

Navegador 3D para arquivos pessoais: um mapa virtual de tópicos informacionais

Trabalho de Conclusão de Curso
Engenharia da Computação

Victor Carvalho de Moura Braz
Orientador: Prof. Fernando Buarque de Lima Neto

Recife, dezembro de 2006



UNIVERSIDADE
DE PERNAMBUCO

Navegador 3D para arquivos pessoais: um mapa virtual de tópicos informacionais

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Victor Carvalho de Moura Braz
Orientador: Prof. Fernando Buarque de Lima Neto

Recife, dezembro de 2006



Victor Carvalho de Moura Braz

**Navegador *3D* para arquivos
pessoais: um mapa virtual de
tópicos informacionais**

Resumo

Dada a crescente proliferação das fontes de informação promovida pelos avanços tecnológicos e científicos, procurar a informação desejada e específica para cada caso se tornou uma tarefa cada vez mais laboriosa, tanto nas empresas quanto em aplicações pessoais. O desafio decorrente disto está centrado, então, na criação de mecanismos capazes de organizar, manipular e aperfeiçoar a navegação através da informação de forma eficiente e adequada. Utilizando os conceitos do padrão *ISO Topic Maps* como mecanismo para a organização e o acesso a arquivos pessoais aliado ainda a uma interface de visualização gráfica tridimensional, este trabalho de conclusão de curso apresenta o *TME*, o protótipo de um *software* que permite a construção de estruturas de conhecimento navegáveis sobre estruturas de arquivos. Com isso, espera-se que cada usuário possa criar uma estrutura de organização própria para seus arquivos, sem que haja para isso a necessidade de manipular seus arquivos fisicamente junto ao sistema de arquivos do sistema operacional de seu computador pessoal. Assim, o principal resultado esperado com este trabalho é contribuir para a apresentação de um novo modelo para a organização e o acesso a arquivos pessoais, que inove pela utilização do padrão *Topic Maps* aliado às facilidades de navegação gráfica tridimensional.

Abstract

Given the increasing proliferation of sources of information derived from technological and scientific advances, searching for the desired information became more laborious as time goes by as in companies as well as in personal applications. The derived challenge is then, the creation of mechanisms capable of organize, manipulate and improve navigation through the sea of information everywhere. Using the concepts of standard *ISO Topic Maps* as mechanism for organizing and accessing personal archives becomes a powerfull way to increase ones personal productivity. With this, we expect that each user can create its own structure of proper organization for its archives, without the need to physically manipulate its archives on the file system of the operational system. Thus, the main result expected for this work is to contribute with a new model for the organization and the easy access to personal archives. Furthermore, a three-dimensional graphical system for navigation innovates and extends the use of the standard *Topic Maps*.

Sumário

Índice de Figuras	v
Tabela de Símbolos e Siglas	vi
1 Introdução	8
1.1 Objetivos	9
1.2 Metodologia	10
1.3 Estrutura do Trabalho	10
2 Introdução à Visualização de Informações	12
2.1 Vantagens e Áreas de Aplicação	13
2.2 Mecanismos de Interação	13
2.3 Técnicas de Visualização	14
2.4 Resumo	16
3 Topic Maps	18
3.1 Histórico	19
3.2 Principais Conceitos	20
3.2.1 Tópicos	20
3.2.2 Ocorrências	20
3.2.3 Associações	21
3.2.4 Escopo	22
3.3 Visualização de <i>Topic Maps</i>	23
3.4 Resumo	25
4 TME	26
4.1 Idealização	26
4.2 Características	28
4.3 O Modelo Computacional para o <i>Topic Maps</i>	29
4.3.1 <i>Scope</i>	29
4.3.2 <i>Topic</i>	31
4.3.3 <i>Occurrence</i>	31
4.3.4 <i>Association</i>	31
4.4 Funcionalidades	32
4.4.1 Mapeamento de Arquivos (Manual)	32
4.4.2 Mapeamento de Diretórios (Automático)	34
4.4.3 Definição dos Escopos	35
4.5 Visualização e Navegação	36
4.5.1 Concepção	36
4.5.2 Interatividade e Navegabilidade	37
4.6 Resumo	38
5 Um Estudo de Caso de Uso Comparativo	39
5.1 Descrição	39

	iv
5.1.1 O Exemplo	39
5.1.2 A Dependência Físico-Hierárquica	40
5.1.3 A Solução do <i>TME</i>	41
5.2 Resumo	42
6 Conclusão e Trabalhos Futuros	43
6.1 Conclusão	43
6.2 Trabalhos Futuros	43

Índice de Figuras

Figura 1. Reconhecendo padrões. (retirada de [3])	9
Figura 2. Esquema básico da técnica de visualização <i>Cone Tree</i> [2].	14
Figura 3. Representação utilizada na técnica de visualização <i>TreeMaps</i> [12].	15
Figura 4. Representação utilizada na técnica de visualização <i>Hyperbolic Tree</i> [13].	15
Figura 5. Representação em dois focos utilizada na técnica <i>Node-Edge Diagram Layout</i> [15].	16
Figura 6. Relacionando recursos de informação. a) Sem <i>Topic Maps</i> . b) Com <i>Topic Maps</i> .	19
Figura 7. Ocorrências. Separação em dois domínios: um de tópicos e outro de recursos [6].	21
Figura 8. Mapa de tópicos: associações entre tópicos (figura adaptada de [6]).	21
Figura 9. Associações independem dos recursos (figura adaptada de [6]).	22
Figura 10. Efeito de se aplicar escopo (figura adaptada de [6]).	23
Figura 11. <i>Topic Maps</i> no <i>Ontopia Omnigator</i> .	24
Figura 12. <i>Topic Maps</i> no <i>web site</i> da <i>Highwire Press</i> .	24
Figura 13. <i>Windows Explorer</i> – utilitário gráfico para o gerenciamento de arquivos do SO da <i>Microsoft</i> .	27
Figura 14. Diagrama de classes <i>UML</i> do modelo computacional para o <i>Topic Maps</i> .	29
Figura 15. Modelo gráfico da implementação do conceito <i>Scope</i> .	30
Figura 16. <i>Association</i> : relacionamento bidirecional.	32
Figura 17. Tela para mapeamento de arquivo do <i>TME</i> .	33
Figura 18. Árvore de diretório no <i>Windows Explorer</i> .	35
Figura 19. Representação da idéia conceitual para a visualização de mapas de tópicos no <i>TME</i> .	37
Figura 20. Ambiente de visualização gráfica do <i>TME</i> .	38
Figura 21. Árvore de diretórios no <i>Windows Explorer</i> : exemplo de organização um.	40
Figura 22. Árvore de diretórios no <i>Windows Explorer</i> : exemplo de organização dois.	40
Figura 23. Exemplo de mapeamento de arquivos para <i>Topic Maps</i> : solução <i>TME</i> .	41

Tabela de Símbolos e Siglas

Web – World Wide Web
ISO – International Standardization Organization
GPS – Global Positioning System
SGML – Standard Generalized Markup Language
CApH – Conventions for the Application of HyTime
XML – eXtensible Markup Language
XTM – XML Topic Maps
TME – Topic Maps Explorer
SOs – Sistemas Operacionais
UML – Unified Modeling Language
URI – Universal Resource Identifier

Agradecimentos

Almejando uma qualificação profissional de melhor qualidade, vim para Recife deixando casa, familiares, amigos e minha querida cidade natal, Caruaru. Agradeço a todos que contribuíram para tornar esse sonho uma realidade.

Em primeiro lugar, meus queridos e amados pais, Anacleto e Izabel, que tanto se esforçaram fazendo sempre o possível – e por vezes o impossível – para me possibilitar as melhores condições de vida e estudo. Se no início reclamei, quando vim para Recife cursar já o 3º ano do ensino médio, hoje reconheço a decisão acertada que eles tomaram.

Um agradecimento especial a meu querido avô, Manoel Afonso de Moura (Seu Moura, o menino Moura), que participa ativamente de minha vida, sempre me apoiando com palavras de incentivo e carinho.

Os novos e grandes amigos que conheci na faculdade, Júlio e Rafael. Companheiros de diversos momentos, seguimos juntos desde o início do curso e hoje os tenho como os dois maiores amigos conquistados nesta empreitada. Filipe, Diogo, Tiago, Marcos e todos os *amigosPoli*.

Impossível esquecer Renata, amizade tardia, mas com a qual tenho convivido e aprendido muito. Sem o apoio dela, talvez eu não tivesse conseguido concluir este TCC. Ela e sua inesquecível frase: “Se você continuar, segunda-feira acaba”. Querendo dizer que o fim do aperreio e da pressão estava próximo (segunda-feira, o dia da entrega do TCC).

Natália, minha doce Natália. Talvez a maior rival desta monografia, com quem disputava todos os meus fins-de-semana, foi também quem sempre soube me acalmar e animar nos momentos de fraquejo.

Não poderia esquecer o professor Fernando Buarque, com quem tanto aprendi ao longo desses anos de academia. Iniciação científica, estágio supervisionado, salas de aula e orientação de TCC o fizeram uma das minhas grandes influências.

Por fim, mais que um agradecimento, eu dedico esta e todas as minhas outras vitórias que estejam por vir a meu saudosíssimo avô, Severino Braz, a quem tenho profundo carinho e admiração. Valeu vô!

Capítulo 1

Introdução

Historicamente, o homem vem criando mecanismos cada vez mais sofisticados para registro e recuperação da informação [1]. Se no início da civilização os homens armazenavam e recuperavam informações apenas em suas mentes, com o advento da escrita, essa tarefa passou a ser realizada em suportes físicos, como o papel. Com estes surgiram as bibliotecas e suas conhecidas técnicas para organização e localização dos documentos e das informações contidas nestes, podendo citar: esquemas de ordenação e classificação, catálogos, sumários, índices etc.

Entretanto, os avanços alcançados pelas telecomunicações deram suporte a um aumento cada vez maior da produção e da demanda de informações. Invenções como o telefone e a radiodifusão quebraram barreiras de tempo e espaço, facilitando o intercâmbio e o conseqüente aumento da produção de informações [1]. A chegada do computador e da *Internet* então, elevou tanto a capacidade de produção como a de armazenamento e disseminação das informações [1]. Procurar a informação desejada e específica para cada necessidade se tornou uma tarefa cada vez mais laboriosa.

Essa nova realidade tecnológica fez com que fossem criadas novas técnicas e ferramentas computacionais capazes de organizar e manipular grandes quantidades de informação. Dentre estas, recentemente se destacam as técnicas de visualização de informações [2][3].

Área emergente da Ciência, Visualização de Informações estuda formas de apresentar dados visualmente de tal modo que relações entre os mesmos sejam melhor compreendidas ou novas informações possam ser descobertas [3]. Seu objetivo final é auxiliar no entendimento de determinado assunto, o qual, sem uma visualização, exigiria maior esforço para ser compreendido [3]. Sua grande vantagem está no fato de ela envolver o sentido humano que possui maior capacidade de captação de informações por unidade de tempo [3]. Treinado para reconhecer padrões, o sistema visual humano é rápido e paralelo, permitindo analisar um objeto em detalhes sem perder a atenção para o que esteja acontecendo a seu redor [3]. Ao analisarmos a Figura 1, por exemplo, fácil e rapidamente encontramos a estrela em meio aos quadrados.

A depender da aplicação, uma visualização pode tomar vários e diferentes enfoques, tais como Visualização Científica, Visualização de Informação, Visualização Geográfica, Visualização de Negócios, Visualização Estatística, Visualização de Processo e Visualização de *Software* [3]. Em comum, todas essas aplicações necessitam armazenar e recuperar informações [4]. Para isso, os sistemas operacionais de computadores fornecem um sistema de arquivos que define a forma físico-estrutural sobre a qual as informações são armazenadas e organizadas.

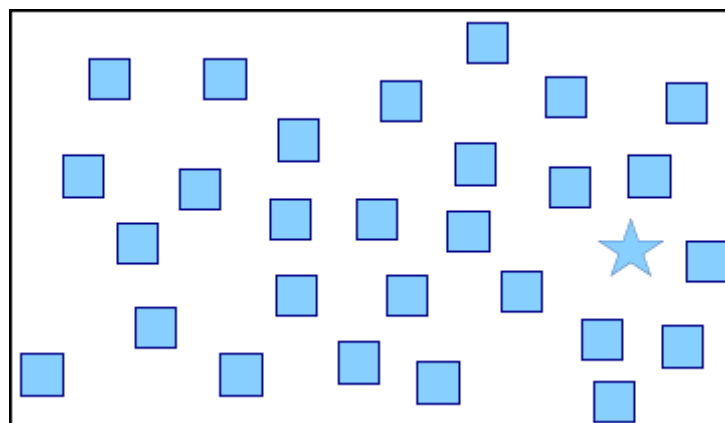


Figura 1. Reconhecendo padrões. (retirada de [3])

Arquivos são uma forma de abstração fornecida pelo sistema operacional, para permitir que as aplicações computacionais possam armazenar informações de maneira exclusiva e persistente, em um meio de armazenamento de dados sobre o qual possa realizar operações de leitura e escrita [4]. No entanto, no âmbito pessoal, as formas como os sistemas operacionais hoje no mercado disponibilizam a visualização e o acesso aos arquivos estão limitadas à estrutura físico-hierárquica da árvore de diretórios sobre a qual seus arquivos são organizados. Isso significa que a identificação dos arquivos é realizada a partir de sua localização física diante de toda a estrutura da árvore de diretórios do sistema, de modo que cada arquivo tem um endereço único.

Neste contexto surge o *Topic Maps*, um padrão ISO que permite a construção de estruturas de conhecimento navegáveis como mecanismo de organização e localização de recursos de informação, que podem ser de qualquer natureza (documentos, repositórios, arquivos etc.) [1][5][6]. Segundo seus conceitos, essas estruturas de conhecimento são representadas a partir da definição de meta dados – dados que descrevem outros dados – e têm como principal entidade o tópico.

Sua grande vantagem está justamente, na separação que realiza entre o domínio dos relacionamentos semânticos dos mapas de tópicos (criados sobre os recursos) e o domínio dos recursos da informação em si. Com a separação de domínios, o *Topic Maps* objetiva a captura do conhecimento e sua visualização de uma forma adequadamente estruturada, sem a necessidade de alterar ou manipular fisicamente os recursos originais. Esse modelo baseado na informação e não nos recursos pode contribuir para uma nova abordagem para a representação e o acesso à informação [1].

Este trabalho apresenta o *TME (Topic Maps Explorer)*, como protótipo de um *software* que propõe um novo método para a visualização, a organização e o acesso a arquivos pessoais, utilizando os conceitos do padrão *Topic Maps*, porém aliado a um modelo de visualização e navegação gráfica tridimensional. Seu principal objetivo é possibilitar a criação de estruturas de conhecimento sobre o conteúdo dos arquivos pessoais. Com isso, espera-se que cada usuário possa criar sua estrutura de organização própria para seus arquivos, sem que haja para isso a necessidade de manipulá-los fisicamente diante do sistema de arquivos do sistema operacional.

1.1 Objetivos

Este trabalho tem como escopo o gerenciamento das informações restritas no âmbito pessoal, tendo como meta aplicar os conceitos do *Topic Maps* ao gerenciamento de arquivos pessoais (de classes diversas).

O ideal da aplicação do *Topic Maps* ao gerenciamento de arquivos pessoais é possibilitar à construção de estruturas de conhecimento sobre o conteúdo desses arquivos. E então, através dessas estruturas de conhecimento, acessar e organizar os arquivos.

Contanto, para tornar efetiva a aplicação do *Topic Maps* ao gerenciamento de arquivos pessoais, é necessário um ambiente gráfico que permita não apenas a criação dos mapas de tópicos sobre os arquivos, mas que permita também a visualização e, sobretudo, a navegação sobre a estrutura de mapas de tópicos criada. Assim, o principal objetivo deste trabalho é a construção de um protótipo de *software* para um navegador gráfico tridimensional que represente a estrutura e os conceitos do padrão *Topic Maps*, possibilitando sua aplicação sobre arquivos pessoais.

1.2 Metodologia

Para a realização do trabalho proposto, foi necessária a sua divisão em algumas etapas. Primeiramente, o aprendizado e a fixação dos conceitos pertencentes ao padrão *Topic Maps*. Então, um estudo detalhado das especificações desse padrão foi realizado nesta fase inicial do trabalho.

Fixados os conceitos acima, a próxima tarefa se caracterizou pela modelagem e posterior implementação computacional para o modelo do padrão *Topic Maps*. É sobre esse modelo computacional que se dá todo o funcionamento do protótipo construído, assim, sua implementação foi realizada de maneira modular e estruturada.

Após definido e implementado o modelo computacional para o *Topic Maps*, a próxima etapa do trabalho se deveu à construção do protótipo propriamente dito. Mais especificamente, nessa etapa foi concebida e desenvolvida a forma visual/gráfica para o ambiente de navegação da aplicação. Na fase de concepção, um estudo inicial sobre técnicas de Visualização de Informações foi realizado.

