documento, muito embora a DTD defina uma gramática para os elementos e atributos do documento, o que a tornaria muito desejável por garantir documentos bem-definidos.

Um arquivo DTD pode ser definido como um dicionário, que além de ajudar na interpretação e na representação do documento XML, ajuda um analisador XML a verificar se este foi escrito de forma sintaticamente correta. Por outro lado, ao ler o documento, as metalinguagens presentes na linguagem XML permitem comprovar a integridade dos dados em qualquer momento. A validação de um documento garante que os dados sejam completos, colocados na ordem correta no documento e com os valores apropriados dos atributos (HOQUE, 2000).

4.1.2.2 Metalinguagem XML *Schema*

A XML *Schema* também é chamada de *XML Schema Definition* - XSD. Ela é uma alternativa ao DTD baseada em XML e descreve a estrutura de um documento XML. O propósito de um XML *Schema* é definir os blocos de construção permitidos em um documento XML, como um DTD.

Segundo W3C (2006), um XML *Schema* define:

- a) elementos que podem aparecer em um documento;
- b) atributos que podem aparecer em um documento;
- c) que elementos são elementos filhos;
- d) a ordem dos elementos filhos;
- e) o número de elementos filhos;
- f) se um elemento é vazio ou pode incluir texto;
- g) tipos de dados para elementos e atributos;
- h) valores padrão e fixos para elementos e atributos.

A Fig. 23 descreve a estrutura de um documento que utiliza XML *Schema*.

```
<?xml version="1.0" encoding="iso-88591"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="recado">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="de" type="xs:string"/>
        <xs:element name="para" type="xs:string"/>
        <xs:element name="mensagem" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Figura 23 - Exemplo de uma estrutura XML *Schema*.

Fonte: W3C, 2006.

Uma das grandes vantagens de XML *Schemas* é o suporte a tipos de dados. O uso dessa metalinguagem torna mais fácil: descrever conteúdo de documentos permissíveis; validar os dados; trabalhar com dados de um banco de dados; definir restrições aos dados; definir padrões/formatos de dados; converter dados entre diferentes tipos.

Acredita-se que em breve espaço de tempo a XML *Schemas* serão usadas na maioria das aplicações *web* em lugar de DTDs.

Em W3C (2006) pode-se encontrar algumas razões que justificam essa afirmação:

- a) XML *Schemas* são extensíveis para adições futuras
- b) XML *Schemas* são mais ricos e úteis que DTDs
- c) XML *Schemas* são escritos em XML
- d) XML *Schemas* suportam tipos de dados
- e) XML *Schemas* suportam *namespaces*

4.2 Descrição Operacional da FGAXML

A FGAXML utilizada para a geração das bases de dados em XML foi desenvolvida no ambiente de programação *Delphi* versão 7.0, da *Borland Software Corporation*. A ferramenta (Fig. 24) apresenta uma interface com o usuário bastante simplificada, dividida em duas áreas. A primeira (Fig. 25a) localizada à esquerda, é a área onde os dados são visualizados em uma estrutura hierárquica e aparecem na medida em que são inseridos pelo usuário. A segunda (Fig. 25b) localizada a direita, contém os botões utilizados na operação dos dados, tais como:

- a) Inserir Nodo - inserção dos novos nodos;
- b) Deletar Família de Nodos - exclusão de nodos existentes na árvore;
- c) Salvar Árvore - salvamento da estrutura representada à esquerda em um arquivo XML;
- d) Fechar/Sair - saída da ferramenta.

A caixa de texto localizada acima do botão Inserir Nodo (Fig. 26) é utilizada para o usuário determinar o nome que o nodo terá na sua representação no arquivo.



Figura 24 - Visão geral da ferramenta.

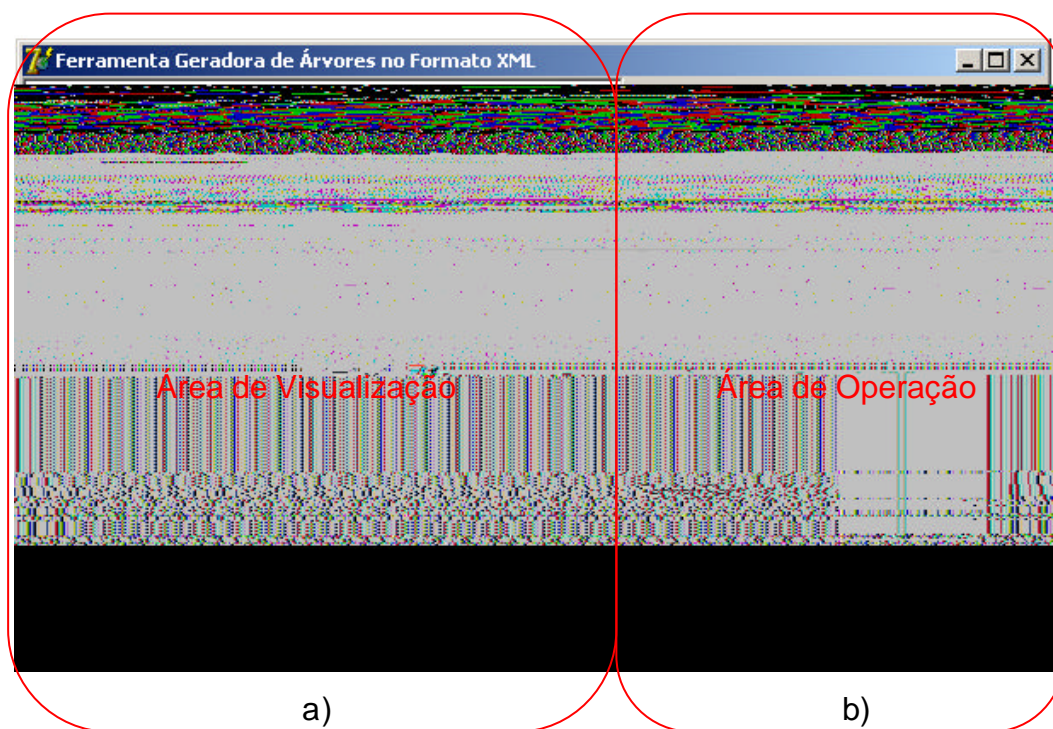


Figura 25 - Visão geral da ferramenta, destacando as duas áreas.



Figura 26 - Visão geral da ferramenta destacando a Caixa de Texto.

A Fig. 27 apresenta o nodo UNIVERSIDADES escrito na caixa de texto correspondente ao botão Inserir Nodo, tornando esse nodo raiz da árvore.

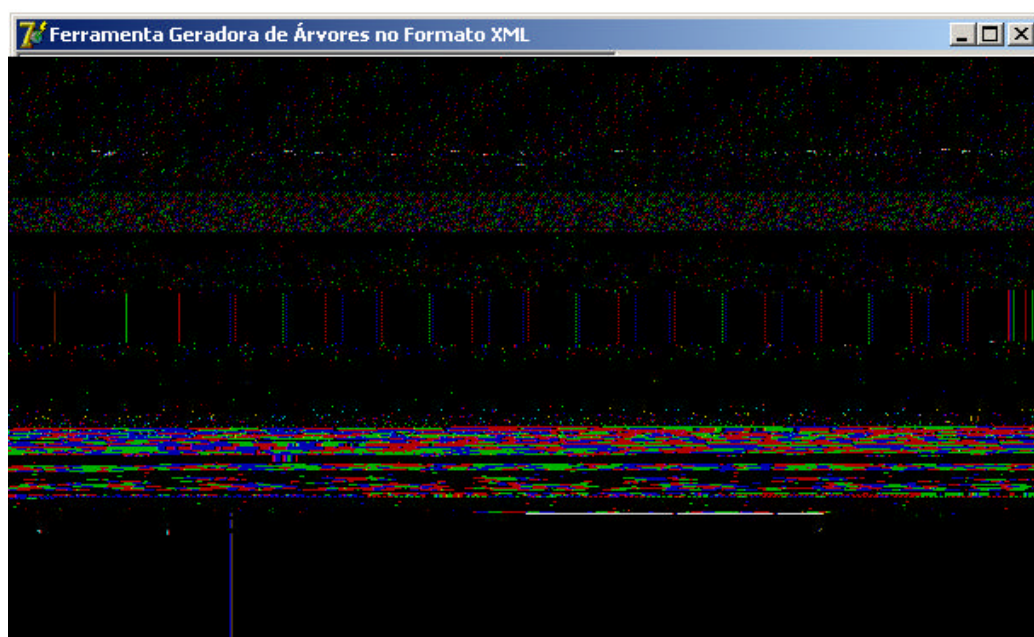


Figura 27 - Inserção do nodo raiz.

