

GraphMiner: Uma Ferramenta de Mineração Visual de Dados em Bases Relacionais

Manoel C. Marques Neto, Manoel Mendonça, Celso A. Saibel Santos

Universidade Salvador

CEPERC – Centro de Pesquisas Interdepartamental em Redes de Computadores

Rua Ponciano de Oliveira, 126, CEP: 41950-275, Salvador, BA – Brasil

{manoel.netom, mgmn, saibel}@unifacs.br

Abstract. The development of the technology has allowed companies and organizations accumulate large amounts of data in databases. However, the ownership of these data does not necessarily translate into useful information. For that, one needs techniques and tools to efficiently explore data. Amongst the data mining approaches, visual data mining is possibly the most intuitive, because it uses the human ability to quickly interpret visual scenes. Visual data mining techniques utilize visual metaphors to facilitate data interpretation by human beings. This work presents a visual data mining tool that uses graphs as its main visual metaphor. This tool, called GraphMiner, maps structured data stored in relational databases into easily interpretable graphs. Moreover, the GraphMiner provides a series of query and visualization widgets to explore the produced graph. It creates an environment that allows the efficient and intuitive exploration of data stored in relational databases.

1. Introdução

A quantidade de informações armazenadas em repositórios computacionais de dados tem crescido mais rápido que a nossa capacidade de explorá-los. Além disso, a complexidade (número de registros e quantidade de atributos) de muitos conjuntos de dados supera a habilidade dos seres humanos de identificar os conceitos subjacentes às informações armazenadas.

O ser humano resolve o problema de compreender uma situação através da identificação e classificação de informação estruturada [12]. Em muitos casos, as estruturas classificadas podem ser visualizadas de maneira que o resultado dessa visualização facilite a compreensão de estruturas conhecidas e de padrões que se repetem. A escolha da metáfora visual mais adequada para visualização de estruturas é fundamental para facilitar o entendimento humano sobre os dados. Nesse sentido, grafos são estruturas visuais interessantes, pois destacam as relações implícitas em dados no formato estruturado. O problema do desenho de grafos em um computador é tratado há algum tempo pela ciência da computação. Dentre os algoritmos disponíveis, destacam-se aqueles pertencentes à família de algoritmos denominada força-dirigida [12].

A mineração visual de dados permite combinar a capacidade de construir cenas visuais complexas com controles de seleção interativa dos dados. Estas funcionalidades combinadas permitem que um especialista possa, de forma iterativa, explorar os dados.

Atualmente, os mecanismos muito comuns para o armazenamento de dados estruturados são bancos de dados relacionais. Esses mecanismos, além de armazenar os dados estruturados, tratam de forma semelhante os metadados que descrevem esses dados. Isto facilita a obtenção das relações entre os dados, que são elementos essenciais para mineração visual através de grafos.

O objetivo deste trabalho é a apresentação de uma ferramenta que apóia a descoberta de informações úteis em bancos de dados relacionais através da representação gráfica desses dados sob a forma de um grafo. A principal contribuição deste trabalho é oferecer um mecanismo que permita a exploração de dados relacionais de forma simples e eficiente. A ferramenta implementada, denominada GraphMiner, é genérica o suficiente para que seja independente de um banco de dados ou domínios específicos.

Este artigo está organizado da seguinte forma: a seção 2 apresenta as linhas teóricas existentes e o estado da arte de visualização de informações baseadas em grafos. A seção 3 apresenta a abordagem para a construção visual de um grafo a partir de dados armazenados em bases relacionais. A seção 4 descreve detalhadamente a ferramenta GraphMiner e sua implementação e por fim, a seção 5 apresenta as conclusões do trabalho.

2. Visualização de Informações Baseada em Grafos

O problema de compreender uma situação complexa é resolvido pela mente humana através de uma abstração, que significa identificar e classificar informação estruturada. Após sua identificação, as estruturas classificadas podem ser visualizadas de maneira que o resultado dessa visualização seja determinístico, isto é, as estruturas similares devem resultar em visualizações similares. Isto facilita a compreensão de estruturas conhecidas e de padrões que se repetem. Estas idéias conduzem à nova metodologia de visualização de dados no formato estruturado, que engloba as seguintes etapas [12]: (1) **Identificação** da estrutura dos modelos que é definida pelos submodelos e por seus relacionamentos; (2) **classificação** das subestruturas individuais e (3) **visualização** das subestruturas e seus relacionamentos.

A Figura 1 ilustra um exemplo de visualização de um circuito hidráulico baseada na metodologia descrita. Primeiro, a estrutura do circuito textualmente definida é identificada, isto é as subestruturas são esboçadas. Estas subestruturas são então classificadas: S denota uma subestrutura do tipo fonte e W significa uma subestrutura do tipo trabalho. Finalmente, o modelo visual é extraído. A disposição dá ênfase à informação estruturada colocando elementos da mesma subestrutura próximos. O padrão repetitivo das três unidades trabalhando é enfatizado através do seu posicionamento paralelo.

A escolha da metáfora visual mais adequada para visualização de estruturas é fundamental para facilitar o entendimento humano sobre os dados. Nesse sentido, grafos são estruturas visuais interessantes, pois, destacam as relações implícitas em dados no formato estruturado.

2.1. Algoritmos para Visualização de Grafos

O desenho de grafos não é uma área nova dentro da ciência da computação [8]. A maior parte dos algoritmos para visualização de grafos pode ser enquadrada em duas famílias: Força-Dirigida e Desenho Orientado a Nível.