Então, ao fim do desenvolvimento do protótipo, uma última etapa de provação e argumentação da eficácia do protótipo construído foi realizada. Sendo feito um estudo de caso de uso, comparando-o ao *Windows Explorer*, da *Microsoft*.

1.3 Estrutura do Trabalho

A presente monografia inicia com uma introdução às técnicas de Visualização de Informações, no Capítulo 2. Onde são apresentadas as vantagens de uma visualização e algumas áreas de aplicação, incluindo alguns exemplos de aplicações reais.

O Capítulo 3 apresenta o padrão *Topic Maps*. Como surgiu, seu propósito e algumas aplicações. Seus principais conceitos são explorados segundo as respectivas relevâncias para este trabalho.

No Capítulo 4, o *TME* é apresentado. A idealização da aplicação do padrão *Topic Maps*, seguida do detalhamento do modelo computacional deste padrão são apresentados de início. Ali também são analisadas e discutidas as suas principais funcionalidades, bem como são realizadas. Terminando com a concepção do modelo de visualização gráfica tridimensional e ilustração da utilização da ferramenta em si.

Para enfatizar os benefícios objetivados pelo *TME*, um estudo de caso de uso comparativo entre o *TME* e o utilitário gráfico para gerenciamento de arquivos do sistema operacional da *Microsoft*, o *Windows Explorer*, é apresentado no Capítulo 5.

Finalmente, o Capítulo 6 conclui o trabalho descrito nesta monografia, apresentando conclusões e propostas para possíveis trabalhos futuros.

Capítulo 2

Introdução à Visualização de Informações

A área de Visualização de Informações [2][3] aplica técnicas de computação gráfica, geralmente interativas, visando a auxiliar o processo de análise e compreensão de um conjunto de informações [2]. Representações visuais ou gráficas correspondem, justamente, às “figuras” ou “imagens” empregadas para representar esse conjunto (ou subconjunto) de informações o qual se deseja analisar [2].

Em geral, as técnicas de visualização procuram representar as informações sob um determinado domínio de aplicação de modo que a representação visual gerada explore a capacidade de percepção do homem, e que este, a partir das relações visuais exibidas, interprete e compreenda as informações apresentadas [2].

Assim, uma das principais considerações a ser feita no desenvolvimento de sistemas de visualização está na escolha de qual técnica de visualização empregar diante de uma determinada aplicação ou situação; segundo o tipo da informação que o sistema venha tratar. De modo que, a técnica de visualização escolhida venha facilitar a interpretação e compreensão das informações por parte dos usuários. Outro fator que deve ser considerado na escolha da técnica de visualização são as próprias tarefas que serão realizadas pelo usuário [2].

Uma vez que uma representação visual estática não constitui, normalmente, um método de grande valia, é necessário definir um mecanismo de interação que possibilite a manipulação das informações sob sua representação visual. Desta forma, o mecanismo deve permitir a realização tanto de manipulações geométricas (rotações e *zoom* na representação gráfica, por exemplo), quanto analíticas (redução ou expansão do conjunto de informações, conforme algum critério determinado pelo usuário) [2].

Aparentemente simples, a construção de sistemas para visualização de informações pode tornar-se muito complexa em virtude, principalmente, da necessidade de criação de uma *metáfora visual* que codifique visualmente o conjunto de informações. Ao mesmo tempo, ela deve garantir o grau de fidelidade necessário à aplicação, os *mecanismos de interação* necessários para organizar e manipular o conjunto de informações [2]. Estes aspectos levam diretamente a questões de projeto de interfaces gráficas e avaliação de usabilidade, inerentes ao contexto interação homem-computador (*HCI – Human-Computer Interaction*) [7].

2.1 Vantagens e Áreas de Aplicação

O ditado popular “uma imagem vale mais que mil palavras” expressa bem a grande vantagem de aplicar técnicas de visualizações de informação: a capacidade de condensar grandes quantidades de informações a partir de uma simples visualização [3]. Isto acontece pelo fato de o processo de visualização envolver o sentido humano que possui maior capacidade de captação de informações por unidade de tempo. Rápido e paralelo, o sentido da visão permite, inclusive, prestar atenção em um objeto de interesse especial sem perda significativa de atenção (obviamente, com menos detalhes) do que esteja acontecendo ao redor [3].

Além disso, o fato de o sistema visual humano ser naturalmente treinado para reconhecer padrões [3], traz às visualizações, por si só, benefícios. Por funcionarem como uma extensão da memória humana o que auxilia o processo cognitivo [3]. Imagens nos ajudam a entender problemas e a encontrar soluções para os mesmos, facilitando inclusive, a memorização do objeto visualizado [3].

Visualizações de informações têm sido desenvolvidas para uma grande diversidade de aplicações, desde árvores genealógicas até monitoramento de bolsa de valores ou organogramas de empresas [3]. Na área de Computação são muito utilizadas em Mineração de Dados (*Data Mining*) [8] e em Engenharia de *Software* [9].

Devido às técnicas de Mineração de Dados atuais ainda não constituírem soluções totalmente eficazes [10], *Data Mining* ainda requer uma interação muito forte com analistas humanos que são em última instância, os principais responsáveis pela determinação do valor dos resultados encontrados. Um bom mecanismo para visualização desses resultados poderia auxiliar em muito o trabalho desses analistas.

Não obstante, visualizações de *software* auxiliam o programador a analisar e a entender a estrutura e o funcionamento de um programa em um nível maior de abstração, quando comparada a uma simples leitura do código fonte, por exemplo [3]. Mas, é na *Internet* que a área de visualizações de informações vem se destacando, dada a grande quantidade de dados que gera e disponibiliza todos os dias.

2.2 Mecanismos de Interação

Basicamente, o mecanismo de interação de um sistema visual permite ao usuário navegar sobre a informação, realizando ações que, eventualmente, causem alterações na sua forma e conteúdo. De modo que possam ser observados e analisados novos aspectos e características do conjunto de informações.

Suas funções de interação estão intimamente ligadas às formas visuais de representação empregadas. Numa representação plana, o deslocamento vertical/horizontal de uma *scroll bar*, caracteriza uma função de interação que, no caso de uma representação tridimensional, seria análoga à função de deslocamento de uma câmera virtual no espaço, por exemplo.

Abstraídas pelas funções de interação visual, estão as funções de seleção dos elementos de informações, que podem provocar tanto um reposicionamento do conjunto de informações para uma melhor visualização, como um detalhamento (*zoom* semântico) de uma parte desse conjunto, em eventual supressão de outra parte [2]. Especialmente estas funções, estão relacionadas a certos critérios inerentes a cada aplicação, podendo ocorrer três situações em geral: (1) a representação visual é alterada para mostrar apenas a região de interesse selecionada; (2) a região de interesse ocupa o campo de visão principal e o restante do conjunto de dados é mantido em área à parte ou (3) a região de interesse e a visão geral são exibidas concomitantemente [2].

2.3 Técnicas de Visualização

Técnicas de visualização se baseiam numa representação visual e em mecanismos de interação que possibilitam ao usuário manipular essa representação, de modo que possa melhor compreender o conjunto de informações ali representado [2].

Em geral, as técnicas de visualização procuram utilizar elementos visuais que sejam conhecidos e condizentes ao conjunto de informações que representam. Deste modo, cada elemento visual carrega consigo algum conteúdo semântico, que seja intuitivo ao usuário. Como exemplo: utilitários para gerenciamento do sistema de arquivos dos sistemas operacionais convencionais. Esses utilizaram uma estrutura hierárquica de diretórios, que tem como principal elemento visual árvores de diretórios na forma de pastas. Não obstante, conjuntos de informações ou elementos que guardam uma organização hierárquica, como diretórios de arquivos, ou que apresentam relações, como documentos num *web site*, são amplamente utilizados na demonstração de técnicas de visualização [2].

A técnica *Cone tree* [11] utiliza uma representação tridimensional de informações hierárquicas, na qual o nodo raiz de uma árvore, representado por um retângulo, fica localizado no ápice de um cone translúcido, enquanto todos os seus filhos estão dispostos na base circular do cone (Figura 2). O objetivo desta técnica é apresentar uma estrutura na qual uma hierarquia inteira ou uma grande parte dela seja visível sem necessidade de *scrolling* e que, ao mesmo tempo, permita a supressão ou exibição de nodos durante a navegação [2].

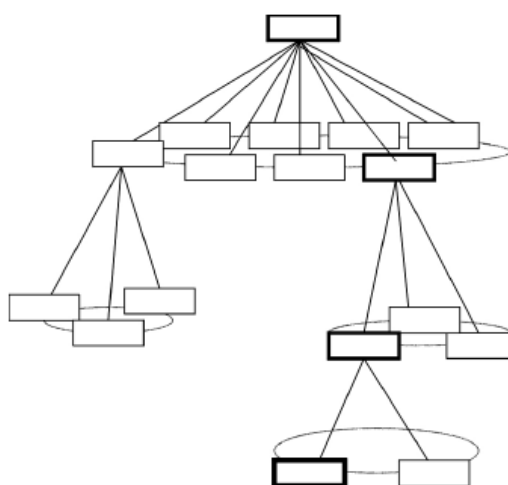


Figura 2. Esquema básico da técnica de visualização *Cone Tree* [2].

Uma abordagem diferente, é apresentada por *Johnson e Shneiderman* [12], com a técnica *TreeMaps*. Nesta técnica (Figura 3), também conhecida como *space-filling*, uma estrutura hierárquica, como a árvore de diretórios de um sistema de arquivos, é representada pela subdivisão sucessiva do espaço de tela. Cada subespaço representa um diretório e é subdividido em função dos subdiretórios e arquivos que o compõem [2]. Um interessante exemplo desta técnica de visualização pode ser observado no *Marumushi*¹, um *web site* de notícias, que permite ao usuário filtrar informações de acordo com o tema, a data e o país.

¹ Disponível em: <<http://www.marumushi.com/apps/newsmap/newsmap.cfm>>.

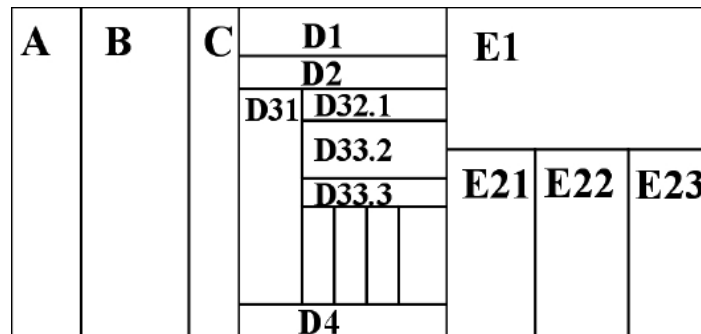


Figura 3. Representação utilizada na técnica de visualização *TreeMaps* [12].

Outra técnica que representou um considerável avanço em termos de apresentação e navegação foi a técnica *Hyperbolic Tree* [13] (Figura 4). Nesta técnica, a estrutura de uma árvore é mapeada em uma representação de um plano hiperbólico a ser projetado na tela. Uma propriedade conveniente dessa técnica é que o plano de representação, definido por uma circunferência, possui seu centro coincidente com o centro da tela e seu raio cresce exponencialmente. Isso significa que há mais espaço disponível para se compactar informações na medida em que se aproxima da periferia do desenho. Com essa representação, consegue-se disponibilizar cerca de dez vezes mais vértices de uma árvore que utilizando uma visualização em plano cartesiano [3].

A navegação em um *Hyperbolic Tree* é também mais efetiva. Mudanças de foco podem ser realizadas através de movimentos simples do *mouse*, objetivando selecionar vértices de maior interesse. Isso move o vértice selecionado para o centro da tela, ocasionando uma compactação das informações que estão distantes do mesmo. Além disso, o sistema realiza transições gradativas e suaves da mudança do ponto de foco, a fim de preservar o mapa mental construído para aquela estrutura [3]. Uma extensão tridimensional dessa técnica foi desenvolvida por *Munzner* [11]. Atualmente, *Hyperbolic Tree* é empregada no *browser* comercializado pela *Inxight Software, Inc*².

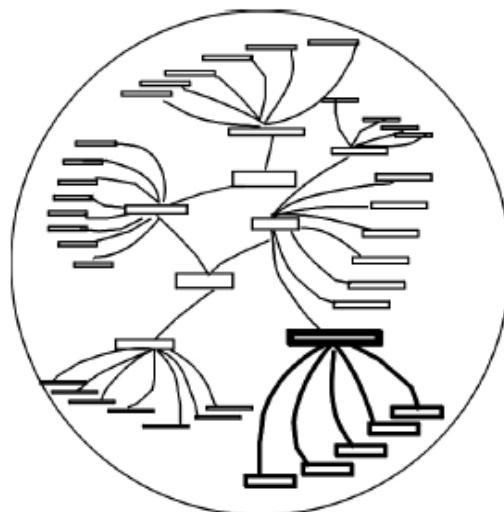


Figura 4. Representação utilizada na técnica de visualização *Hyperbolic Tree* [13].

² Disponível em: <<http://www.inxight.com>>.

Contendo características de duas abordagens, hiperbólica e *space-filling*, a técnica *Node-Edge Diagram Layout* [15] exibe uma hierarquia através de um diagrama composto em duas áreas conexas, uma área de contexto e uma área de detalhe (Figura 5).

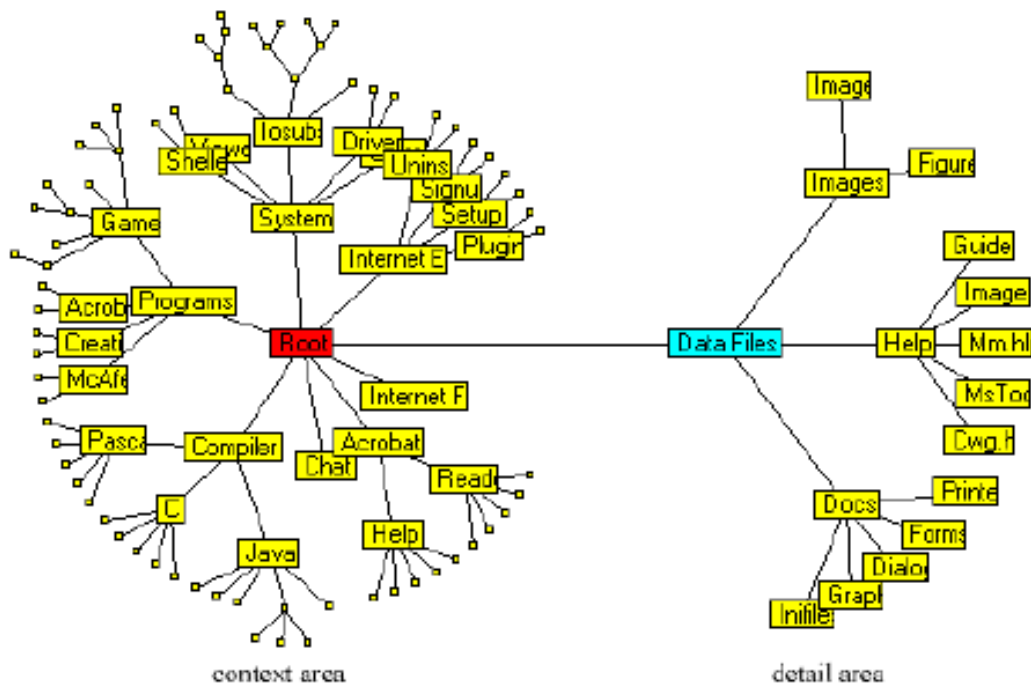


Figura 5. Representação em dois focos utilizada na técnica *Node-Edge Diagram Layout* [15].

Inicialmente, a raiz da hierarquia está no foco da área de detalhe, não havendo diagrama na área de contexto. Quando um nodo é selecionado, este passa a ser o foco na área de detalhe e o seu nodo-pai passa a ser o foco na área de contexto. Estes focos definem áreas circulares e são separados por uma distância arbitrária. Desta forma, possibilita ao usuário uma visão detalhada da sub-árvore que contém o nodo de interesse sem que a percepção da hierarquia completa seja perdida. Expansão, poda (*prunning*) e seleção são operações disponibilizadas por esta técnica, que são realizadas sempre mantendo a sub-árvore visível, o que reduz no usuário a sensação de perda de contexto [2].

2.4 Resumo

Visualização de Informações estuda formas visuais para apresentação de dados, de modo que as relações entre os mesmos possam ser melhor analisadas e compreendidas. Uma visualização busca a explorar a capacidade de percepção natural que o homem tem de, a partir de relações visuais, interpretar e compreender as informações apresentadas.