A Fig. 28 mostra o nodo escrito inserido como raiz, sendo representado visualmente na área localizada à esquerda da ferramenta.

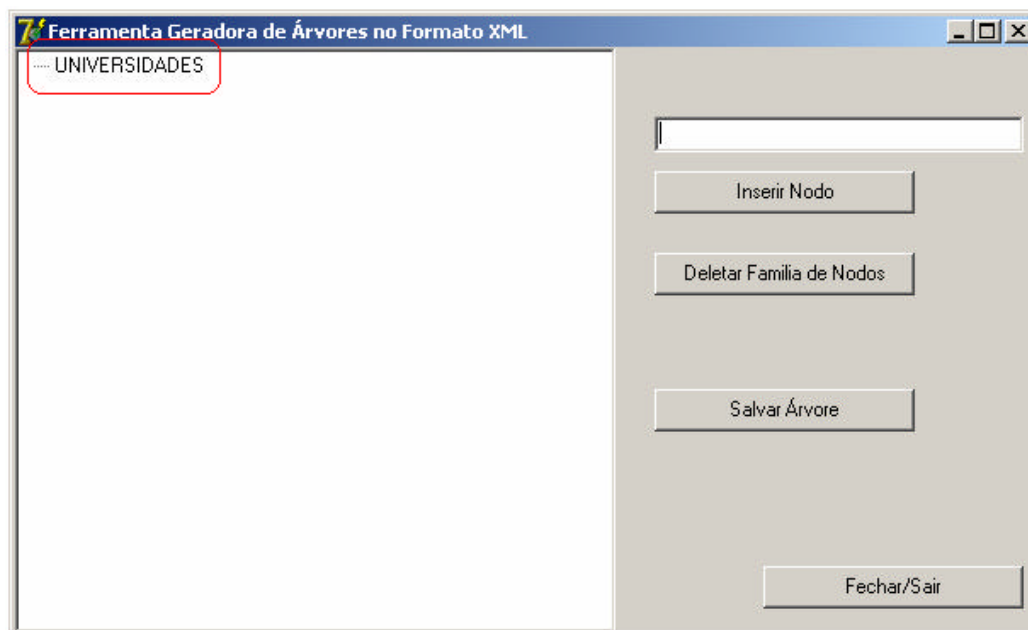


Figura 28 - Nodo inserido na área de visualização dos dados.

Para que um nodo filho seja adicionado, inicialmente o seu nodo pai deve ser selecionado na área de visualização dos dados (área localizada à esquerda da ferramenta). Na Fig. 29, o nodo raiz UNIVERSIDADES é selecionado antes de serem inseridos os seus nodos filhos, por exemplo, UFPEL, que deverá ser inserido na caixa de texto localizada na área de operação.

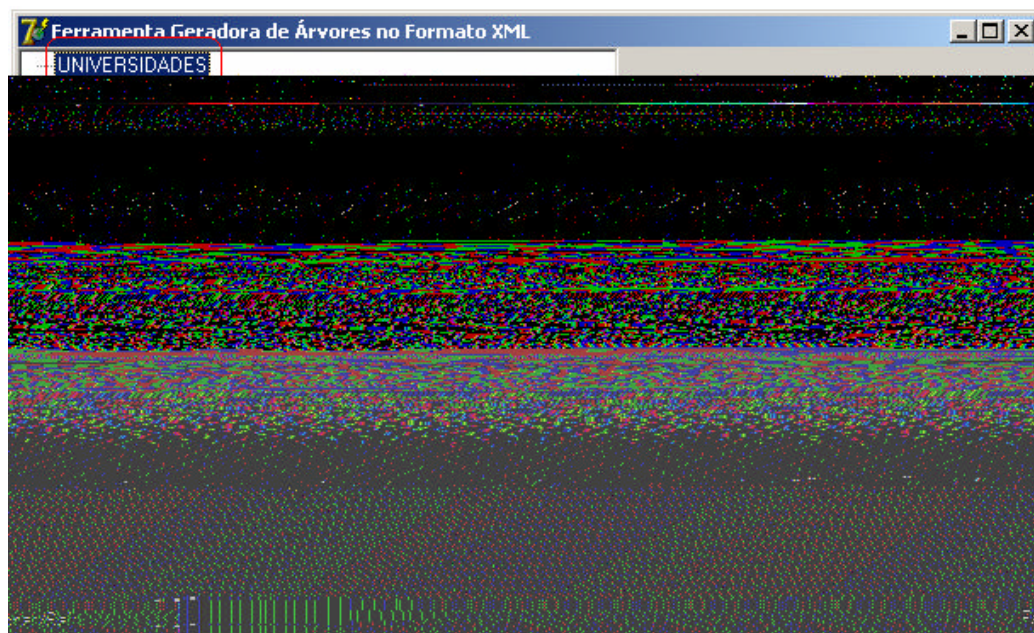


Figura 29 - Nodo UFPEL sendo inserido como filho de UNIVERSIDADES.

Depois de inserido, o novo nodo UFPEL fica abaixo do nodo UNIVERSIDADE na área de visualização, como observado na Fig. 30. Neste ponto é possível visualizar a hierarquia da árvore representada através do *layout* de listas indentadas.

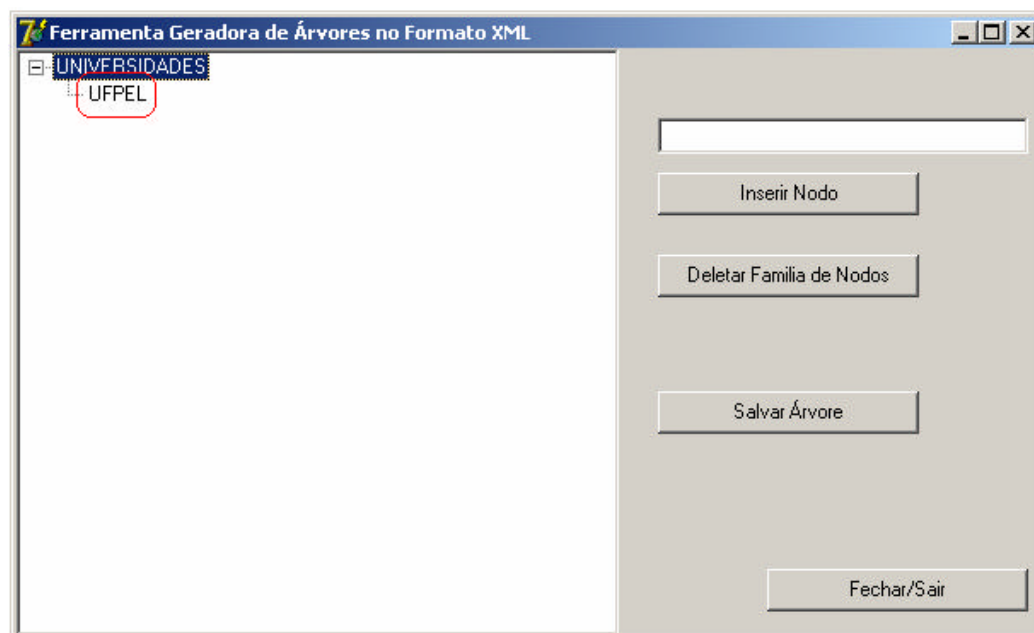


Figura 30 Localização do nodo filho inserido na árvore.

Novos nodos podem ser adicionados e logo visualizados na ferramenta como mostram as Fig. 31 e 32. Desta forma, a base de dados começa a ser montada, através da inserção desses nodos por parte do usuário.

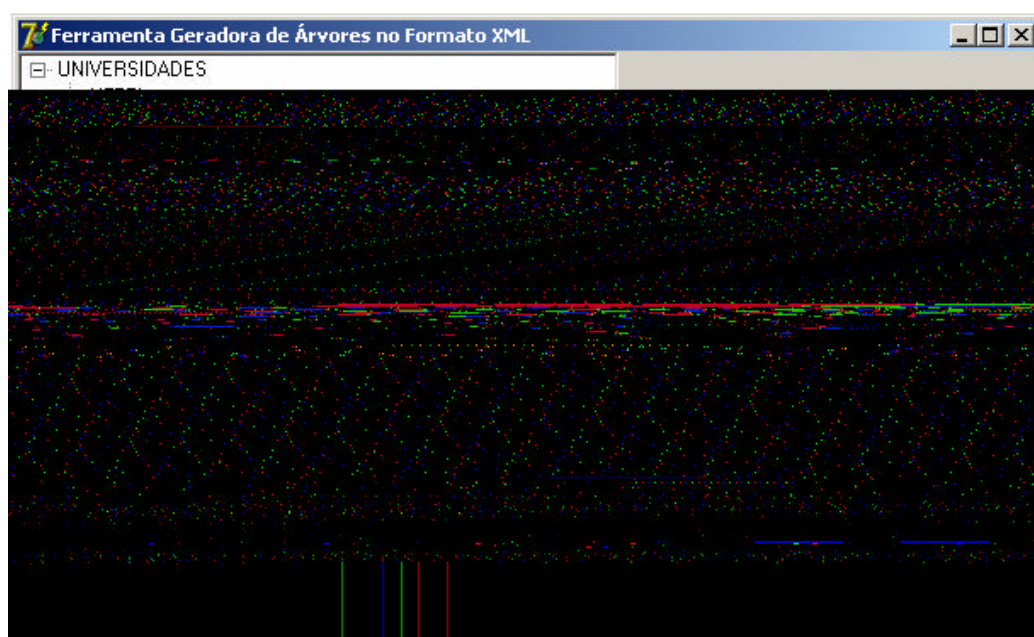


Figura 31 - Inserção do nodo UFRGS.

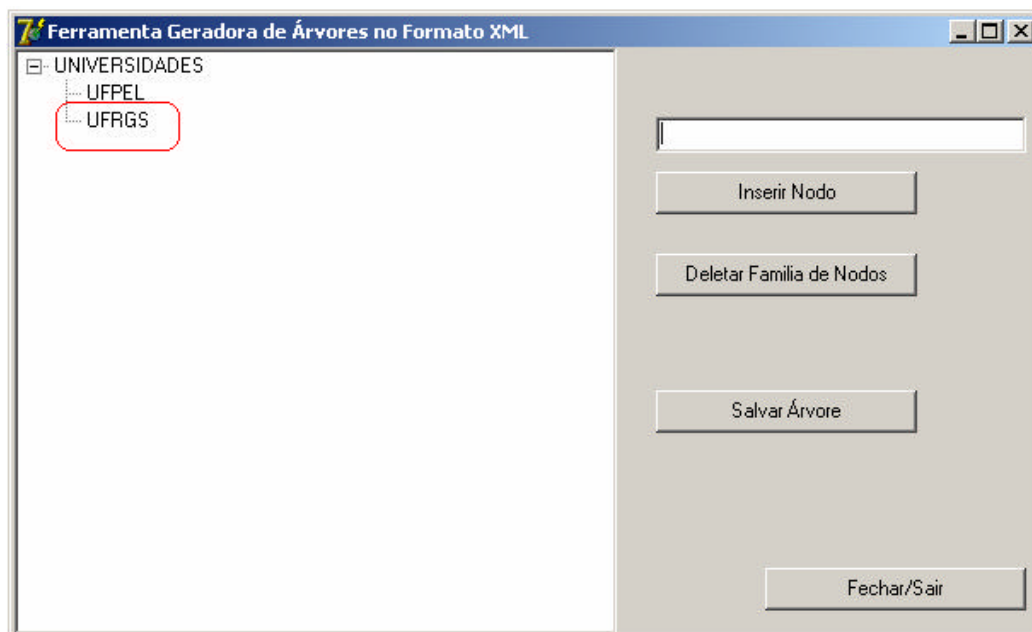


Figura 32 - Visualização do novo nodo inserido na árvore.

A ferramenta não possui restrição quanto ao número de nodos que podem inseridos, ficando a critério do usuário ou avaliador decidir qual o tamanho que a sua base de dados terá.

A ferramenta permite duas outras operações sobre os nodos inseridos: mover e renomear. As duas operações são realizadas utilizando a região à esquerda da ferramenta (área de visualização). Para mover um nodo (trocar sua posição na hierarquia) é necessário que ele seja selecionado na área de visualização e arrastado até a sua nova localização, podendo esse nodo ser filho ou pai de qualquer outro nodo. O processo pode ser visualizado através da Fig. 33, onde o nodo DINFO, inserido inicialmente como sendo filho de UFPEL, é movido para se tornar agora filho do nodo IFM (Fig. 34). Também não há nenhuma restrição em relação ao número de nodos que podem ser movidos na ferramenta, desde que eles sejam movidos um de cada vez.

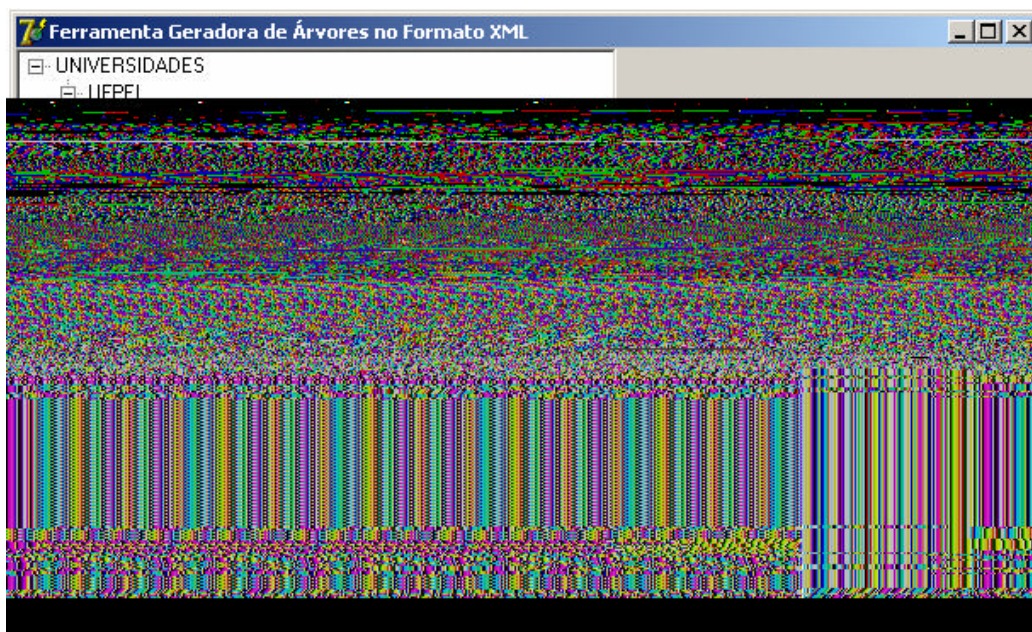


Figura 33 – Apresenta o nodo DINFO, inicialmente filho de UFPEL e irmão do IFM.