Diversas aplicações que utilizam representações visuais de dados vêm sendo desenvolvidas. Particularmente no campo da Computação, elas ganham atenção especial nas áreas de Mineração de Dados (*Data Mining*) e Engenharia de *Software*.

Essa diversidade de aplicações faz com que uma das principais considerações a ser feita no desenvolvimento de sistemas de visualização seja justamente, a escolha de qual técnica de visualização empregar diante de uma determinada aplicação ou situação.

Uma técnica de visualização é baseada numa representação visual e em um mecanismo de interação que possibilite ao usuário manipular essa representação, de modo que possa melhor

compreender o conjunto de informações ali representado. Em geral, três situações podem ocorrer: (1) a representação visual é alterada para mostrar apenas a região de interesse selecionada; (2) a região de interesse ocupa o campo de visão principal e o restante do conjunto de dados é mantido em área à parte ou (3) a região de interesse e a visão geral são exibidas concomitantemente.

A técnica *Cone Tree* apresenta uma estrutura visual tridimensional na forma geométrica de um cone, na qual uma hierarquia inteira ou grande parte dela pode ser visualizada sem a necessidade de *scrolling*. Ao mesmo tempo, permitindo a supressão ou exibição de nodos durante a navegação.

Na técnica *TreeMaps* (ou *space-filling*), uma estrutura hierárquica de diretórios é representada a partir da subdivisão sucessiva do espaço de tela. Onde cada subespaço representa um diretório, sendo subdividido em função de seus subdiretórios correspondentes.

Já as técnicas *Hyperbolic Tree* e *Node-Edge Diagram Layout* apresentam a estrutura de uma árvore na forma de um plano hiperbólico. Sendo essa última, uma mistura das técnicas *Hyperbolic Tree* e *space-filling*.

Capítulo 3

Topic Maps

Topic Maps é um padrão ISO³ (norma ISO 13250) que permite a construção de estruturas de conhecimento navegáveis sobre recursos de informação, os quais podem ser de qualquer natureza (arquivos, documentos, repositórios etc.) [1][5][6]. Segundo o modelo do *Topic Maps*, essas estruturas de conhecimento são representadas na forma de arranjos topológicos.

Seus principais conceitos são: tópicos, associações (entre tópicos) e ocorrências (recursos relacionados aos tópicos) – esses três conceitos são conhecidos como *The TAO of Topic Maps*, “T” de *Topic*, “A” de *Association* e “O” de *Occurrence*. Uma introdução ao *Topic Maps* pode ser feita com uma analogia de seus principais conceitos aos antigos e conhecidos índices de livros. Segundo Peeper [6]:

- Cada entrada do índice corresponde a um tópico;
- A ação ver (*see*) corresponde aos nomes variantes para um mesmo tópico;
- Os números das páginas correspondem às ocorrências;
- E o ver também (*see also*) corresponde ao que seriam as associações;

A grande vantagem de *Topic Maps* está na maneira estruturada como esta técnica representa as informações, o que possibilita uma interação muito mais intuitiva que os antigos índices de livros, sendo inclusive, muitas vezes comparável ao *GPS*⁴, pelo fato de guiar seu utilizador sobre o espaço informacional.

Sua arquitetura objetiva a captura do conhecimento e sua visualização de forma estruturada. Para isso, o *Topic Maps* faz uma separação entre os recursos e o mapa de tópicos, o que resulta em dois domínios: um de tópicos e outro de recursos. Essa separação permite que as informações sejam representadas nos mapas de tópicos independentes dos recursos. Possibilitando, dessa maneira, navegar sobre a informação sem que haja, para isso, a necessidade de alterar ou

³ *International Organization for Standardization*. Entidade não-governamental criada em 1947 e com sede em Genebra - Suíça. Seu objetivo é promover, no mundo, o desenvolvimento da normalização e de atividades relacionadas, com a intenção de facilitar o intercâmbio internacional de bens e serviços e de desenvolver a cooperação nas esferas intelectual, científica, tecnológica e de atividade econômica.

⁴ *Global Positioning System*. Permite que uma pessoa, utilizando um aparelho localizador, saiba onde se encontra por meio de coordenadas geográficas (latitude, longitude e altitude).

manipular os recursos, criando um nível de abstração sobre os recursos, como um mapa virtual – um mapa virtual de tópicos informacionais.

A Figura 6 ilustra bem a separação de domínios mencionada acima. Num primeiro momento, observa-se um conjunto de confusas relações semânticas entre dois repositórios hipotéticos (Figura 6a). Posteriormente, com a aplicação do *Topic Maps*, essas mesmas relações semânticas são representadas em uma meta-camada, na forma de estruturas de conhecimento, externa aos repositórios – os recursos (Figura 6b). Ao longo deste capítulo são fornecidos mais detalhes sobre os potenciais benefícios obtidos com essa separação de domínios.

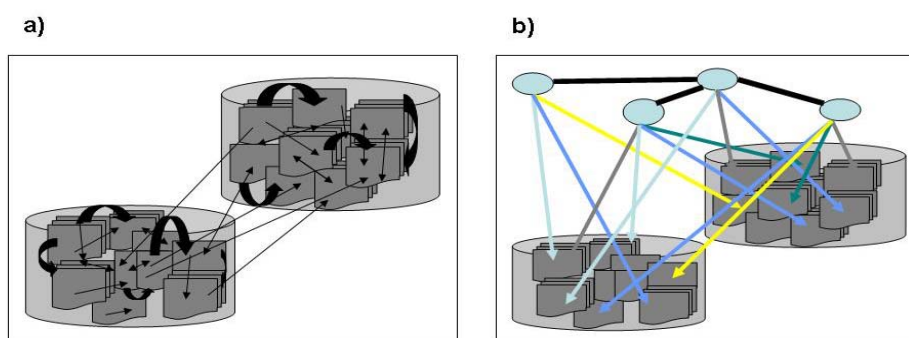


Figura 6. Relacionando recursos de informação. a) Sem *Topic Maps*. b) Com *Topic Maps*.

3.1 Histórico

Os primeiros trabalhos sobre o conceito *Topic Maps* datam do princípio de 1991. Buscando encontrar possíveis soluções para inconsistências nas documentações e publicações sobre seus sistemas computacionais, um grupo de fornecedores de sistemas *UNIX*, denominado *Devenport Group*, decidiu dar início ao desenvolvimento do então padrão.

Mais tarde, esse grupo seria dividido em outros dois: o *DocBook* e o *CAPH*. Criado em 1993, com a participação de *Fred Dalrymple*, *Michel Biezunski*, *Wayne Wholer*, dentre outros, especificamente o *CAPH* (*Conventions Application of HyTime*), prosseguiu com o desenvolvimento do que mais tarde viria a se tornar o *Topic Maps* [16].

A primeira tentativa de criação do padrão foi realizada em 1996, quando o padrão ainda não tinha o nome atual, sendo chamado de *Topic Maps Navigation*. A tentativa foi bem sucedida e o padrão foi aceito pelo grupo *ISO SGML* (*ISO/IEC 13250*). Desde esse primeiro momento, o padrão vinha passando por seguidas modificações, quando finalmente, em 1999, foi concluído.

Tendo como editores *Michael Biezunski*, *Martin Bryan* e *Steve Newcomb*, o *Topic Maps* teve sua publicação efetuada no início do ano 2000. O resultado final foi um compromisso que se pensa ser de extremo poder e flexibilidade por um lado, e de uma semântica suficientemente bem definida pelo outro [5].

Atualmente, o *Topic Maps* está voltado para aplicações *web*. Um consórcio denominado *TopicMaps.org*⁵ vem trabalhando no desenvolvimento da *XTM* (*XML Topic Maps*). Publicada em 2001, a diferença básica da *XTM* à *ISO 13250* é a notação *XML*, bem menos complexa que a *SGML*.

⁵ Disponível em: <<http://www.topicmaps.org>>.

3.2 Principais Conceitos

Nesta seção serão abordados os principais conceitos do padrão *Topic Maps*, levando-se em consideração sua relevância diante da abordagem proposta neste trabalho.

3.2.1 Tópicos

A principal entidade do *Topic Maps*, um tópico representa uma idéia ou assunto, que pode ser qualquer coisa, por exemplo: pessoa, entidade, conceito etc. [6].

Basicamente, os tópicos têm três características: possuem nomes, ligam-se a ocorrências e desempenham funções nas associações [6]. Mais adiante neste capítulo, poderá-se ver que a atribuição dessas características está limitada ao contexto no qual o tópico estiver inserido. Será visto, por exemplo, que um tópico pode ter mais de um nome, a depender do contexto de referência. E que da mesma forma, isso também se aplica às ocorrências e às associações. *Topic Maps* modela essa idéia de contexto através do conceito de escopo (*scope*).

Quando se mencionam nomes de tópicos, refere-se aos *topic names*. Um *topic name* é composto por: *base name*, *display name* e *sort name*. Sendo o *base name* sempre requerido, enquanto os outros dois são, apenas, opcionais [6]. Vale ressaltar que, dentro de um mesmo contexto, dois tópicos quaisquer não podem ter o mesmo nome, obviamente. Para tratar isso, *Topic Maps* define o conceito de *merge* (fusão), onde dois tópicos podem ser fundidos para formar um só, contendo as características de ambos [6].

Além de nomes, associações e ocorrências, um tópico pode ainda pertencer a uma ou mais categorias, definidas como *topic types*, que também são tópicos. Como exemplo, *UPE* poderia ser um tópico do tipo (*topic type*) *Universidade*, que, por sua vez, seria um outro tópico do mapa de tópicos. Em outras palavras, *topic types* representam o típico relacionamento classe-instância [6]. Voltando ao exemplo anterior, *Universidade* seria a classe e *UPE* a instância.

3.2.2 Ocorrências

Um tópico pode estar ligado a um ou mais recursos de informação os quais sejam, de alguma forma, relevantes para o mesmo. À ligação de um tópico a um recurso, denomina-se *occurrence* (ocorrência) – ocorrência do tópico [6].

O grande potencial do *Topic Maps* está, justamente, na separação realizada entre o mapa de tópicos e os recursos de informação sobre os quais os mapas são criados. Com isso, é possível que a um mesmo tópico sejam relacionados vários e diferentes recursos e, da mesma forma, um mesmo recurso também pode ser relacionado a vários e diferentes tópicos, já que, tópicos e recursos pertencem a domínios distintos. A Figura 7 ilustra bem essa separação de domínios. Na parte superior estão representados os tópicos, enquanto a parte inferior ilustra um domínio de recursos com os quais os tópicos matêm algumas ligações, através das ocorrências.

A separação de domínios faz do *Topic Maps* um modelo centrado na informação, e não, nos recursos. Sua estrutura voltada para conceitos, representados pelos tópicos – o elemento fundamental do *Topic Maps* – faz com que os recursos existam em função dos conceitos (os tópicos). Já os tópicos (conceitos), independem dos recursos para existirem. De maneira que, podem existir tópicos sem ocorrências, mas nunca o contrário [1].

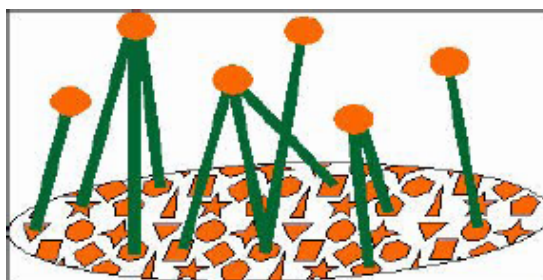


Figura 7. Ocorrências. Separação em dois domínios: um de tópicos e outro de recursos [6].

Da mesma maneira que os tópicos, as ocorrências também podem pertencer a diferentes tipos, denominados *occurrence role types* (tipo do papel da ocorrência), que da mesma forma, também são tópicos.

O tipo do papel da ocorrência de um tópico é utilizado para indicar a forma como o recurso contribui informação para o tópico, podendo esta forma ser de qualquer natureza (artigo, monografia, imagem, vídeo, música etc.). Por exemplo, uma ocorrência da presente monografia no tópico *Victor* poderia ter como *occurrence role type* o tópico *Monografia*.

Em resumo, ocorrência é a indicação de um tópico para um recurso informacional, indicado e classificado por um tipo, o *occurrence role type* que, por sua vez, também é um tópico [6].

Uma ocorrência possui ainda outro conceito, o *occurrence role* (papel da ocorrência). Este conceito apresenta significado muito semelhante ao *occurrence role type*. Porém, o papel (indicado pelo atributo *role*) é simplesmente um mnemônico, enquanto o tipo (indicado pelo atributo *type*) é uma referência para um tópico que caracteriza a relevância do papel (*role*) por ele assumido [6].

Muitas vezes o *occurrence role* descreve o mesmo que o *occurrence role type*. Contudo, faz sentido especificar o tipo do papel da ocorrência, pois desse modo, as capacidades do conceito *Topic Maps* podem ser usadas para comunicar mais informação sobre o papel de uma ocorrência [6].

3.2.3 Associações

As características até agora apresentadas permitem, apenas, organizar os recursos de acordo com tópicos, por intermédio das ocorrências. Contudo, um dos aspectos mais interessantes do *Topic Maps* é a capacidade de descrever relacionamentos entre tópicos: as associações.

Uma associação permite relacionar dois ou mais tópicos [6]. É a partir desses relacionamentos que se estruturam os mapas de tópicos (Figura 8).

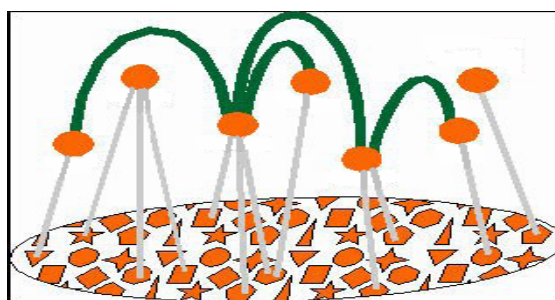


Figura 8. Mapa de tópicos: associações entre tópicos (figura adaptada de [6]).

Assim como os tópicos podem ser agrupados segundo seus tipos e as ocorrências segundo seus papéis, também as associações podem ser agrupadas por tipos. Não obstante, os tipos das associações (*association types*) também são definidos em termos de tópicos [6].

Numa associação devem ser identificados o tipo da associação (*association type*) e a função (*role*) desempenhada por cada tópico. Por exemplo, numa associação entre *Fernando Buarque* e *Victor*, teria-se o *association type* *Orientação*, onde *Fernando Buarque* desempenharia a função (*role*) de *Orientador* (ou *Professor*), e *Victor*, a função de *Orientando* (ou *Aluno*).

Contudo, diferentemente das relações matemáticas, as associações são multidirecionais. Numa relação matemática, se *A* está relacionado com *B*, *B* pode não estar relacionado com *A*. Já em mapas de tópicos, se *A* estiver relacionado com *B*, então *B* deve, por definição, estar relacionado com *A* [6]. Com isso, pelo fato de uma associação entre tópicos ser multidirecional, o conceito de *association role* assume grande importância. Pois, saber apenas o tipo de uma associação não é suficiente, agora é necessário identificar, também, o papel (*role*) desempenhado por cada um dos tópicos da associação. No exemplo anterior, indicar apenas o *association type* *Orientação*, sem identificar as funções desempenhadas (*association roles*) por cada um dos tópicos, tornaria impossível saber qual dos tópicos seria o *Professor* e qual seria o *Aluno*, por exemplo.

Note-se que o conceito de *topic types* pode ser considerado como uma forma de *association type*. Por exemplo, indicar que o tópico *Victor* é do tipo (*topic type*) *Aluno*, pode ser expresso como uma associação do tipo (*association type*) *classe-instância*. Em que, o tópico *Aluno* desempenharia a função (*role*) de *classe* e o tópico *Victor* a função (*role*) *instância* [6].

A princípio, uma associação pode parecer semelhante a uma referência cruzada comum. Contudo, enquanto numa referência cruzada os componentes são os próprios recursos da informação, uma associação possui tópicos como seus componentes, que por pertencerem a um domínio diferente do domínio dos recursos, tornam as associações completamente independentes de todo e quaisquer recursos aos quais seus componentes (os tópicos) estiverem ligados [6], como ilustrado na Figura 9, abaixo.

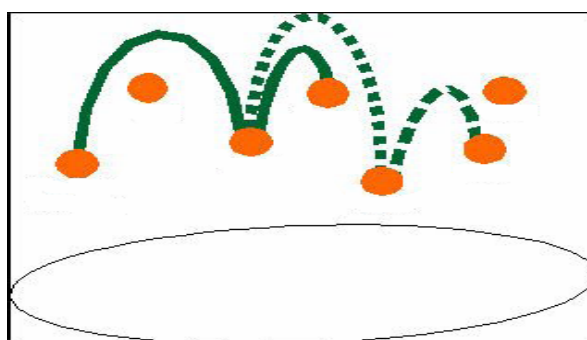


Figura 9. Associações independentem dos recursos (figura adaptada de [6]).