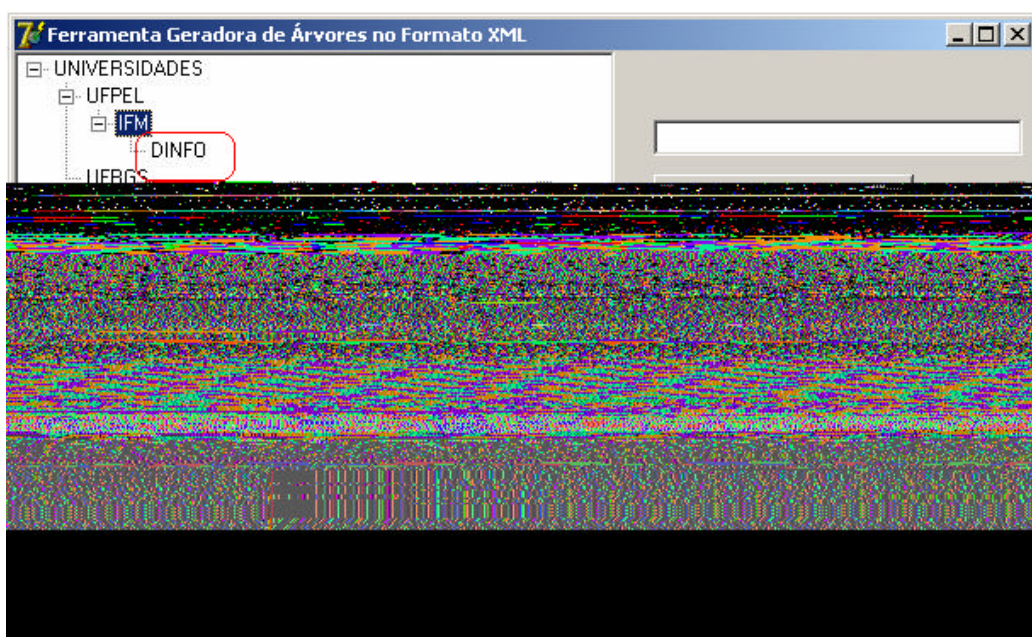


Figura 34 - Localização do nodo DINFO após ser movido para filho de IFM.

Em relação à operação de renomear um nodo, o usuário pode alterar o nome de qualquer nodo que já tenha sido inserido na árvore. Bastando para isso que o nodo seja selecionado e receba um novo “click” com o botão esquerdo do mouse. O campo onde está inserido o seu nome será realçado e um cursor piscante indicará que o usuário pode alterar o nome do nodo, conforme mostrado na Fig. 35.

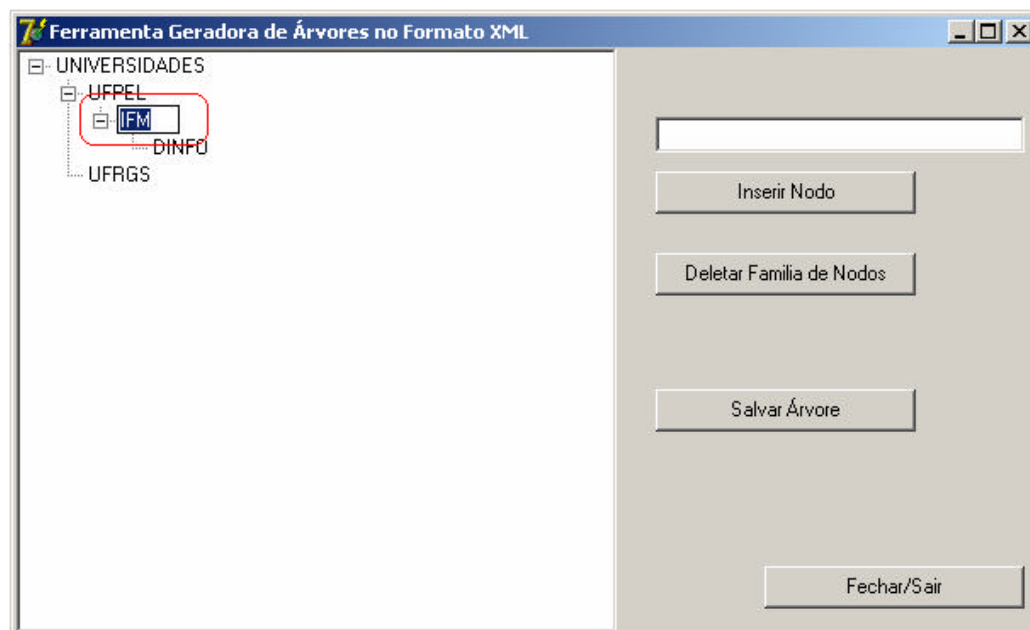


Figura 35 - Nodo IFM sendo renomeado.

A Fig. 36 apresenta o nodo selecionado já com o nome alterado para INSTITUTO DE FÍSICA E MATEMÁTICA.

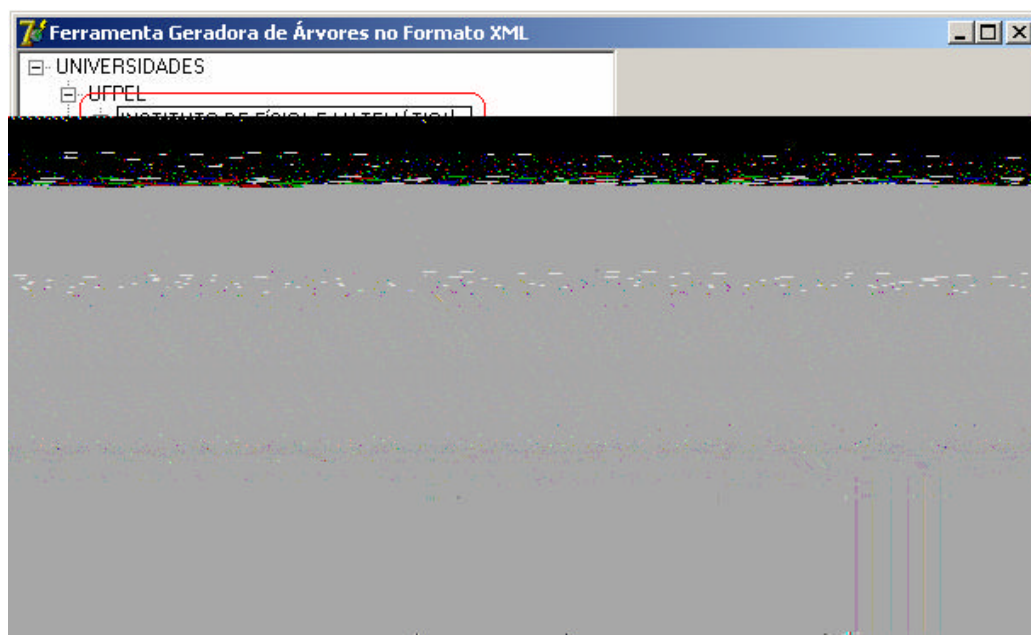


Figura 36 - Nodo renomeado para INSTITUTO DE FÍSICA E MATEMÁTICA.

A Fig. 37 mostra um novo nodo inserido na árvore com o nome de MECATRÔNICA. Porém, ele não se encaixa em nenhuma classificação adotada na construção da hierarquia. Portanto há a necessidade de remover o nodo da estrutura. Para que um nodo, ou uma família de nodos (subárvore), seja excluído

deve-se primeiro selecioná-lo e então utilizar o botão Deletar Família de Nodos, para que o mesmo não faça mais parte base de dados, conforme é mostrado na Fig. 38.

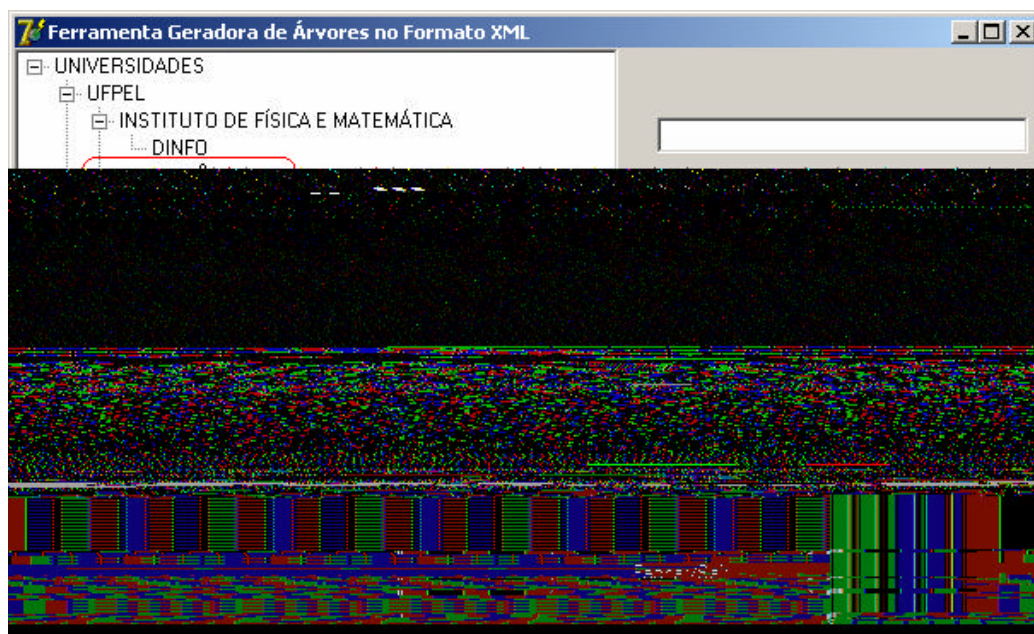


Figura 37 - Nodo MECATRÔNICA visualizado na árvore de dados.

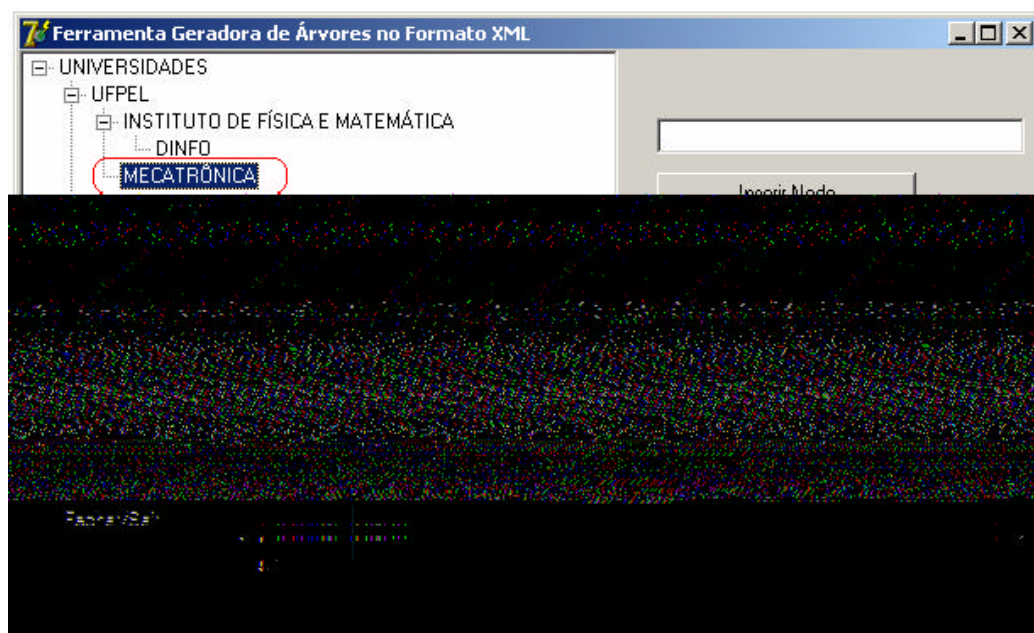


Figura 38 - Nodo MECATRÔNICA selecionado para sua remoção.

Com a construção da estrutura hierárquica terminada, a árvore deve ser salva em disco em um arquivo com a extensão XML. Este arquivo permitirá que a base dados seja acessada e visualizada por qualquer ferramenta que o reconheça.

O processo de salvar toda a estrutura é realizado pelo botão Salvar Árvore, visto pela Fig. 39.

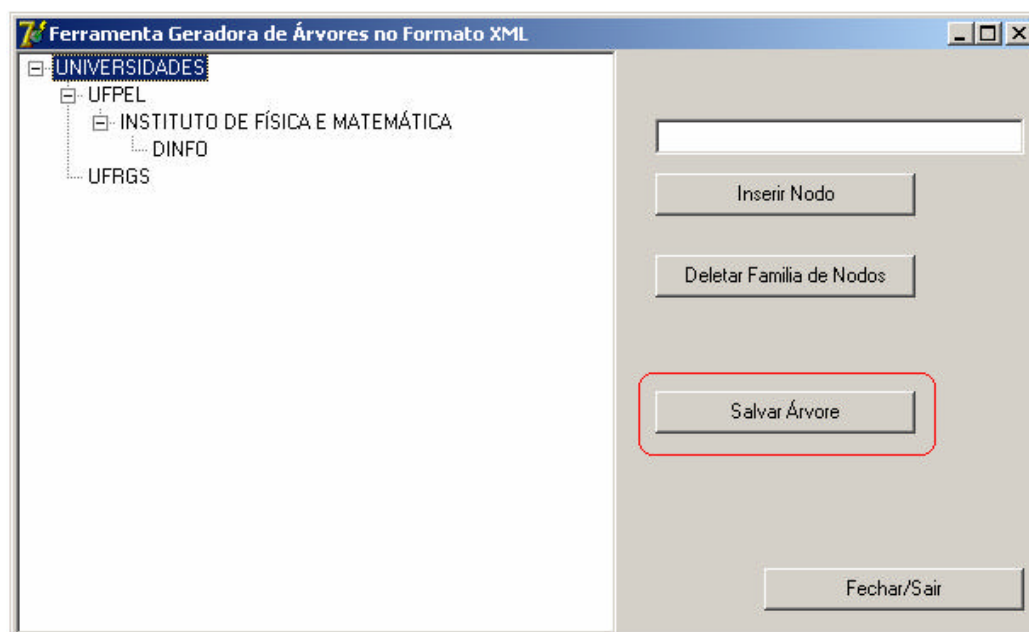


Figura 39 - Botão para salvar árvore construída.

Para nomear o novo arquivo é necessário digitar o seu nome na janela reservada para essa operação, como mostra a Fig. 40. O nome do arquivo gerado no processo descrito acima chama-se “exemplo”. O arquivo será representado em disco pelo nome final de exemplo.xml.

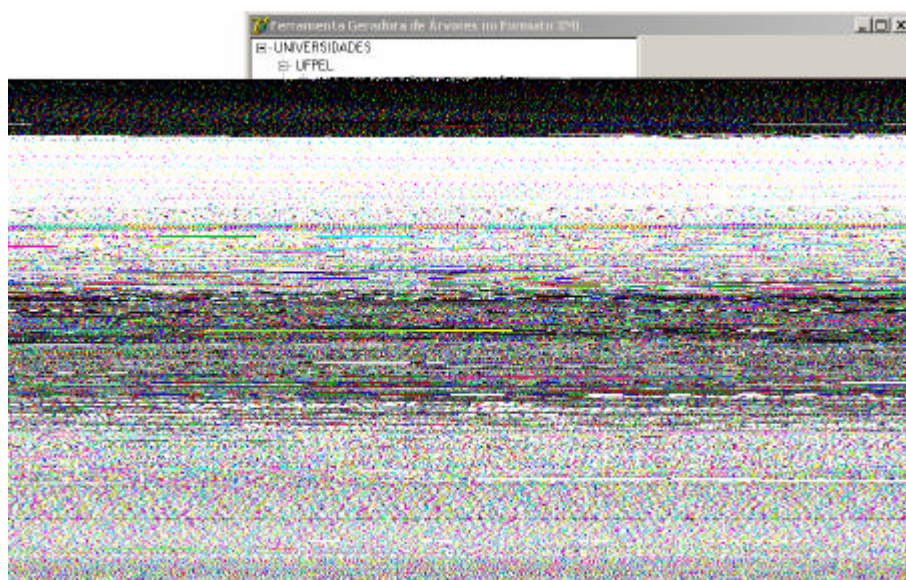


Figura 40 - Escolha do nome do arquivo a ser salvo em disco.