3.2.4 Escopo

Conforme seção 3.2.1, um tópico é constituído de três características principais: nomes, associações e ocorrências. Essas três características juntas são conhecidas como *topic characteristics* [6].

Contanto, as atribuições de características a um tópico estão sempre limitadas ao contexto no qual este tópico estiver inserido [6]. Por exemplo, se novamente for mencionado o tópico *Victor*, concomitantemente virá o conceito “o aluno”. Isto porque, neste caso, o contexto é o mesmo dos exemplos utilizados acima. Supondo-se considerar o contexto “Casa”, *Victor* poderia passar a ser “o filho”, por exemplo.

Humanos têm essa habilidade de identificar e delimitar contextos. Computadores, contudo, ainda não possuem essa especial habilidade. Pensando nisso, *Topic Maps* considera que quaisquer características atribuídas a um tópico, seja ela um nome, uma ocorrência ou uma associação, são válidas apenas no contexto no qual o tópico estava inserido no momento da atribuição. A esse “limite de validade” *Topic Maps* chama *scope* (escopo) [6].

Topic Maps define *scope* em termos de temas, que por sua vez, são definidos como membros de um conjunto de tópicos especialmente utilizados para especificar escopos [6].

Inicialmente, pode-se pensar que o conceito de *scope* serve apenas para remover ambigüidades. Contudo, outra característica derivada desse conceito é ainda mais importante, a navegabilidade. Esse conceito permite, por exemplo, que a visão do mapa de tópicos possa ser dinamicamente alterada segundo seus escopos, uma vez que, as características dos tópicos (nomes, ocorrências e associações) estão diretamente limitadas a estes. Assim, dois tópicos podem estar associados em um escopo, e em outro, não, por exemplo. A Figura 10 ilustra bem os resultados obtidos com a aplicação de *scope* a um mapa de tópicos.

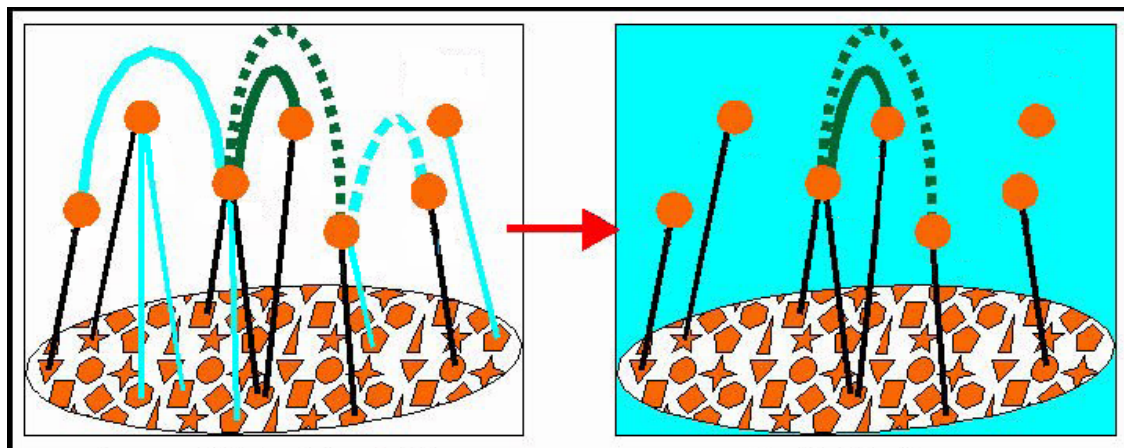


Figura 10. Efeito de se aplicar escopo (figura adaptada de [6]).

3.3 Visualização de *Topic Maps*

O *Topic Maps* não define um modo de visualização específico, ficando livre a cada aplicação sua especificação. *Kamiya* [1] apresenta e discute alguns dentre muitos possíveis modos de visualização que podem ser apresentados para o *Topic Maps*. A seguir, são fornecidos dois exemplos reais de aplicações de *Topic Maps* [1], os quais remetem às técnicas de visualização de informações observadas no capítulo anterior deste trabalho.

O software *Ontopia Omnigator*⁶, uma tecnologia desenvolvida, justamente, para demonstrar o poder do *Topic Maps*, como se pode observar na Figura 11, sugere a técnica de visualização de informações *Node-Edge Diagram Layout* [15].

⁶ Tecnologia desenvolvida para demonstrar o poder do *Topic Maps*. Disponível em: <<http://www.ontopia.net/omnigator>>.

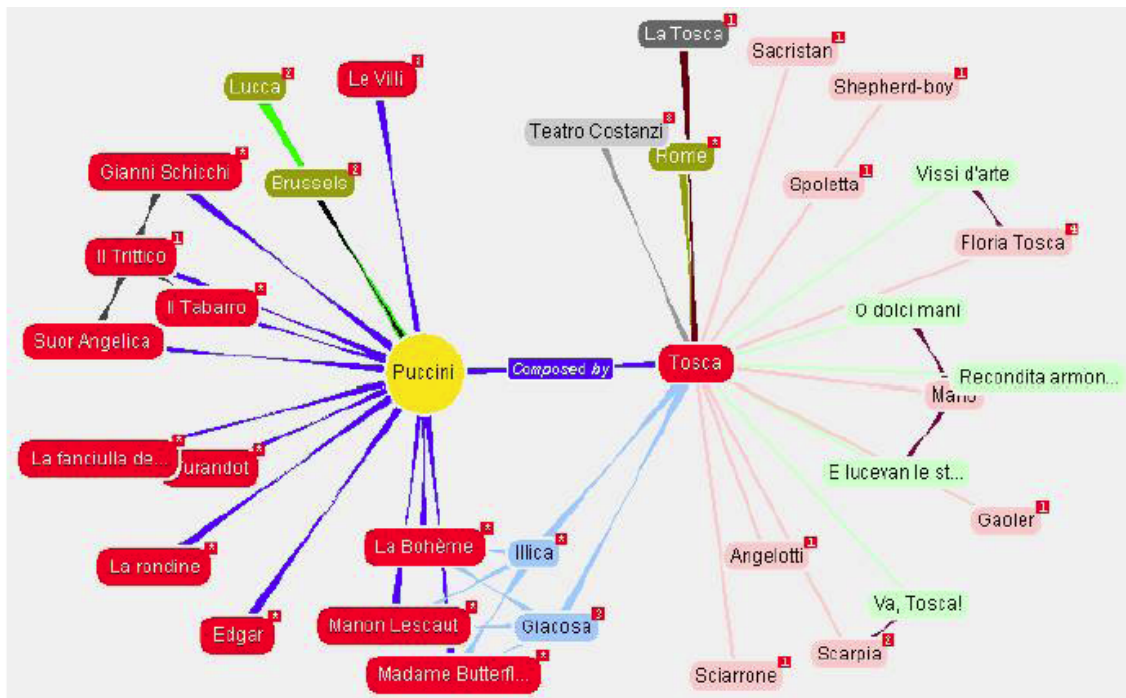


Figura 11. *Topic Maps no Ontopia Omnigator.*

Já a *Highwire Press*⁷, outro exemplo de aplicação do *Topic Maps*, preferiu o uso da técnica de visualização *Hyperbolic Tree* [13], como ilustrado abaixo, na Figura 12:

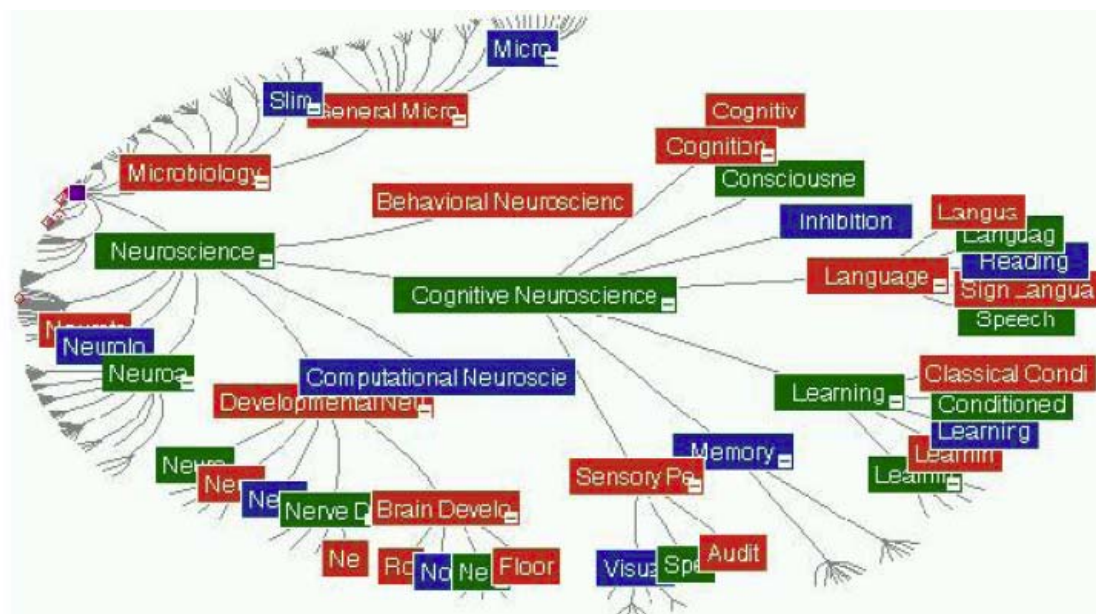


Figura 12. *Topic Maps no web site da Highwire Press.*

Desejavelmente, o modo de visualização para o *Topic Maps* deve considerar o poder de navegabilidade sobre a informação – possibilitado através das associações entre os tópicos. Bem como os benefícios da separação de domínios realizada entre o mapa de tópicos e os recursos.

⁷ Divisão das bibliotecas de Universidade de Stanford. Disponível em: <<http://highwire.stanford.edu/>>.

3.4 Resumo

Topic Maps cria abstrações sobre os recursos de informação, mantendo-os inalterados, facilitando assim, à criação dinâmica de estruturas de conhecimento sobre os mesmos. Segundo o seu modelo, essas estruturas de conhecimento são representadas na forma de arranjos topológicos, sendo seu principal conceito o tópico.

Um tópico representa uma idéia ou assunto qualquer, podendo ser qualquer coisa (pessoa, entidade, conceito etc.) e, basicamente, têm três características: possuem nomes, ligam-se a ocorrências e desempenham funções nas associações.

Ocorrência é a ligação de um tópico a um recurso de informação. Enquanto associações descrevem os relacionamentos entre os tópicos. Esses dois conceitos, juntamente com o conceito de tópico definem os principais conceitos do *Topic Maps*, sendo conhecidos como *The TAO of Topic Maps*, “T” de *Topic*, “A” de *Association* e “O” de *Occurrence*.

Outro importante e interessante conceito do *Topic Maps* é o conceito de escopo (*scope*). Escopo nada mais é que um tópico utilizado para delimitar um determinado contexto (ou tema) no mapa de tópicos. Na prática, um tópico pode possuir diferentes características (nome, ocorrências e associações) sob diferentes escopos, o que possibilita diferentes visualizações sobre um mesmo mapa de tópicos a partir da simples mudança de escopo.

Isso é possível devido à maneira estruturada como o *Topic Maps* representa as informações, ao realizar uma separação entre o domínio dos relacionamentos semânticos do mapa de tópicos e o domínio dos recursos de informações. O fato de um tópico poder estar ligado a um ou mais recursos de informação e, da mesma forma, um mesmo recurso também pode estar relacionado com diferentes tópicos, faz do *Topic Maps* um modelo centrado na informação, e não nos recursos.

Capítulo 4

TME

Os sistemas operacionais (SOs) existentes hoje no mercado utilizam um sistema de arquivos hierárquico, onde seus arquivos estão organizados segundo uma estrutura de árvore de diretórios. O mais conhecido e popular dentre todos os SOs, é o sistema operacional *Windows*, da *Microsoft*. A Figura 13 ilustra o utilitário gráfico para gerenciamento de arquivos desse SO, o *Windows Explorer*. Pode-se observar especificamente à esquerda na figura, que o *Windows Explorer* preserva a organização hierárquica de seus arquivos, permitindo ao usuário navegar por toda a estrutura de diretórios de seu sistema.

Contudo, talvez a grande e maior limitação do utilitário esteja, justamente, na dependência com a localização física imposta a seus arquivos. Isto significa que, com a ressalva de um utilitário de busca que permite encontrar arquivos a partir de nomes, datas e tipos (dentre outros), para localizar um arquivo é necessário saber qual a sua localização física diante de toda a árvore de diretórios do sistema de arquivos. A possibilidade de configurar os nomes desses diretórios pode amenizar um pouco esse processo de localização, ao passo que atribuir nomes relacionados ao conteúdo dos arquivos torna a busca mais intuitiva. Contudo, com uma análise do ponto de vista prático, a dependência ao endereçamento físico se traduz, ainda, em uma grande limitação. Como exemplo, o simples fato de procurar por um arquivo (ou por um conjunto deles) segundo uma organização mental (por parte do usuário) que seja diferente da organização física onde o arquivo esteja localizado, hoje é impraticável.

Então, buscando solucionar as limitações de localização física impostas pelo sistema de arquivos dos SOs atuais, este capítulo apresenta o *TME (Topic Maps Explorer)*, o protótipo de um *software* que propõe uma nova abordagem para a organização e o acesso aos arquivos pessoais a partir da aplicação dos conceitos do padrão *Topic Maps (ISO 13250)*.

O capítulo inicia explicando a forma como o *TME* foi idealizado, seguida da descrição de todo o protótipo, contemplando suas características, detalhes de implementação e funcionalidades.

4.1 Idealização

O ideal maior da aplicação do padrão *Topic Maps* ao gerenciamento de arquivos pessoais surgiu da possibilidade provida por este padrão de construir estruturas de conhecimento sobre recursos de informação (no nosso caso, sobre arquivos pessoais). Sobretudo, por essas estruturas de conhecimento serem definidas em um domínio diferente do domínio dos arquivos.

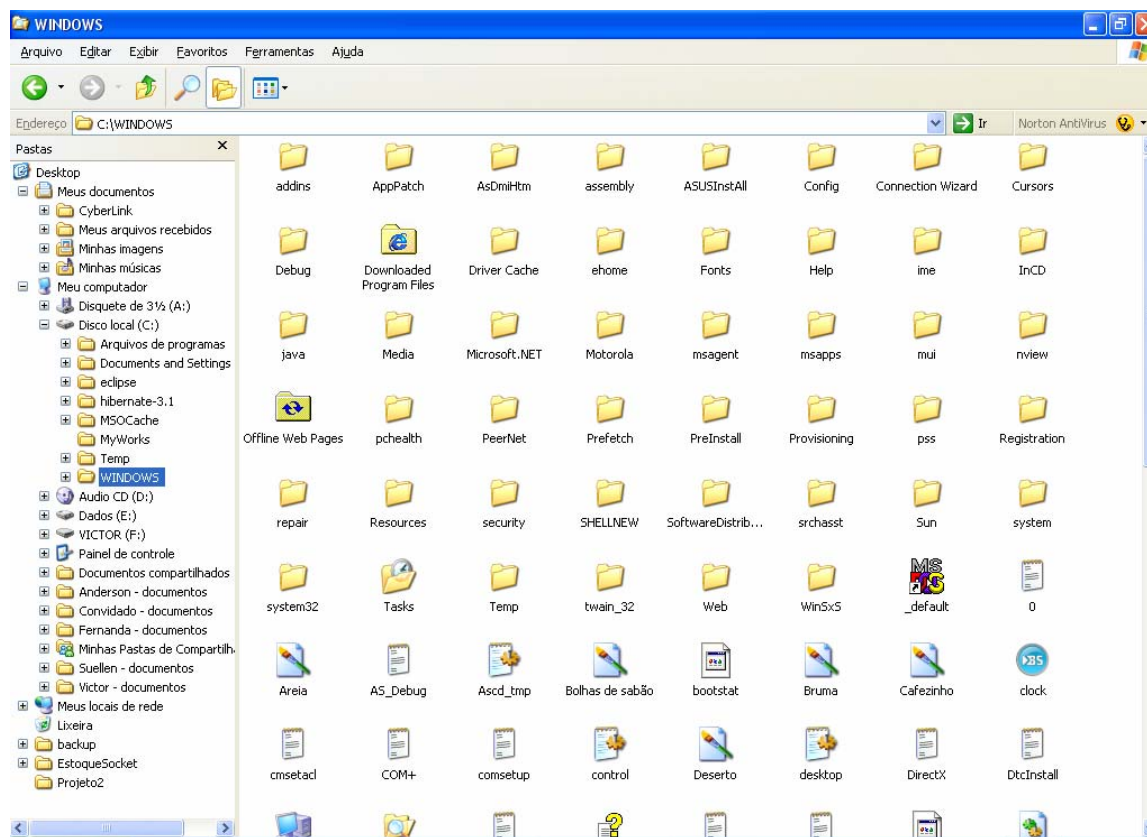


Figura 13. *Windows Explorer* – utilitário gráfico para o gerenciamento de arquivos do SO da *Microsoft*.

Como pôde ser visto no capítulo anterior, o *Topic Maps* faz uma separação entre o domínio semântico dos mapas de tópicos e o domínio dos recursos, de forma que as limitações de dependência quanto à localização física dos arquivos encontradas no sistema de arquivos dos SOs atuais, e introduzidas no início deste capítulo, podem ser totalmente abstraídas pelo padrão.

Se antes o acesso e a organização dos arquivos pessoais estavam limitados às estruturas físico-hierárquicas das árvores de diretórios, agora com a aplicação do *Topic Maps*, essas tarefas poderiam ser realizadas em uma meta-camada abstraída dos arquivos, representada na forma de mapas de tópicos informacionais. Fazendo uma analogia, os tópicos tomariam o lugar dos diretórios e junto às associações, que representariam as relações de dependência hierárquica da árvore de diretórios (só que, bem mais flexíveis e nem um pouco dependentes), carregariam o conteúdo semântico dos relacionamentos entre os arquivos. Enquanto as ocorrências funcionariam como o mecanismo para endereçar os arquivos, a partir dos tópicos.

Essa arquitetura pode ser observada como uma forma de endereçar os arquivos a partir dos seus conteúdos. Onde, o “conteúdo-endereço” estaria representado nos mapas de tópicos, através dos tópicos e das associações entre os tópicos. De forma que um arquivo não teria necessariamente, apenas um endereço definitivo, pois devido à separação de domínios realizada pelo *Topic Maps*, a um mesmo arquivo poderiam estar relacionados vários e diferentes tópicos, podendo-se chegar a esse mesmo arquivo a partir de qualquer um dos tópicos a ele relacionado. Logo, os “conteúdos-endereços” dos arquivos seriam todos os diferentes e possíveis “caminhos” de tópicos que pudessem ser percorridos para se chegar a eles (os arquivos).

Se bem observada, esta abordagem pode ainda, estender-se a um mecanismo de busca. Pois, como para cada tópico criado sobre um arquivo haveria uma ocorrência que relacionaria

cada um desses tópicos a este arquivo, navegar sobre o mapa de tópicos (através das associações entre os tópicos) seria análogo a utilizar um mecanismo de busca, sendo que aqui as palavras-chaves da busca seriam os próprios tópicos. Em resumo, já que a cada tópico, possivelmente, haverá ocorrências para diferentes arquivos, ao se navegar sobre a estrutura dos mapas de tópicos (através das associações entre os tópicos), o “caminho” de tópicos percorridos pode funcionar como um filtro, em que apenas as ocorrências aos arquivos correspondentes e relacionados a todos os tópicos pertencentes ao “caminho” seriam filtradas.

4.2 Características

O *TME* foi desenvolvido utilizando, essencialmente, tecnologia Java [17][18][19]. A escolha dessa tecnologia se deveu entre outras, à característica de portabilidade que a Java possui, permitindo que uma mesma aplicação possa ser executada sob diferentes plataformas de operação, sem que haja, para isso, a necessidade de qualquer tipo de alteração em seu código fonte. Em outras palavras, o *TME* pode ser executado em *Linux* da mesma forma em que é executado no *Windows*, por exemplo.

Essa característica de portabilidade é possível graças ao conceito de máquina virtual. Onde o compilador Java não gera instruções específicas a uma plataforma, mas sim, um programa em um código intermediário, denominado *bytecode*, que pode ser descrito como uma linguagem destinada a um processador virtual específico. O código Java compilado é, então, executado por um interpretador de *bytecodes*, a *JVM – Java Virtual Machine*, que é um emulador de processador e pode ser instalado sob qualquer plataforma operacional.

Também, por se tratar de uma linguagem de programação orientada a objetos, Java facilita o desenvolvimento de sistemas modulares ou que sejam baseados em componentes, naturalmente. O desenvolvimento de *software* baseado em componentes, ou DBC, é definido como o desenvolvimento de sistemas através da integração planejada de componentes de *software* reutilizáveis [20]. Fora a possibilidade de reutilização de componentes, a manutenção e eventuais alterações em um sistema modular se torna muito mais simples, na medida em que as alterações são feitas no nível dos módulos, não no sistema como um todo. Ou seja, alterações em um módulo não traduzem, necessariamente, alterações nos demais módulos do sistema, tampouco no sistema como um todo. Especificamente o *TME* é constituído de alguns módulos.

Como já dissemos, o *TME* possui um interface de visualização gráfica tridimensional. Esta interface se traduz em um módulo do sistema que é responsável por representar as estruturas dos mapas de tópicos criados sobre os arquivos em um ambiente de visualização e interação tridimensional. Esse módulo também foi desenvolvido utilizando tecnologia Java, através biblioteca Java 3D [21], que consiste de uma hierarquia de classes Java destinadas ao desenvolvimento de sistemas gráficos tridimensionais. Porém, a biblioteca Java 3D não acompanha a instalação convencional da *JVM*. Assim, além da *JVM* para executar o *TME*, também é necessário ter instalada a Java 3D.

Dadas as considerações descritas acima, podem-se atribuir três características principais para o *TME*:

- Portável. Na medida em que pode ser executado sob diferentes plataformas operacionais;
- Modular. Por ter sua implementação computacional baseada em componentes;
- Interface de visualização gráfica tridimensional;

4.3 O Modelo Computacional para o *Topic Maps*

O modelo computacional implementado para o *Topic Maps* contempla vários dentre os seus principais conceitos, como: *Topic*, *Association*, *Occurrence* e *Scope*. Constituindo um componente do *TME*, a definição desse modelo foi realizada de forma a contemplar os conceitos do *Topic Maps* que fossem relevantes para o objetivo maior do *TME* – aplicar o padrão *Topic Maps* para o gerenciamento de arquivos pessoais.

Na Figura 14 está ilustrado o diagrama de classes *UML*⁸ do modelo. Nesta seção, é realizada uma análise detalhada da implementação computacional desses e de outros conceitos relacionados. Para tornar este processo mais intuitivo e significativo, constantes analogias são feitas às originais especificações do *Topic Maps*.

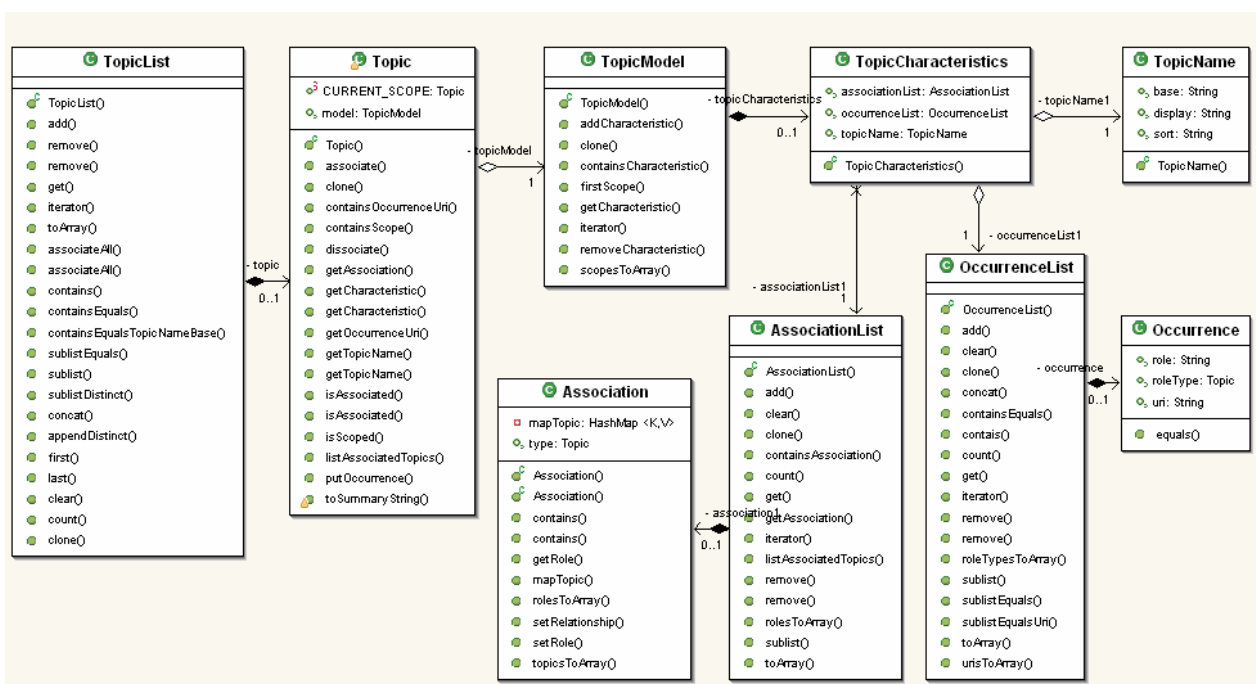


Figura 14. Diagrama de classes *UML* do modelo computacional para o *Topic Maps*.

4.3.1 *Scope*

Da sua definição no *Topic Maps*, um escopo é um tópico do mapa de tópicos como outro qualquer, com o diferencial de também ser utilizado para delimitar as características dos demais tópicos de um determinado mapa de tópicos. De forma que para diferentes escopos, um tópico, possivelmente, possa apresentar diferentes características. O conceito de *Scope* se encontra modelado no *TME* através das classes *TopicModel* e *TopicCharacteristics*.

Quando se fala em características de um tópico, entende-se serem: seu nome (*topic name*), mais as suas associações com outros tópicos (*associations*), mais as suas ocorrências nos recursos de informação (*occurrences*). Assim, a classe *TopicCharacteristics* é constituída por três atributos: um objeto da classe *TopicName*, uma lista de objetos da classe *Association* e outra lista de objetos da classe *Occurrence*.

⁸ A *Unified Modeling Language (UML)* é uma linguagem de modelagem que auxilia no desenvolvimento de aplicativos computacionais, que sejam construídos em uma linguagem de programação orientada a objetos, a visualizar a modelagem e a comunicação entre objetos.

Já a classe *TopicModel* é utilizada, justamente, para realizar o mapeamento de objetos da classe *TopicCharacteristics* sob os devidos “tópicos-escopos” (tópicos que estejam sendo utilizados para representar o conceito *Scope* do *Topic Maps*), da classe *Topic*. Ou seja, dado um “tópico-escopo”, o *TopicModel* pode mapear e recuperar um respectivo objeto *TopicCharacteristics*, representando que aquelas características (*TopicCharacteristics*) estão definidas e delimitadas para ele, o “tópico-escopo”.

Para isso, a classe *TopicModel* contém como atributo um objeto da classe *HashTable*, uma classe nativa de Java que provê o mapeamento de objetos, basicamente. Em resumo, um objeto *TopicModel* pode ser observado como uma tabela de *TopicCharacteristics*, cujos índices são objetos da classe *Topic* (novamente, que estejam sendo utilizados para representar o conceito *Scope* do *Topic Maps*).

Assim, para caracterizar o mapeamento de um objeto *TopicCharacteristics* em um determinado “tópico-escopo”, deve-se invocar o método *addCharacteristics* (*Topic scope*, *TopicCharacteristics characteristics*) da classe *TopicModel*, que recebe como parâmetros um objeto *TopicCharacteristics* e o objeto *Topic*, que desempenha a função de escopo no qual devem ser inseridas as características e, portanto, delimitadas. Então, para mapear esse mesmo *TopicCharacteristics*, basta invocar outro método da classe *TopicModel*, o método *getCharacteristics* (*Topic scope*), que recebe como parâmetro o “tópico-escopo” e retorna o respectivo *TopicCharacteristics*.

É importante salientar, que não existe uma classe *Scope* para representar o homônimo conceito do *Topic Maps*, em vez disso, o conceito está inferido no modelo através das classes *TopicModel* e *TopicCharacteristics*. Isso pode ser observado na Figura 15, onde cada objeto da classe *Topic* tem um *TopicModel* como atributo que é responsável por armazenar e mapear todas as características (*TopicCharacteristics*) de um *Topic*, para os diferentes *Scopes*. Com isso, pode-se fácil e dinamicamente, modificar e visualizar os diferentes conjuntos de características de um *Topic*, apenas, pelo mapeamento de suas características, realizado por seu atributo *TopicModel*.

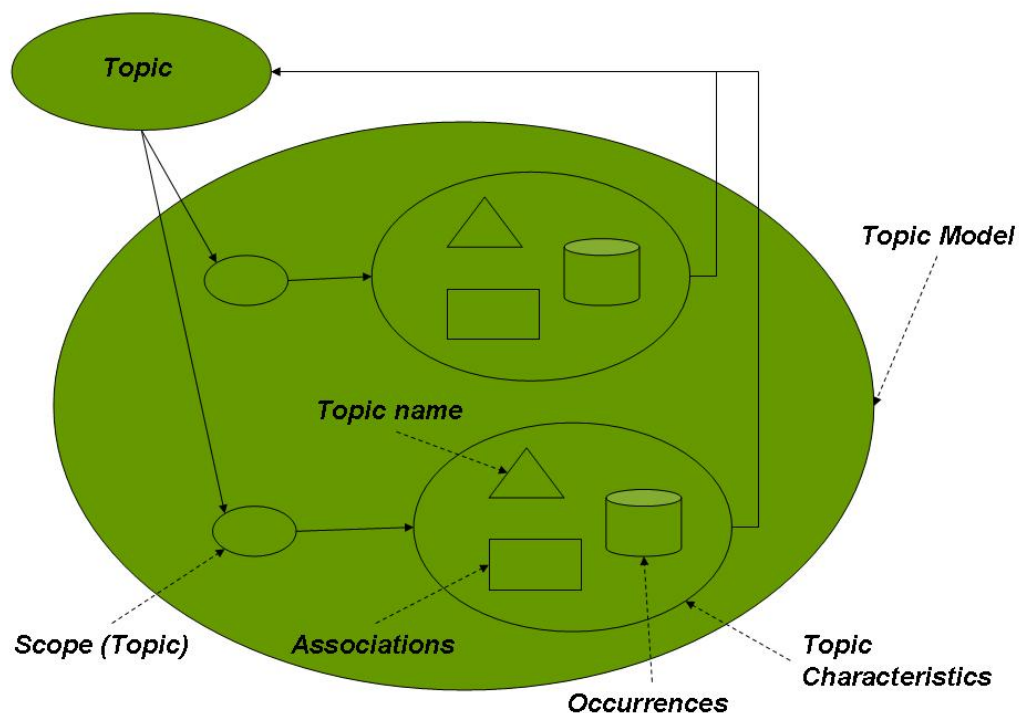


Figura 15. Modelo gráfico da implementação do conceito *Scope*.

4.3.2 Topic

Referenciado as especificações do padrão *Topic Maps*, o modelo computacional de um tópico seria constituído dos atributos: *topic types*, *topic names*, *associations* e *occurrences*. Sendo cada um desses atributos implementado através de listas. Como exemplo, o atributo *topic types*, seria uma lista de objetos *Topic*, em que estes objetos *Topic* representariam os *topic types* de um outro *Topic* o qual pertencer a lista. Contudo, como já adiantado na subseção 3.2.4, devido ao conceito de escopo, as características de um *Topic* são armazenadas através da classe *TopicModel*. Com isso, a classe *Topic* possui apenas dois atributos: uma lista de objetos da mesma classe *Topic*, representando seus *topic types*, e um objeto da classe *TopicModel*.

Também devido ao conceito de *Scope*, a classe *Topic* define um outro atributo, o *CURRENT_SCOPE*. Este atributo nada mais é que outra instância da classe *Topic*, só que desta vez é definido como um atributo estático e serve para representar o atual escopo em que o *Topic* esteja sendo apresentado ou visualizado. Sua função é desempenhada de forma que: mapeia-se no atributo *TopicModel* o *TopicCharacteristics* referente ao atual escopo (o *CURRENT_SCOPE*).

O fato de o atributo *CURRENT_SCOPE* ser definido como estático porque, em Java, um atributo definido como estático faz com que este atributo seja o mesmo para todas as instâncias daquela classe. Com isso, visualizar um objeto *Topic* em um determinado escopo faz com que todas as demais instâncias da classe *Topic*, contidas no mapa de tópicos, também sejam apresentadas segundo o mesmo escopo, automaticamente.

4.3.3 Occurrence

Fiel à especificação do padrão *Topic Maps*, a implementação da classe *Occurrence* é composta de três atributos: *role*, *occurrenceRoleType* e *URI*. Como define o padrão, o atributo *occurrenceRoleType* é um tópico, assim sendo, um objeto da classe *Topic*. Especificamente no *TME*, as ocorrências sempre serão a arquivos pessoais, logo, o atributo *URI* de um objeto *Occurrence* sempre será o caminho do arquivo na árvore de diretórios do sistema de arquivos.