O botão Fechar/Sair, observado na Fig. 41, encerra o funcionamento da ferramenta assim que acionado. Por motivos de segurança, foi inserida a opção de salvar a árvore gerada no momento que se deseja encerrar a ferramenta, pois o usuário pode esquecer de salvá-la antes de sair do programa, evitando assim a perda desnecessária dos dados.

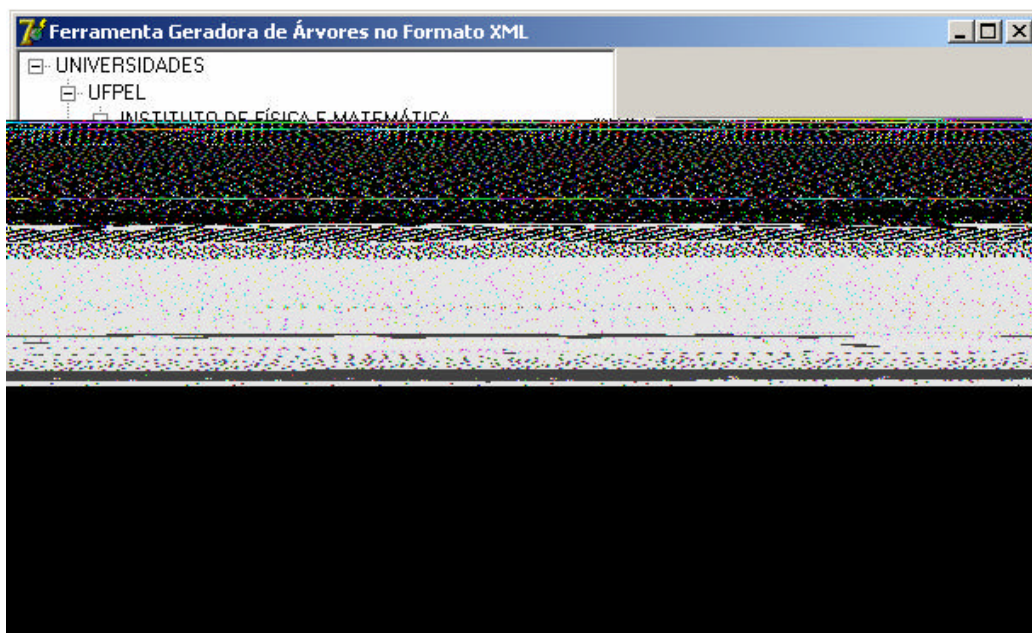


Figura 41 - Botão de saída da ferramenta.

A Fig. 42 apresenta a caixa de confirmação se o usuário deseja salvar ou não salvar a sua árvore no instante em que ele fecha a ferramenta.

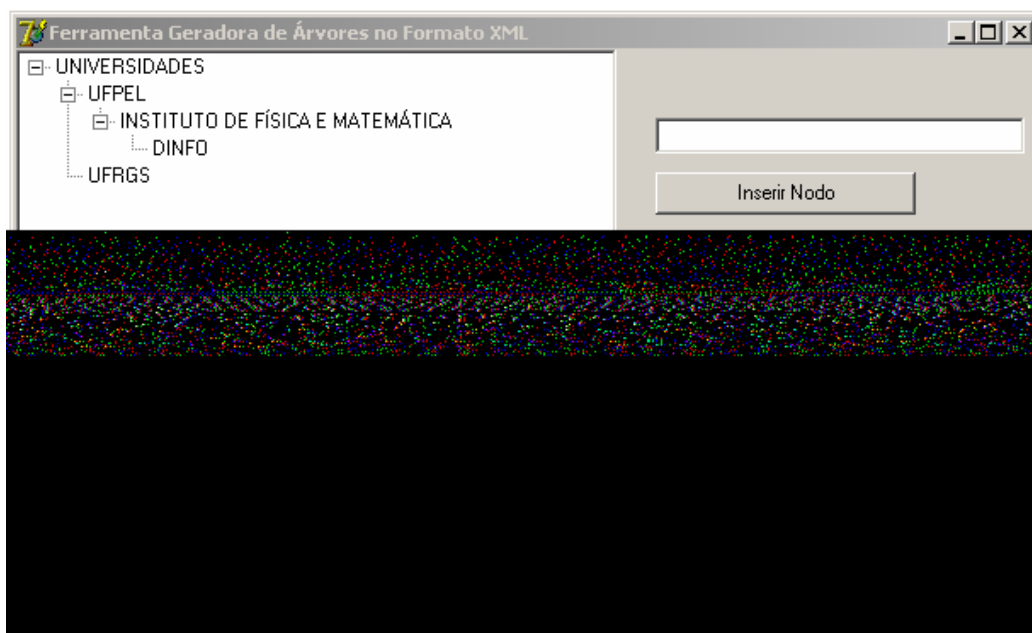


Figura 42 - Mensagem de confirmação de saída da ferramenta.

Escolhendo a opção “Sim” surge a mesma caixa de diálogo para escolher o nome do arquivo da opção Salvar Árvore. A opção “Não” encerra a ferramenta sem salvar o arquivo e a opção “Cancelar” retorna para a visualização da ferramenta, no mesmo estado antes da acionar o botão de saída.

4.3 Apresentação da Base de Dados no Formato XML

Na edição do congresso de InfoVis 2003 (INFOVIS, 2003), foi proposta uma padronização das bases dados na linguagem XML, para serem utilizadas no concurso da melhor técnica ou ferramenta de visualização para um cenário específico. Porém, ao se tentar utilizar as bases geradas, observou-se que as mesmas não podem ser utilizadas por boa parte das ferramentas de visualização, o que dificulta o processo de comparação entre as mesmas.

Procurando solucionar esse problema, uma vez que a existência de uma base de dado comum, entre outros elementos, é um parâmetro relevante no processo de avaliação das ferramentas de visualização de informações, foi desenvolvida a FGAXML. A estrutura que compõe a base de dados gerada oferece uma organização hierárquica para representar o armazenamento da informação, o que facilita o processo de visualização da estrutura mesmo quando não se utiliza alguma ferramenta específica para visualização de informações (Fig. 43).

```

- <tree version="1.0">
- <declarations>
  <attributeDecl name="name" type="String" />
</declarations>
- <branch>
  <attribute name="name" value="UNIVERSIDADES" />
- <branch>
  <attribute name="name" value="UFPEL" />
- <branch>
  <attribute name="name" value="INSTITUTO DE FÍSICA E MATEMÁTICA" />
- <leaf>
  <attribute name="name" value="DINFO" />
</leaf>
</branch>
</branch>
- <leaf>
  <attribute name="name" value="UFRGS" />
</leaf>
</branch>
</tree>

```

Figura 43 - Estrutura do arquivo exemplo.xml gerado pela FGAXML.

Para verificar a portabilidade da base de dados gerada pela FGAXML, foram utilizadas algumas ferramentas de visualização de informações. A Fig. 44 apresenta

um exemplo de visualização da estrutura gerada pela FGAXML, utilizando a ferramenta *Treemaps* versão 4.1.