4.3.4 Association

Como vimos no Capítulo 3, numa associação entre dois tópicos devem ser identificados o tipo da associação (*association type*) e a função (*role*) desempenhada por cada um dos tópicos.

Para a implementação da classe *Association*, tomou-se o cuidado em preservar a característica de bidirecionalidade definida pelo conceito, que diz: se um tópico *A* for associado a um tópico *B*, então, por definição, o tópico *B* também deve ser associado ao tópico *A*.

Assim, a classe *Association* é formada pelos atributos: *associationType* e *mapTopics*. O atributo *associationType* é um tópico (classe *Topic*), como o padrão define. Já o atributo *mapTopics* pertence à classe *HashMap* nativa de Java que, salve algumas pequenas diferenças, assemelha-se à classe *HashTable*, sobre a qual já foi falado quando a classe *TopicModel* foi apresentada.

Basicamente, o atributo *mapTopics* faz o mapeamento dos papéis (*roles*) desempenhados pelos tópicos na associação, sob os próprios tópicos. O mapeamento é necessário devido, justamente, à característica bidirecional de uma associação. Num primeiro instante, pensou-se em definir a classe *Association* com atributos específicos como: *topic1*, *role1*, *topic2*, *role2* e *associationType*. Contudo, como uma associação é a mesma para os dois tópicos (bidirecional), como seria possível saber qual seria o *topic1* e qual seria o *topic2*? A Figura 16 ilustra essa impossibilidade de atributos específicos no caso de relacionamento bidirecional. No caso unidirecional seria simples, pois o *tópico A* sempre seria o *topic1* ou o *topic2*. Mas no caso

bidirecional, o mesmo objeto *Association* seria armazenado em cada um dos tópicos, desse modo, ao analisar a associação a partir do *tópico B*, por exemplo, não seria possível determinar se o *tópico B* seria o *topic1* ou o *topic2* da associação. Além do mais, uma associação pode não ser constituída de dois tópicos apenas, mas de dois ou mais tópicos [6].

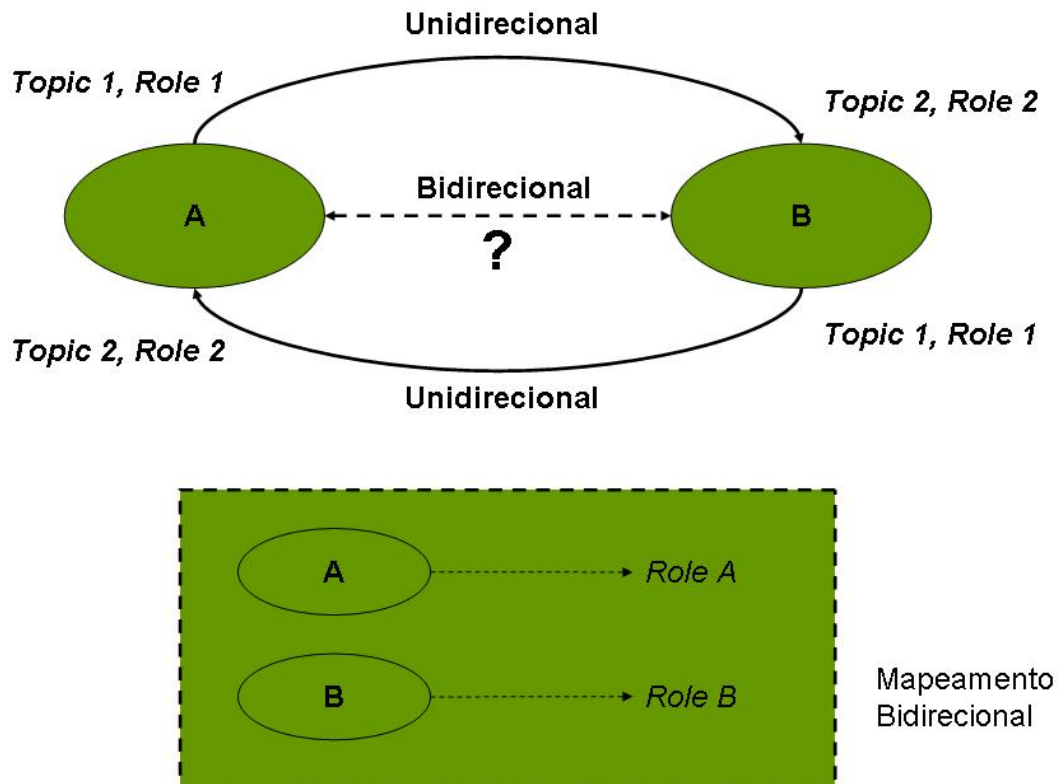


Figura 16. *Association*: relacionamento bidirecional.

4.4 Funcionalidades

O *TME* permite a seu usuário criar ou mapear estruturas de conhecimento sobre arquivos pessoais, que são traduzidas na forma de mapas de tópicos. Representando essas estruturas de mapas de tópicos através de uma interface de visualização gráfica tridimensional, de modo que seu usuário possa navegar sobre a informação construída (ou mapeada), num domínio abstraído e independente da organização físico-hierárquica do sistema arquivos.

Esta seção apresenta algumas das principais funcionalidades do *TME*, explicando o passo-a-passo do funcionamento de cada operação, juntamente a alguns exemplos práticos.

4.4.1 Mapeamento de Arquivos (Manual)

A funcionalidade de mapeamento de arquivo consiste em criar um mapa de tópicos sobre um dado arquivo. De modo que, para cada tópico do mapa de tópicos seja criada uma ocorrência ao dado arquivo.

Do ponto de vista do usuário, essa operação consiste, basicamente, na seleção do arquivo seguida da digitação de uma lista de tópicos informacionais sobre o arquivo selecionado. A Figura 17 contém a tela da interface gráfica do *TME* específica para esta operação.

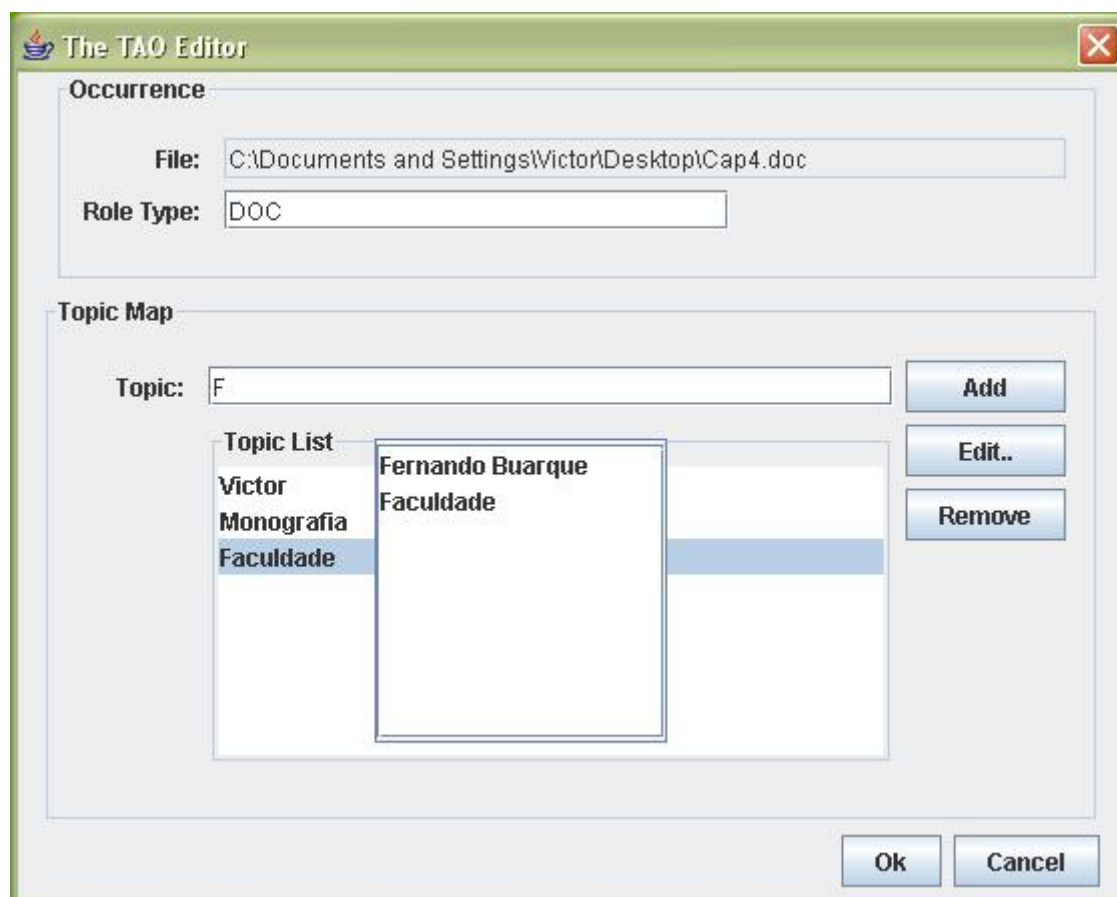


Figura 17. Tela para mapeamento de arquivo do *TME*.

Inicialmente, o usuário seleciona o arquivo sobre o qual quiser criar um mapa de tópicos. Feito isso, o próximo passo é informar qual o *occurrence role type* da ocorrência (imagem, vídeo, artigo, etc.). Visando a diminuição do trabalho para o usuário, o *TME* sempre sugere como *occurrence role type* o próprio tipo do arquivo (*jpeg*, *avi*, *doc*, *ppt*, etc.). Contudo, é aconselhável que os usuários descrevam seus próprios *occurrence role types*, de forma que os arquivos sejam de alguma maneira, unificados por *occurrence role types* genéricos. Como exemplo de *occurrence role type* genérico, pode-se citar um tópico *imagem*, que unificaria os arquivos dos vários tipos de arquivos existentes para imagens (*jpeg*, *bmp*, *gif*, etc.), o que é muito mais simples e intuitivo para o usuário.

Pensando nisso, o ideal seria ainda que essa sugestão automática fosse baseada em operações anteriores. De modo que, se um arquivo do tipo *jpeg* já houvesse sido mapeado antes e o *occurrence role type* para a ocorrência daquele arquivo tivesse sido o tópico *imagem*, ao mapear um próximo arquivo do mesmo tipo *jpeg*, automaticamente fosse sugerido o mesmo tópico *imagem* como *occurrence role type*, por exemplo.

Depois de selecionado o arquivo e definido o *occurrence role type* para sua ocorrência, o passo seguinte constitui-se em listar os tópicos que formarão o mapa de tópicos sobre o arquivo. Neste momento, e em todos os outros que estejam relacionados diretamente a tópicos, o *TME* disponibiliza um recurso de *auto-completar*. Na descrição anterior do próprio *occurrence role type* este recurso de *auto-completar* é, também, disponibilizado.

O recurso de *auto-completar* lista todos os tópicos já existentes no sistema, filtrando-os de acordo com a digitação do usuário. Na realidade, além de facilitar o trabalho para o usuário, este mecanismo tem, também, o intuito de evitar futuras operações de *merge* entre tópicos semelhantes (neste caso, entre tópicos de mesmo nome) no mapa de tópicos. Podemos melhor observar o funcionamento desse recurso na própria Figura 17, quando ao digitar a letra *f* o recurso de *auto-completar* listou dois tópicos: *Faculdade* e *Fernando Buarque*, que já existiam.

Então, após listar todos os tópicos, restaria configurar as associações entre esses tópicos. Especificamente para o propósito do *TME*, não se entendeu que a identificação dos papéis (*roles*) dos tópicos que compõem uma associação acrescenta muito. Da mesma forma para o *association type*. O simples fato de dois tópicos estarem relacionados (por meio de uma associação) já é suficiente. Até porque, os próprios nomes dos tópicos que formam uma associação já possuem alguma semântica implícita ao relacionamento. Além do mais, seria inconveniente para o usuário ter que informar todos os papéis desempenhados por cada um dos tópicos para todo arquivo que fosse mapear.

Assim, após listar todos os tópicos que formarão o mapa de tópicos sobre o arquivo selecionado, o *TME* cria associações entre todos esses tópicos listados, automaticamente. Porém, com *roles* e *association types* nulos.

Também se entendeu desnecessária ou de pouca valia a indicação dos papéis dos recursos numa ocorrência (*occurrence roles*), sendo também configurados sempre como nulos pelo *TME*.

Então, o que a operação de mapeamento de arquivo faz é criar e associar tópicos, quando esses tópicos já não existem no mapa de tópicos, configurando uma ocorrência para o arquivo mapeado em cada um dos tópicos.

Quando um tópico da lista já existe, a ele são apenas acrescentadas as características (associações e ocorrências) que já não contém. Isto é, se um tópico já estiver associado a outro da lista, uma nova associação não precisa ser criada entre eles, até porque num mesmo escopo não pode haver mais de uma associação entre dois tópicos.

4.4.2 Mapeamento de Diretórios (Automático)

O que esta funcionalidade faz é aproveitar a organização hierárquica da árvore de diretórios do sistema de arquivos para criar um mapa de tópicos sobre todos os arquivos que estiverem abaixo de um diretório específico, até o nível mais baixo e de forma automática.

A operação é realizada da seguinte maneira:

1. O usuário seleciona um diretório-raiz a ser mapeado;
2. Um tópico com o nome do diretório selecionado é criado, assim como, para cada um dos seus subdiretórios, até o nível mais baixo, de forma recursiva;
3. São criadas associações entre os tópicos-diretórios criados, segundo a organização hierárquica da árvore de diretórios. De forma que, cada tópico-diretório é associado a todos os tópicos-diretórios imediatamente abaixo na hierarquia;
4. Para cada arquivo, uma ocorrência é criada em cada um dos tópicos-diretórios que estejam acima do arquivo na hierarquia, até o tópico-diretório que corresponda ao diretório-raiz que foi selecionado para mapeamento;

Com a ressalva de que se já houver tópicos no mapa de tópicos do sistema com o mesmo nome de algum dos diretórios, um novo tópico não é criado em correspondência ao mapeamento do diretório, apenas o mapeamento é feito no tópico que já existe. Da mesma forma, para as associações e as ocorrências.

Por exemplo, mapear o diretório *UPE*, ilustrado na Figura 18, criaria o tópico *UPE*, bem como um tópico para cada um de seus subdiretórios e também para os subdiretórios de seus subdiretórios, de maneira recursiva. Assim, teriam-se os tópicos: *UPE*, *2006.1*, *2006.2*, *Estágio Supervisionado*, *Iniciação Científica*, *TCC*, *Arquitetura Avançada*, *Modelagem e Simulação*, *Redes Neurais*, *Sistemas de Informação*, *Sistemas Distribuídos* e, da mesma forma, para cada um dos subdiretórios existentes até o nível mais baixo sob toda a árvore do diretório *UPE*. Com relação às associações, tomando como referência o tópico mapeado para o diretório *Sistemas de Informação*, este tópico teria sido associado aos tópicos *UPE* e *2006.2* e se associaria a todos os tópicos referentes aos subdiretórios do diretório *Sistemas de Informação*. Então, para os arquivos existentes no mesmo diretório *Sistemas de Informação* seriam criadas ocorrências nos tópicos: *Sistemas de Informação*, *2006.2* e *UPE* – em referência aos diretórios acima na hierarquia da árvore, até o diretório-raiz (*UPE*).

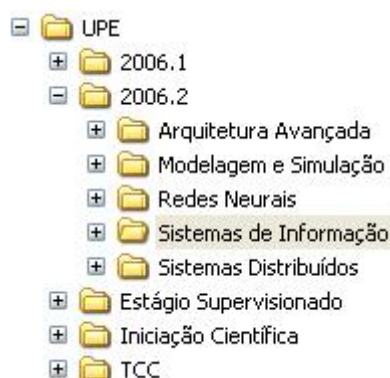


Figura 18. Árvore de diretório no *Windows Explorer*.

Uma das razões para o mapeamento de arquivos é poupar o usuário de ter que mapear arquivo por arquivo. Contudo, para que o mapeamento de um diretório produza o efeito desejado, é necessário que já haja uma boa organização no sistema de arquivos do usuário.

Aparentemente, o mapeamento de um diretório para o domínio dos mapas de tópicos não gera diferença alguma, uma vez que os relacionamentos existentes no sistema de arquivos são apenas refletidos para um mapa de tópicos. Contudo, não se deve esquecer a separação que o *Topic Maps* realiza entre o domínio dos arquivos e o domínio dos tópicos. Essa diferença, bem como os benefícios decorrentes, ficam mais claros ao se fazer um estudo de caso de uso comparativo entre as duas abordagens, como poderá ser observado no próximo capítulo.

4.4.3 Definição dos Escopos

Normalmente, os usuários do *Windows Explorer* já definem de certa forma, seus próprios escopos no sistema de arquivos. Se bem analisado, o próprio sistema de arquivos já define alguns escopos para o usuário, na forma de seus diretórios que compõem os níveis mais altos na hierarquia da árvore de diretórios, como: *Meu Computador*, *Meus Documentos* etc.

A idéia do *TME* é que o usuário possa definir seus próprios escopos. Mas que para tanto, possa aproveitar a organização dos arquivos já existente no sistema de arquivos, uma vez que geralmente, o usuário já faz alguma distinção de escopos na forma como organiza seus arquivos, diante do sistema de arquivos, ou seja, esta é uma visualização implícita advinda do sistema de arquivos do usuário. Por exemplo, algumas sugestões para escopos poderiam ser: *Trabalho*, *Faculdade* e *Casa*, aos quais, possivelmente, já existam respectivos diretórios no sistema de arquivos.

Pensando nisso, a definição de um escopo no *TME* significa, apenas, mapear um diretório do sistema de arquivos como outro qualquer (como visto na subseção anterior), com o diferencial de o tópico referente ao diretório mapeado ser utilizado e definido como um escopo no mapa de tópicos do *TME*.

Em contrapartida às operações de mapeamento, o *TME* possui um mecanismo de monitoramento, responsável por detectar falhas de mapeamento. De maneira que, se um arquivo ao qual existam ocorrências no mapa de tópicos for excluído do sistema de arquivos, o mecanismo de monitoramento detecta e exclui as ocorrências àquele arquivo nos tópicos que continham ocorrências a ele. Isto é, o que esse mecanismo faz é checar todas as ocorrências em todos os tópicos que existam no *TME*. Esse mecanismo está implementado como um processo que executa paralelamente ao funcionamento do *TME* (uma *Thread*⁹ da Java).

Por o escopo ser uma entidade especial e fundamental ao *TME*, um diretório mapeado em um escopo do *TME* também é monitorado, pelo mesmo processo que monitora as ocorrências nos tópicos. De forma que eventuais novidades no diretório, como novos subdiretórios e/ou arquivos são automaticamente mapeadas para o mapa de tópicos. E, como antes de realizar qualquer espécie de mapeamento, o *TME* verifica se aquele mapeamento já existe (ex.: se um tópico a ser criado já existir, ele, apenas, é referenciado), o que o recurso de monitoramento de escopos faz é, simplesmente, o constante mapeamento daquele diretório.

4.5 Visualização e Navegação

4.5.1 Concepção

O intuito de um mecanismo de visualização de informações para um sistema computacional qualquer é abstrair do usuário todas as complexidades existentes e necessárias ao seu funcionamento [3]. Só que no caso do *TME*, essa abstração já é alcançada através do padrão *Topic Maps*. Logo, o mecanismo de visualização do *TME* deve ser baseado no *Topic Maps*, podendo ou não abstrair-lo.

Abstrair o *Topic Maps*, no *TME*, com um mecanismo de visualização significaria criar dois níveis de abstração. Um nível seria o que o *Topic Maps* já realiza sobre os arquivos e o segundo nível seria a abstração do próprio *Topic Maps*. Por se tratar de um protótipo, escolheu-se por não abstrair o *Topic Maps* no *TME*, pelo menos não totalmente.

A idéia do *TME* para a representação visual do *Topic Maps* está ilustrada na Figura 19. Onde a esfera central, maior, representa o tópico que estiver sendo analisado, e as esferas menores, que estão a seu redor, os outros tópicos aos quais o tópico central estiver associado. Essa representação deixa a desejar na medida em que não contempla o contexto, uma vez que não permite ao usuário a percepção do restante do mapa de tópicos.

A princípio, a idéia era ter uma representação visual geral que contemplasse todo o mapa de tópicos, mas que permitisse também, visualizações detalhadas. A grande dificuldade de uma representação geral é devida às associações. A flexibilidade que o *Topic Maps* provê ao permitir que quaisquer dois tópicos do mapa de tópicos possam ser associados resulta numa inexistência total de qualquer tipo de dependência ou organização hierárquica entre os tópicos, na realidade existe apenas um nível hierárquico, o nível dos mapas de tópicos. De maneira que, a menos que

⁹ *Thread* é uma forma de divisão de um processo em dois ou mais que possam ser executados simultaneamente, que pode ser implementada pelo próprio sistema operacional ou através de uma biblioteca de uma determinada linguagem de programação.

seja redundante e contenha várias representações visuais para um mesmo tópico, uma representação visual geral para um mapa de tópicos muito grande (com muitos tópicos e associações) seria muito confusa do ponto de vista das associações entre os tópicos.

Mais adiante, quando será falado sobre interatividade e navegabilidade, será apresentada a solução encontrada pelo *TME* para superar a perda de contexto do seu modelo de visualização.

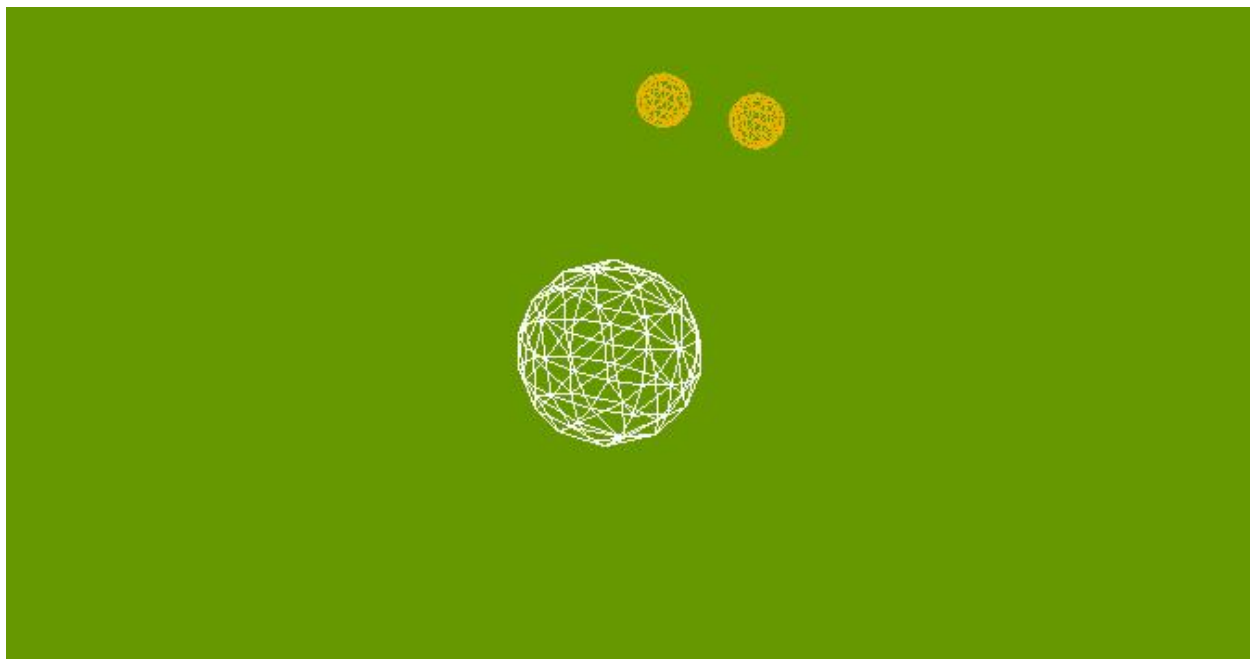


Figura 19. Representação da ideia conceitual para a visualização de mapas de tópicos no *TME*.

4.5.2 Interatividade e Navegabilidade

Quanto à interatividade, o *TME* realiza ações correspondentes a eventos específicos de *mouse*. Movimentar o *mouse* com o botão esquerdo pressionado reproduz movimentos de rotação na cena tridimensional dos mapas de tópicos. Enquanto dois cliques com o mesmo botão esquerdo sobre alguma das esferas que representam os tópicos, reflete uma ação de navegação no mapa de tópicos, fazendo com que o tópico em questão passe a ser o tópico focalizado (esfera central maior). Já dois cliques com o botão direito exhibe a tela para mapeamento manual de arquivos e apenas um clique exhibe a tela para gerenciamento e mapeamento de diretórios para definição de escopos.

Analisando a Figura 20, identificam-se no *TME* a existência de quatro elementos gráficos principais: a cena tridimensional dos mapas de tópicos, as tabelas das ocorrências (*filtered occurrences* e *topic occurrences*) e o *path navigation* e um mecanismo de busca (*search*).

Quanto à cena de visualização tridimensional, o *TME* gera uma cena independente para cada escopo. De modo que para alternar de um escopo para outro basta selecionar a aba correspondente na parte superior da tela.

Já o *path navigation* é responsável por armazenar e gerenciar o caminho de tópicos atualmente percorridos no escopo vigente. E é a partir desse caminho que a tabela de ocorrências filtradas (*filtered occurrences*) é preenchida. Esse preenchimento é realizado de forma que apenas a ocorrências aos arquivos em comum a todos os tópicos pertencentes ao *path navigation* são filtradas. Funcionando como um mecanismo de busca, onde as palavras-chaves estariam representadas pelo conjunto de tópicos informacionais percorridos. Enquanto a tabela *topic occurrences* lista todas as ocorrências existentes no tópico em foco (no escopo vigente).

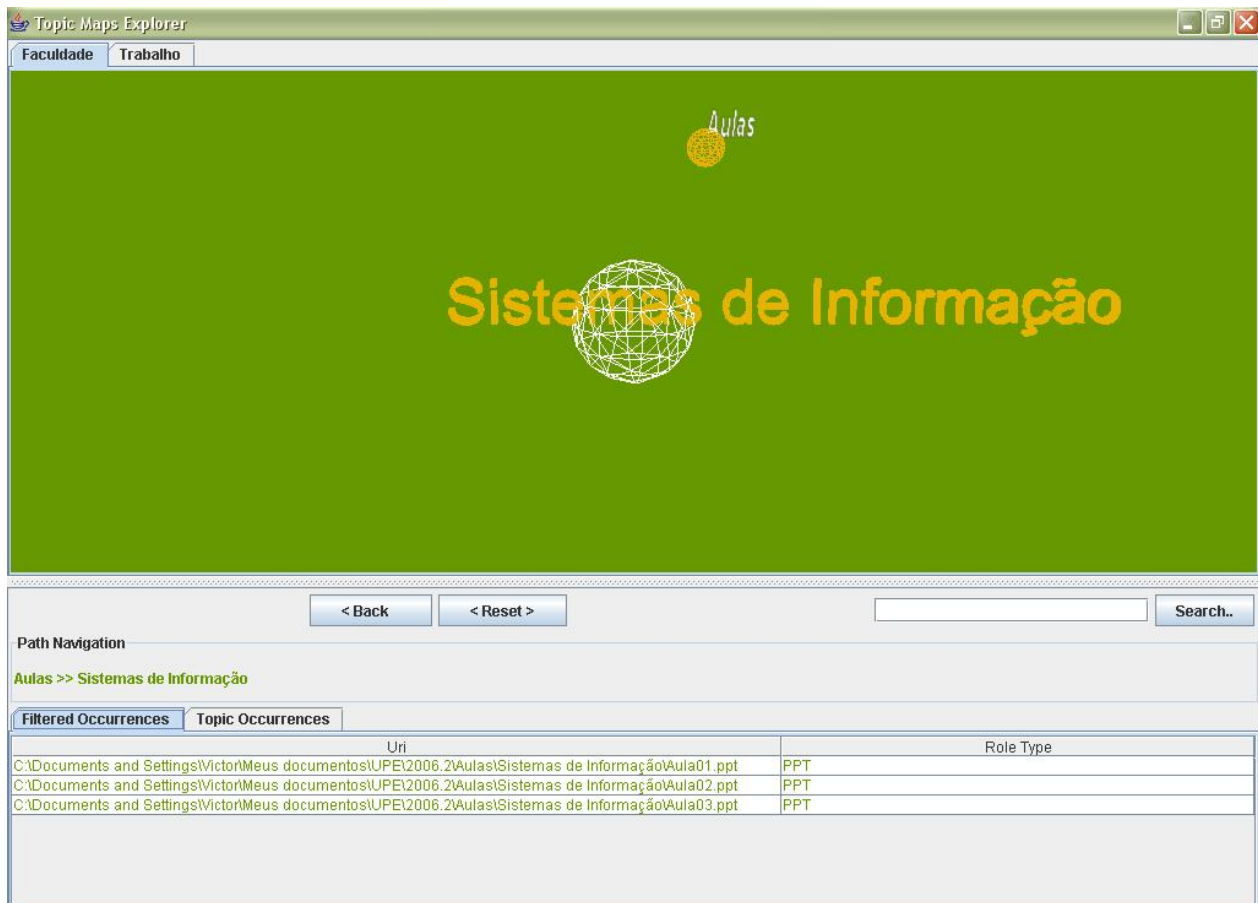


Figura 20. Ambiente de visualização gráfica do *TME*.

O *path navigation* também pode ser visualizado como o mecanismo de endereçamento dos arquivos a partir dos mapas de tópicos. Onde o endereço dos arquivos seria baseado em seu conteúdo representado pelas estruturas de conhecimento criadas nos mapas de tópicos. O que não deixa de ser um mecanismo de busca, como já afirmamos antes.

Por fim o mecanismo de busca em si, a partir dele, permite localizar qualquer tópico que esteja definido no escopo vigente. Para facilitar ainda mais essa operação, o *TME* disponibiliza em conjunto um recurso de *auto-completar* que lista os tópicos a partir da digitação do usuário.

4.6 Resumo

Protótipo de um *software* desenvolvido com a tecnologia Java, o *TME* implementa e utiliza os principais conceitos do padrão *ISO Topic Maps* como mecanismo para a organização e o acesso a arquivos pessoais.

Para isso, disponibiliza mecanismos para mapeamento e criação de mapas de tópicos a partir de diretórios e arquivos do sistema de arquivos. Bem como uma interface gráfica tridimensional para visualização e navegação desses mesmos mapas de tópicos.

Capítulo 5

Um Estudo de Caso de Uso Comparativo

5.1 Descrição

No decorrer deste trabalho, optou-se por dar enfoque especial ao problema da dependência físico-hierárquica existente nos sistemas de arquivos dos SOs existentes hoje no mercado – A grande motivação do *TME*. Nesse sentido, destacaram-se os benefícios obtidos pelo padrão *Topic Maps*, ao realizar uma separação entre o domínio dos relacionamentos semânticos dos mapas de tópicos e o domínio dos recursos da informação.

Neste capítulo é feita uma análise comparativa entre o *TME* e o *Windows Explorer* da *Microsoft*, a partir de um exemplo prático. Dando ênfase, justamente, ao problema da dependência físico-hierárquica encontrada no *Windows Explorer*, o qual o *TME* consegue superar, através da separação de domínios realizada pelo *Topic Maps*.

5.1.1 O Exemplo

O estudo inicia partindo de um exemplo no *Windows Explorer*. Supondo-se querer organizar os arquivos referentes às disciplinas cursadas por um estudante universitário do curso de Engenharia da Computação da Universidade de Pernambuco (UPE), seus arquivos poderiam estar dispostos no sistema de arquivos do *Windows Explorer* segundo a organização da árvore de diretórios ilustrada na Figura 21.

Analisando a Figura 21, tem-se um diretório-raiz *UPE*, cujos subdiretórios imediatos seriam referentes aos períodos letivos cursados pelo aluno (*2005.1*, *2005.2*, *2006.1*, *2006.2*), estes, por sua vez, teriam um subdiretório para cada disciplina cursada pelo estudante no referente período. Especificamente o diretório *2006.2* teria os subdiretórios para as homônimas disciplinas: *Arquitetura Avançada*, *Modelagem e Simulação*, *Sistemas Distribuídos*, *Sistemas de Informação* e *Redes Neurais*. Descendo um nível na árvore de diretórios, observamos que o diretório *Sistemas de Informação* conteria os subdiretórios: *Aulas*, *Projetos*, *Seminários* e *Subsídios* – consideremos que esta organização seria realizada em todos os diretórios referentes às disciplinas.



Figura 21. Árvore de diretórios no *Windows Explorer*: exemplo de organização um.

5.1.2 A Dependência Físico-Hierárquica

A dependência físico-hierárquica é facilmente identificada a partir do momento em que o estudante quiser encontrar algum arquivo segundo uma organização mental que seja diferente do caminho para se chegar a ele (o arquivo) diante da árvore de diretórios do *Windows Explorer*, uma vez que, este caminho é único.

O estudante poderia querer, por exemplo, que seus arquivos estivessem organizados como mostra a Figura 22, onde se observa uma pequena diferença em relação à organização anterior, com uma inversão na hierarquia dos diretórios. Agora, os diretórios referentes aos períodos letivos seriam, diretamente, constituídos pelos subdiretórios: *Aulas*, *Projetos*, *Seminários* e *Subsídios*. E estes, é que teriam como seus subdiretórios os diretórios referentes às disciplinas.