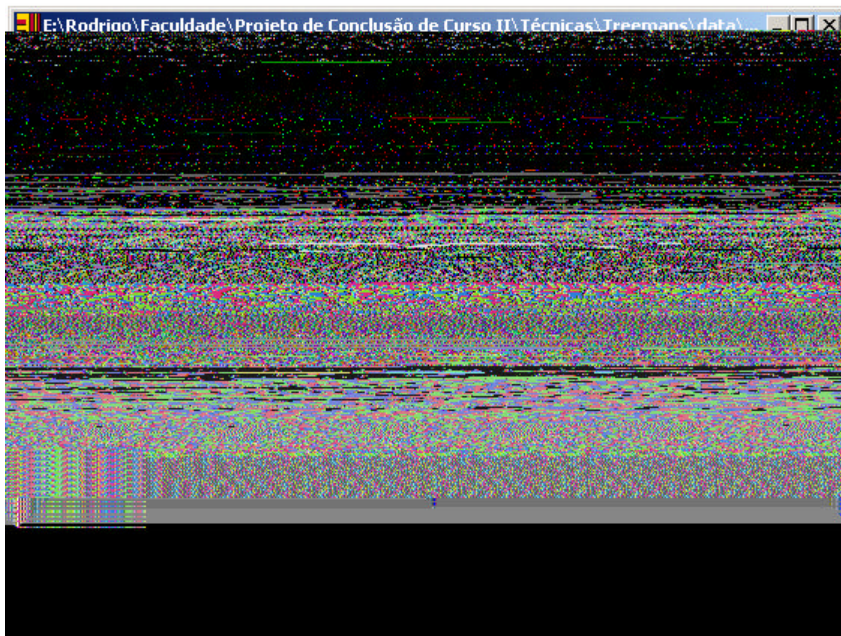


Figura 44 - Visualização da árvore gerada utilizando a ferramenta *Treemaps* 4.1.
Fonte: Adaptada de SHNEIDERMAN, 1992.

Outra ferramenta que utilizou a base de dados gerada pela FGAXML foi a *SpaceTree* (SPACETREE, 2006), como pode ser visto na Fig. 45.

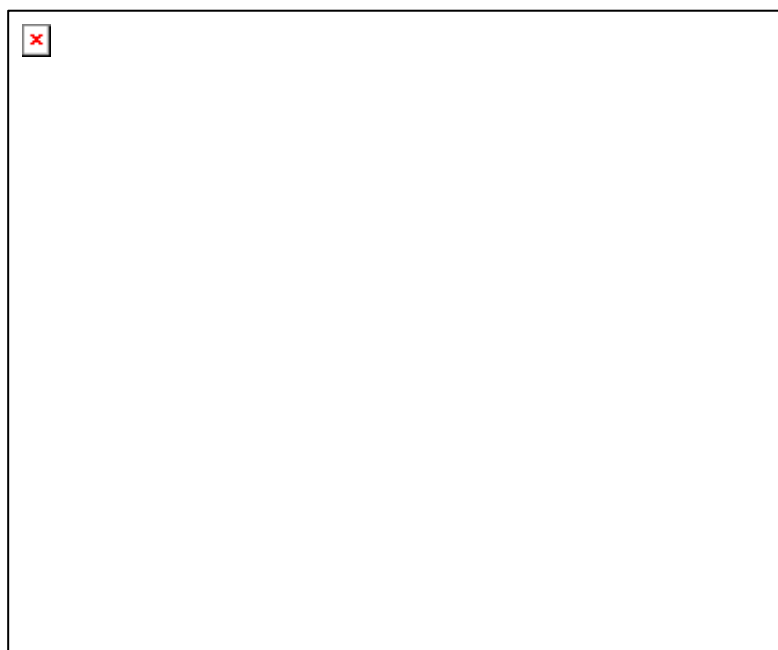


Figura 45 - Visualização da base de dados gerada pela FGAXML, através da ferramenta *SpaceTree*.
Fonte: adaptada de SPACETREE; 2006.

Como se pode observar pelos exemplos apresentados nas Fig. 43, 44 e 45, a construção de uma ferramenta que possibilite a geração de base de dados que seja utilizada pelas ferramentas de visualização é um elemento que deve ser ressaltado. Pois, isso permitirá disponibilizar um conjunto de dados comum, que possam ser utilizados no processo de avaliação das ferramentas de visualizações. A necessidade de gerar um conjunto de dados comum se deve pela falta de critérios utilizados na construção das bases de dados. O que trás dificuldades quando se deseja avaliar duas ou mais ferramentas e comparar os resultados obtidos utilizando as mesmas bases de dados. Muitas ferramentas utilizam uma codificação própria, criada especialmente para a base de dados de cada uma e somente esta ferramenta está apta a interpretar de forma automática e correta. Com a representação do armazenamento padronizada, os resultados da avaliação desses testes tenderiam a se aproximar da realidade, condizendo de forma determinante na escolha da ferramenta de visualização que melhor apresenta a informação.

Um outro agente motivador para a construção da FGAXML foi a constatação de que poucas ferramentas de visualização oferecem a possibilidade do usuário criar a sua própria base de dados, disponibilizando apenas exemplos prontos para testes. Além disso, até mesmo as ferramentas mais recentes que utilizam a linguagem XML não possuem uma documentação descrevendo a forma que acessam e salvam os arquivos das bases de dados.

A FGAXML, através de sua interface simples, clara e fácil de usar, procura preencher a lacuna existente, no que se refere à elaboração de bases de dados hierárquicas, fazendo com que os avaliadores possuam um mecanismo que possibilite a geração automática de base de dados.

5 Conclusão

Neste trabalho foi apresentado como objetivo principal a construção de uma ferramenta para a geração de uma base de dados que fosse utilizada pelas ferramentas de visualização que lêem o formato XML. Esse objetivo foi alcançado com a construção da FGXML, que através de uma interface simples, clara e fácil de usar, permite a geração dessa base de dados hierárquica. A escolha da linguagem XML na geração da base de dados realizada pela ferramenta desenvolvida neste trabalho foi a mais coerente, já que atende às necessidades exigidas para a obtenção de um “padrão”: ser uma linguagem de marcação aceita mundialmente, representa os dados de forma hierárquica, com a portabilidade garantida além, é claro, do fácil entendimento.

A geração de uma base de dados no formato XML, possibilitará que exista um parâmetro comum a diversas ferramentas facilitando, dessa maneira, o processo de avaliação das mesmas, pois permitirá o uso de um único grupo de bases de dados para avaliar diferentes técnicas de visualização de informação, normalizando o processo de avaliação e permitindo comparar ferramentas diferentes através dos resultados obtidos.

Entretanto, uma das maiores dificuldades encontradas neste trabalho, relacionada à escolha da linguagem XML, esteve na característica de criar infinitas linguagens a partir dela própria, chamadas metalinguagens. Esse fato dificulta a obtenção de um conjunto de dados comum, já que cada ferramenta poderá utilizar uma definição de tipos específica e única, o que impede a realização de comparações entre ferramentas distintas. Sendo a uniformização da base de dados gerada pela FGXML uma maneira de minimizar distorções, pelo menos no parâmetro base de dados, existentes no processo de avaliação.

Espera-se que este trabalho possa ainda dar contribuições importantes no que se refere à tentativa da utilização de um padrão de armazenamento das informações a serem visualizadas e, principalmente, possa fornecer subsídio para futuras pesquisas, mediante os estudos apresentados nos capítulos iniciais.

Como trabalho futuro, propõe-se a continuação do desenvolvimento desta ferramenta, pois ainda percebem-se algumas limitações da mesma quanto à possibilidade de realizar edições ou modificações em um arquivo XML salvo em disco. Também se nota a necessidade de adicionar a funcionalidade de converter bases de dados geradas por diferentes ferramentas de visualização, construídas ou não em XML, para o padrão adotado nesta pesquisa, com o auxílio de arquivos de definição de tipos em XML.

Referências

- ANDERSON, R. **Professional XML**. 1. ed. Rio de Janeiro: Ciência Moderna, 2001. 1266p.
- ANDREWS, K., HEIDEGGER, H. - Information Slices: Visualization and Exploring Large Hierarchies using Cascading, Semi-Circular Discs. In: IEEE SYMPOSIUM ON INFORMATIONVISUALIZATION, 1998. **Proceedings**: (InfoVis'98), 1998.
- CARD, S. K.; MACKINLAY, J.D. e SHNEIDERMAN, B. Information Visualization. In: READINGS IN INFORMATION VISUALIZATION - USING VISUALIZATION TO THINK, 1999, San Francisco. **Proceedings**: Morgan Kaufmann Publ., 1999. p. 1-34.
- CAVA, R. A. **Bifocal Tree: uma técnica de visualização de estruturas organizadas hierarquicamente**. 2002. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- CHERNOFF, H. The use of faces to represent points in k-dimensional space. **Journal of the American Statistical Association**, v.68, n. 342, p.361-368, 1973
- CHI, E. H. e RIEDL, J. T. An Operator Interaction Framework for Visualization Spreadsheets. In: PROCEEDINGS OF IEEE INFORMATION VISUALIZATION SYMPOSIUM, 1998. **Proceedings**: ..., 1998.
- DEITEL, H. M; DEITEL, P.J.; NIETO, R.; LIN, T. e SADHU, P.. **XML: Como Programar**. 1. ed. Porto Alegre: Bookman, 2003. 972p.
- FREITAS, C. M. D. S.; CHUBACHI, O. M.; LUZZARDI, P. R. G. e CAVA, R. A. Introdução à Visualização de Informações. **RITA**, v.8, n.2, p.143-158, 2001.
- FREITAS, C. M. D. S. e WAGNER, F. R. Ferramentas de suporte às tarefas da análise exploratória visual. **Revista de Informática Teórica e Aplicada**, v.2, n.1, p. 5-36, jan. 1995.
- FURNAS, G. W. Generalized fisheye views. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1986, New York. **Proceedings**: ACM Press, 1986. p.16-23.
- FURNAS, G. W. The FISHEYE View: A New Look at Structured Files. In: READINGS IN INFORMATION VISUALIZATION: USING VISION TO THINK, 1999, San Francisco. **Proceedings**: Morgan Kaufmann Publishers, 1999. p.312-330.
- GERSHON, N.; EICK, S. G. Information Visualization, **IEEE Computer Graphics and Applications**, v.17, n.4, p. 29-31, jul./ago., 1997.

GERSHON, N.; EICK, S. Visualization's New Tack. **IEEE Spectrum**, v.32, n.11, p.38-56, nov.1995.

HABER, R. B. e MCNABB, D. A. Visualization Idioms: A conceptual model for scientific visualization systems. **Visualization in Scientific Computing**, p. 74-93, 1990.

HOQUE, Reaz. **XML: for Real Programmers**. San Francisco – CA - USA: Morgan Kaufmann, 2000. 462p.

INFOVIS: IEEE Symposium on Information Visualization 2003. Disponível em: <<http://www.infovis.org/infovis2003/>>. Acesso em: 2006.

KEIM, D. A. Information Visualization and Visual Data Mining. **IEEE Transaction on Visualization and Computer Graphics**. v.7, n.1, p.100-107, Jan.-Mar. 2002.

LAMPING, J.; RAO, R.; PIROLI, P. A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In: CONFERENCE ON HUMAN FACTORS IN COMPUTER SYSTEMS, CHI, 1995, New York. **Proceedings**: ACM Press, 1995. p.401-408.

LIGHT, R. **Iniciando em XML**. São Paulo: Makron Books, 1999. p. 404.

LUZZARDI, P. R. G. **Avaliação de Técnicas de Visualização de Informações**. 2001. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

MACKINLAY, J. D., ROBERTSON, C. G., e CARD, S. K. The Perspective Wall: Detail and Context Smoothly Integrated. **Proceedings**: of ACM CHI'91, 1991. pp. 173-179

RAO, R. e CARD, S. K. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1994, Boston. **Proceedings**: ACM SIGCHI, 1994. pp. 318-322.

RAY, E.T. **Aprendendo XML**. 1.ed. Rio de Janeiro: Campus, 2001. 372p.

ROBERTSON, G., CARD, S., MACKINLAY, J. Cone Trees: Animated 3D Visualizations of Hierarchical Information. **Proceedings**: of ACM CHI'91, 1991, p.189-194.

SHNEIDERMAN, B. The eyes have it: a task by data type taxonomy for information visualizations. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES, BOULDER, 1996, Boulder. **Proceedings**: ... Set. 3-6, 1996. p. 336-343.

SHNEIDERMAN, B. Tree Visualization with Treemaps: A 2d space filling approach. **ACM Transactions on Graphics**, v.11, n.1, p.92-99, Jan. 1992.

SPACETREE: a novel node-link tree browser. University of Maryland. Disponível em: <<http://www.cs.umd.edu/hcil/spacetree/>>. Acesso em: julho de 2006.

SPENCE, R. e APPERLEY, M. D. Data Base Navigation: An Office Environment for the Professional. **Behaviour and Information Technology**, v.1, n.1, p.43-54, 1982.

TESLER, H. e HEILMEIER, G.H. **FSN: The 3D File System Navigator**. Mountain View, Silicon Graphics, 1992.

VELOSO, P.; SANTOS, C.; AZEREDO, P. e FURTADO, A. **Estruturas de Dados**. Rio de Janeiro: Editora Campus, 1986. 228p.

W3C: Introdução a XML Schema. Disponível em : <http://www.w3schools.com/schema/schema_intro.asp>. Acesso em: 2006.

WARE, C. **Information visualization: Perception for Design**. San Francisco – CA – USA: Morgan Kaufmann, 2000.

Anexos

ANEXO A – Representação da estrutura do documento XML disponibilizada pelo congresso InfoVis (INFOVIS, 2003).

```
<tree>
  <declarations>
    <attributeDecl name="name" type="String"/>
    <attributeDecl name="number" type="Real"/>
    <attributeDecl name="type" type="String"/>
  </declarations>
  <branch>
    <attribute name="name" value="sample things"/>
    <branch>
      <attribute name="name" value="plants"/>
      <leaf>
        <attribute name="name" value="oak"/>
        <attribute name="number" value="10"/>
        <attribute name="type" value="wild"/>
      </leaf>
      <leaf>
        <attribute name="name" value="afican violet"/>
        <attribute name="number" value="3"/>
        <attribute name="type" value="domestic"/>
      </leaf>
    </branch>
  </branch>
  <branch>
    <attribute name="name" value="animals"/>
    <branch>
      <attribute name="name" value="mammals"/>
      <branch>
        <attribute name="name" value="felines"/>
        <leaf>
          <attribute name="name" value="cat"/>
          <attribute name="number" value="10"/>
          <attribute name="type" value="domestic"/>
        </leaf>
        <leaf>
          <attribute name="name" value="lion"/>
          <attribute name="number" value="3"/>
          <attribute name="type" value="wild"/>
        </leaf>
      </branch>
    </branch>
    <branch>
      <attribute name="name" value="primates"/>
      <leaf>
        <attribute name="name" value="human"/>
        <attribute name="number" value="30"/>
        <attribute name="type" value="domestic"/>
      </leaf>
      <leaf>
        <attribute name="name" value="gorilla"/>
        <attribute name="number" value="8"/>
        <attribute name="type" value="wild"/>
      </leaf>
    </branch>
  </branch>
</branch>
</branch>
</tree>
```

ANEXO B - DTD que descreve a estrutura de árvore para visualização disponibilizada pelo congresso InfoVis (INFOVIS, 2003).

```
<!ELEMENT tree (declarations?, (branch | leaf)* )>
<!ATTLIST tree
  version CDATA #FIXED "1.0"
  src CDATA #IMPLIED
  desc CDATA #IMPLIED
  date CDATA #IMPLIED
  author CDATA #IMPLIED >

<!ELEMENT declarations (attributeDecl+ ) >
<!ELEMENT attributeDecl EMPTY>
<!ATTLIST attributeDecl
  name CDATA #REQUIRED
  type (Int|Integer|Long|Float|Real|String|Date|Category) "Int"
  control CDATA #IMPLIED >

<!ELEMENT branch ( attribute*, (branch | leaf)* ) >
<!ATTLIST branch
  label CDATA #IMPLIED >

<!ELEMENT leaf (attribute*) >
<!ATTLIST leaf
  label CDATA #IMPLIED >

<!ELEMENT attribute (#PCDATA) >
<!ATTLIST attribute
  name CDATA #REQUIRED
  value CDATA #REQUIRED >
```