Figura 22. Árvore de diretórios no *Windows Explorer*: exemplo de organização dois.

Com isso é verificado que, dada à dependência físico-hierárquica imposta pelo *Windows Explorer*, o estudante estaria limitado a escolher uma ou outra organização, nunca as duas. A menos que fizesse cópias de seus arquivos para cada uma das organizações. Além do mais, verifica-se a existência de redundância quanto aos nomes dos diretórios. Nesta segunda

organização, por exemplo, em cada um dos subdiretórios do diretório 2006.2 (*Aulas, Projetos, Seminários, Subsídios*) há a replicação dos diretórios referentes às mesmas disciplinas.

5.1.3 A Solução do TME

O que antes eram os diretórios e seus relacionamentos hierárquicos no *Windows Explorer*, no TME esses elementos passam a ser tópicos e associações entre tópicos, respectivamente. Utilizando o mesmo exemplo do estudante universitário e a mesma organização de arquivos suposta no *Windows Explorer* teríamos, no TME, os tópicos: *UPE, 2005.1, 2005.2, 2006.1, 2006.2, Arquitetura Avançada, Modelagem e Simulação, Sistemas Distribuídos, Sistemas de Informação, Redes Neurais, Aulas, Projetos, Seminários e Subsídios*;

Diferentemente do *Windows Explorer*, não há no TME um tópico *Aulas* para cada disciplina, mas sim, apenas um tópico *Aulas* em todo o mapa de tópicos. E neste tópico *Aulas* existiriam ocorrências a todos os arquivos referentes às aulas de todas as disciplinas, inclusive de períodos letivos distintos. O que o TME faz para diferenciar esses arquivos é filtrá-los segundo o caminho de tópicos percorridos pelo usuário no mapa de tópicos. De maneira que, se o usuário quiser acessar seus arquivos referentes às aulas da disciplina *Sistemas de Informação*, basta que “visite” os dois tópicos do mapa de tópicos: *Aulas* e *Sistemas de Informação* (Figura 23).

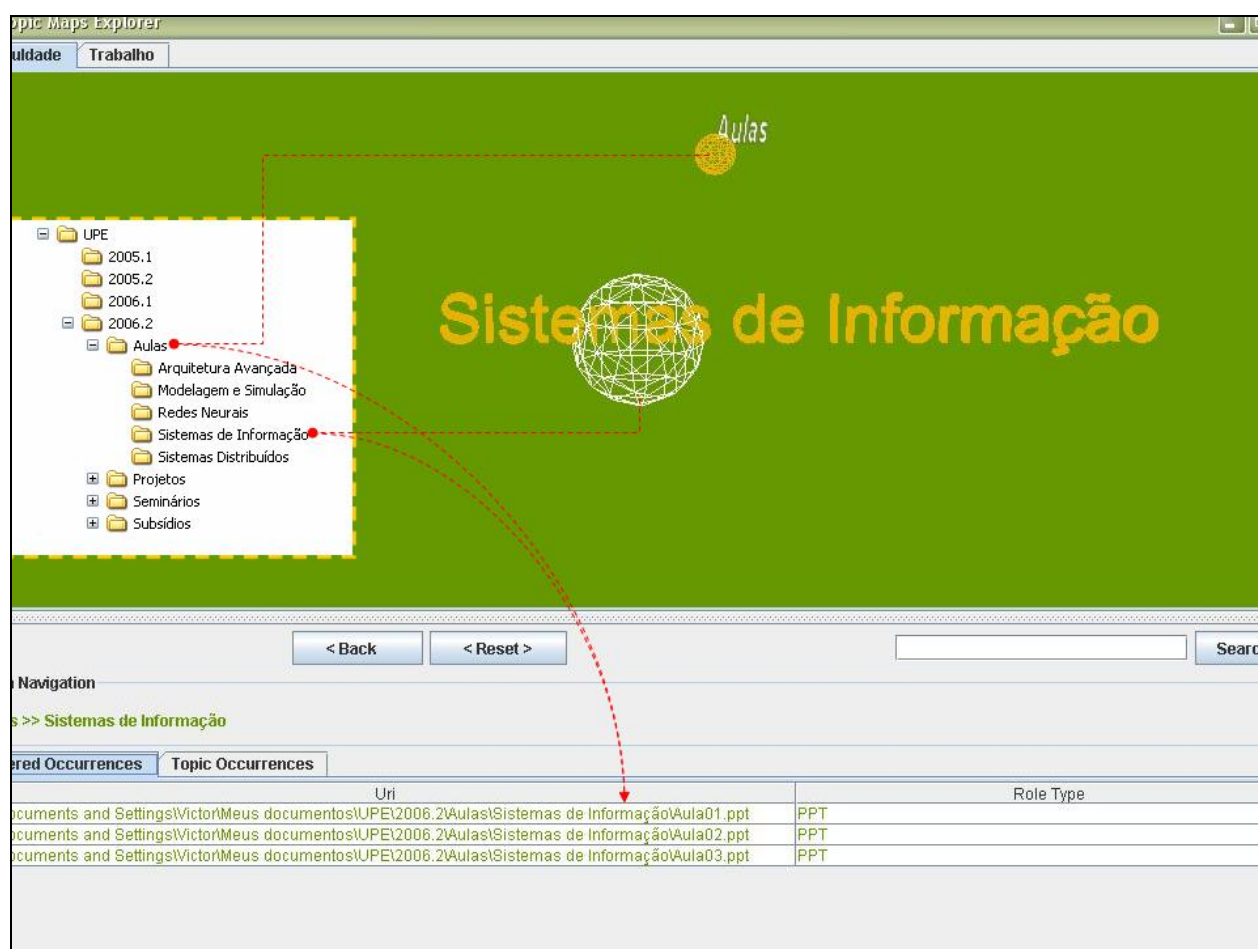


Figura 23. Exemplo de mapeamento de arquivos para *Topic Maps*: solução TME.

Vale ressaltar que, por não existir níveis de hierarquia nos mapas de tópicos, mas sim, um único nível ou domínio de todos os tópicos, não importa se o estudante irá “visitar” primeiro o

tópico *Aulas* ou o tópico *Sistemas de Informação*, para o caso do exemplo do parágrafo anterior. Se “visitar” primeiro o tópico *Aulas*, inicialmente serão listadas todas as ocorrências aos arquivos referentes às aulas de todas as disciplinas, então, posteriormente, ao “visitar” o tópico *Sistemas de Informação*, essas ocorrências serão filtradas para apenas aquelas que forem comuns a ambos os tópicos, resultando no mesmo conjunto de arquivos que existia no subdiretório *Aulas* do diretório *Sistemas de Informação* no *Windows Explorer*. Da mesma forma para o caso de visitar primeiro o tópico *Sistemas de Informação*, só que, de início, seriam listadas as ocorrências a todos os arquivos referentes à disciplina, incluindo não somente as aulas, mas também, projetos, seminários e subsídios.

5.2 Resumo

O princípio da utilização do *Topic Maps* vem da separação entre o domínio dos relacionamentos semânticos e o domínio dos recursos de informações que este padrão realiza.

No sistema de arquivos dos Sistemas Operacionais hoje no mercado, os relacionamentos entre os arquivos estão misturados a eles, diante da organização e correspondentes disposições na estrutura da árvore de diretórios. Isso significa que: para relacionar dois arquivos quaisquer é necessário manipulá-los, fisicamente na estrutura da árvore de diretórios. Isto, além de limitar as possibilidades de relacionamentos entre os arquivos, dificulta o acesso e a localização dos mesmos, que é única.

Com a separação de domínios provida pelo *Topic Maps*, o *TME* não só abstrai limitações como essas – relativas à dependência quanto à localização física dos arquivos, que são observadas no sistema de gerenciamento de arquivos dos Sistemas Operacionais atuais. Como também permite a construção de verdadeiras estruturas de conhecimentos sobre arquivos pessoais, representadas na forma de mapas de tópicos. E então, permite operar na forma de um mecanismo de busca e navegação sobre arquivos (a partir dessas estruturas de conhecimento), sem a necessidade de alterar ou manipular os arquivos diretamente.

Capítulo 6

Conclusão e Trabalhos Futuros

6.1 Conclusão

Dada a crescente dificuldade de acesso às informações pessoais, nem sempre se chega facilmente a elas, seja porque seus modos de apresentação não sejam convenientes, ou seus volumes e diversidades sejam grandiosos. Julga-se, portanto, ser bastante necessária a construção de uma ferramenta que simplifique a organização e o acesso a informações pessoais.

O fato de o padrão *Topic Maps* possibilitar a construção de estruturas de conhecimento de forma dinâmica, permite que cada usuário defina uma estrutura própria para organizar seus arquivos; da maneira que melhor lhe convir; de forma pessoal, não se limitando a nenhum modelo fixo ou preestabelecido.

Isso, aliado ainda às facilidades de navegação tridimensional, pode contribuir para uma nova abordagem para a representação e o acesso às informações pessoais.

Neste trabalho, foi desenvolvida uma ferramenta computacional que possibilita uma nova forma para a organização de arquivos pessoais. Permitindo a seu usuário definir, visualizar e navegar sobre estruturas de tópicos informacionais que funcionam como mecanismos indexadores dos arquivos. Seu grande potencial está na separação que realiza entre o domínio dos tópicos informacionais e o domínio relativo à localização física dos arquivos. Com essa separação de domínios, a ferramenta permite tanto integrar a informação contida nos arquivos – o que representa a construção de conhecimento – quanto distinguir e classificar os arquivos de acordo com os relacionamentos criados no domínio dos tópicos. Procurar um arquivo é como remeter à memória, buscando-o a partir de tópicos informacionais, que quando somados e relacionados distinguem e classificam os arquivos.

6.2 Trabalhos Futuros

Novas funcionalidades podem ser adicionadas, especificamente ao aperfeiçoamento em sua interface de visualização e navegação gráfica. Esforços podem ser empreendidos no sentido de melhorar a forma como o usuário cria as novas visualizações.

Outra possibilidade é incrementar os efeitos visuais, que tornem o ambiente de visualização do *TME* mais sofisticado e agradável. A própria Java 3D possui uma infinidade de

recursos, tais como aparência, comportamento, localização, textura, luz, som, direção de observação, dentre outros, que permitem a configuração de ambientes de visualização mais sofisticados.

Há, também, a possibilidade de aplicar uma nova abordagem ou técnica de visualização que melhor apresente e represente os mapas de tópicos. Ou ainda, uma que abstraia os mapas de tópicos. *Kamiya* [1] sugere, por exemplo, apresentar os mapas de tópicos como cidades, onde os edifícios seriam os tópicos, cujas associações estariam representadas pelas ruas e avenidas que os interligassem.

Outra possibilidade seria tornar todo o ambiente gráfico do *TME* tridimensional. De forma que, cada botão ou janela de configuração fosse transformado em um elemento do universo virtual, e não apenas a representação dos mapas de tópicos fosse tridimensional.

Um trabalho fundamental a ser realizado está relacionado ao quesito persistência. Uma vez construídos os mapas de tópicos sobre os arquivos, esses mapas de tópicos precisam ser armazenados, para que possam ser recuperados em outro momento da execução do *TME*. Assim, é fundamental ao *TME* um mecanismo de persistência de informações, neste caso, um mecanismo de persistência de mapas de tópicos, que permita armazenar e recuperar os mapas de tópicos construídos sobre os arquivos.

Atualmente, a persistência dos mapas de tópicos é realizada de maneira improvisada, em um arquivo *XML*, sendo utilizada, apenas, para efeito dos testes funcionais do *TME*. Uma boa alternativa seria utilizar o *HSQLDB*¹⁰ – *Hipersonic SQL Database*, um banco de dados embarcado, escrito em Java, que tem como grande vantagem a possibilidade de ser agregado ao pacote das aplicações.

Por se tratar de um protótipo, o *TME* desperta, também, para novos nichos de aplicação do padrão *Topic Maps*. Desde 2001, com a publicação da *XML Topic Maps (XTM)*, criada para ser facilmente utilizada em ambiente *Web*, o principal foco para a utilização desse padrão é a *Internet*, dado o grande volume de informação que circula. Contudo, dada a sua flexibilidade e caráter genérico, o *Topic Maps* pode ser utilizado como mecanismo de representação e acesso aos mais diversos tipos de informação.

Empresas e organizações diariamente recebem, geram e armazenam grandes quantidades de diferentes tipos de informações. A existência de um mecanismo que seja capaz de sintetizar esses diferentes tipos de informações numa mesma base de conhecimentos, possibilitaria sua melhor gestão, permitindo que sejam analisados todos os relacionamentos que existam, sem o risco de deixar de levar em consideração algum importante e eventual parâmetro. Neste sentido, *Topic Maps* pode ser uma boa alternativa, ao menos num nível de abstração mais alto.

¹⁰ Disponível em <<http://www.hsqldb.org/>>.

Referências Bibliográficas

- [1] KAMIYA, E.I. *Topic Maps* como instrumentos de representação e recuperação da informação. Monografia (Bacharelado em Biblioteconomia). Departamento de Biblioteconomia e Documentação, Escola de Comunicações e Artes, Universidade de São Paulo. São Paulo. 2005.
- [2] FREITAS, C.M.D.S, CHUBACHI O.M., LUZZARDI, P.R.G., CAVA, R.A. Visualização de Informações, Revista de Informática Teórica e Aplicada (RITA). Edição VIII. Rio de Janeiro. 2001. pp. 143-158.
- [3] HUGO A.D.N. e CRISTIANE B.R.F. Visualização de Informações – Uma Abordagem Prática. XXV Congresso da Sociedade Brasileira de Computação. Rio Grande do Sul. 2005.
- [4] TANENBAUM, A. S. and WOODHULL, A. S. Operational Systems - Design and implementation. Prentice Hall, 2 edition, 1997.
- [5] BIEZUNSKY, M., BRYAN, M. and NEWCOMB, S. ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34, December, 1999.
- [6] PEPPER, S. *The TAO of Topic Maps: finding the way in the age of infoglut*. 11 Apr. 1999. Disponível em: <<http://www.ontopia.net/topicmaps/materials/tao.html>>, visitado pela última vez em 05/10/2006.
- [7] SHNEIDERMAN, B. Designing the user interface - Strategies for effective human-computer interaction, Addison-Wesley, 2.ed., Maryland, USA, 1992.
- [8] WITTEN, I. H. & FRANK, E. (2000). Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations, volume 1. Morgan Kaufmann. <http://www.cx.waikato.ac.nz/ml/weka/index.html>.
- [9] PRESSMAN, R. S. : Engenharia de Software, Makron Books/McGraw-Hill, 1995, Tradução do original “Software Engineering: A Practitioner’s Approach”, 3rd Ed., McGraw-Hill, 1992.
- [10] REZENDE, S.O. Sistemas Inteligentes: Fundamentos e Aplicações. Editora Manola, 2002.
- [11] ROBERTSON, G., CARD, S., MACKINLAY, J. Cone Trees: Animated 3D Visualizations of Hierarchical Information. Proceedings of ACM CHI’91, 1991, pp. 189-194.
- [12] JOHNSON, B. e SHNEIDERMAN, B. TreeMaps: A space - filling approach to the visualization of hierarchical information structures. Proceedings of IEEE Visualization. San Diego, USA. 1991. pp 284 – 291
- [13] LAMPING J., RAO, R. The hyperbolic browser: a focus+context technique for visualizing large hierarchies, Journal of Visual Languages and Computing, 7(1):33-55, Marco 1996.
- [14] MUNZNER, T. H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. In: Proceedings of the IEEE Information Visualization Symposium. 1997. pp 2-10.

- [15] CAVA, R.A. e FREITAS, C.M.D.S. Node-Edge Diagram Layout for Display Hierarchies. Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, Santa Catarina, 2001. (pôster).
- [16] XML Topic Maps: Finding Aids for the Web. Disponível em <<http://www.coolheads.com/SRNPUBS/ieee-mm-topicmaps-article.pdf>>, visitado pela última vez em 17/09/2006.
- [17] CORNELL, G. e HORSTMAN, C. S. Core Java 2: Fundamentos. Primeira edição. Volume 1. Editora Makron Books, 2000.
- [18] DEITEL, H.M. e DEITEL, P. J. Java: Como Programar. Terceira edição. Editora Bookman, 2001.
- [19] Sun Developer Network. Disponível em: <<http://java.sun.com/>>, visitado pela última vez em 15/10/2006.
- [20] MCILROY, M.D. Mass-Produced Software Components. In: Software Engineering. Petrocelli/Charter, Brussels7, Belgium. 1976. pp.88-98.
- [21] Sun Microsystems Java 3D Engineering Team. Java 3D API Tutorial. Disponível em: <<http://developer.java.sun.com/developer/onlineTraining/java3d/>>, visitado pela última vez em 10/10/006.