Os algoritmos de força-dirigida usam o tamanho das arestas para desenhar um grafo de forma que a disposição resultante seja intuitiva ao ser humano. Os diferentes comprimentos de cada uma das arestas são corrigidos movendo os nós correspondentes para uma posição melhor.

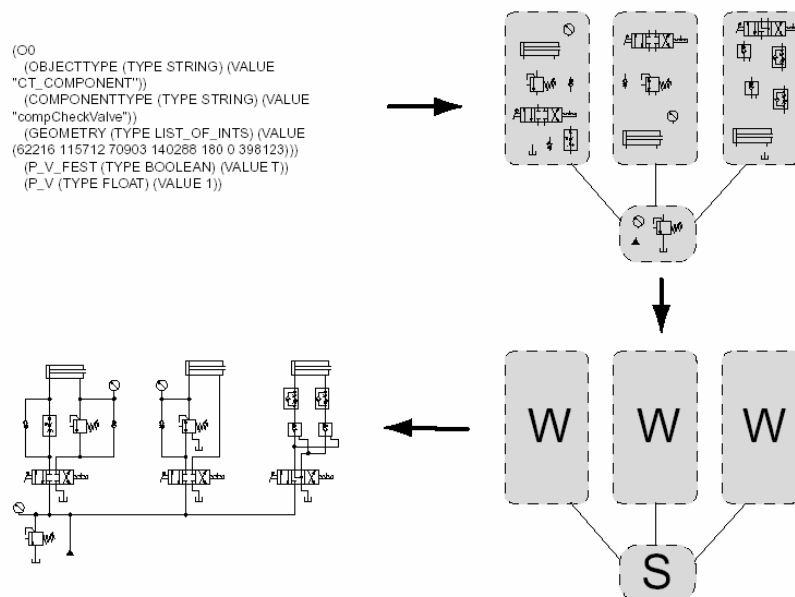


Fig 1. Metodologia de visualização de dados estruturados [12].

Os algoritmos de desenho de grafos orientados a nível são normalmente usados em grafos dirigidos e compreendem três passos [13]: Primeiro os nós são distribuídos em camadas horizontais; depois são organizados de forma a minimizar o número de cruzamentos de arestas; e finalmente, a coordenada horizontal x é ajustada para cada um dos nós.

2.2. Ferramentas para Visualização Baseada em Grafos

Diversos trabalhos têm sido desenvolvidos na área de visualização de informações baseados em grafos. Alguns deles diferem no modelo de arquitetura adotado ou nos algoritmos usados para construção visual. Porém, de alguma forma, todos implementam a visualização das relações entre os dados. As ferramentas e modelos analisados neste trabalho foram agrupados com base nos seguintes critérios: área de

aplicação ou objetivo (desenho de grafos, visualização de informação); família de algoritmos utilizados (a maioria deles utiliza algoritmos da família força-dirigida e orientado a nível) e controles de visualização.

O sistema denominado IGD [4] é utilizado para construção visual de grafos de forma interativa na Web. O IGD permite um alto grau de interação do usuário com a cena visual e contempla operações que vão desde a escolha do algoritmo utilizado até funções básicas de visualização, como rotação e aproximação.

O GRAPPA [6] é um arcabouço (framework) para visualização de grafos que simplifica o processo de apresentação visual, bem como a manipulação de características específicas dentro de aplicações. Além disso, ele possui uma série de funcionalidades e características importantes para visualização de dados como os métodos para construir, manipular, navegar e mostrar grafos com todos os seus elementos (nós, arestas e possíveis subgrafos).

O sistema denominado Zoomgraph [1] permite que usuários possam visualizar grafos com atributos complexos de nós e arestas, fornecendo ainda uma linguagem de script para manipular e explorar estes desenhos. A ferramenta, que é pública, inclui também um sistema de visualização baseado em aproximação, que permite aos usuários uma movimentação fácil entre as diversas visões do grafo.

Arcabouços de análise, tais como o Ucinet [3], são usados para dividir o grafo e na execução de vários cálculos sobre a sua estrutura. A saída destes arcabouços é passada então a outros como Pajek [2], Netdraw [3], GraphViz [6], ou KrackPlot [9] que são poderosos para desenhar e executar operações de disposição e que, por necessidade, tendem a atender somente aos requisitos mais comuns para visualização de grafos (cores, formas, etc.).

O TouchGraph [14] é um arcabouço escrito em Java que permite a construção de aplicações para visualizar redes de informação inter-relacionadas, tais como em bases de dados relacionais ou documentos hiper-mídia na Web. As redes são construídas como grafos interativos que se prestam a uma variedade de atividades. Acoplado a uma imagem, um usuário pode navegar através de grandes redes e explorar maneiras diferentes de arrumar os seus componentes na tela, realizando operações como aproximação, rotação e navegação nos diferentes níveis do grafo. O arcabouço tem código fonte aberto e implementa os algoritmos de força-dirigida. Várias aplicações usam este arcabouço como base da sua construção [14].

3. O Desenho de Grafos a partir de Bancos de Dados Relacionais

A utilização das técnicas de visualização de dados baseadas em grafos apresentadas na seção 2 exige que os dados estejam estruturados. Geralmente, as ferramentas de mineração visual de dados relacionais devem realizar o pré-processamento dos dados, uma vez que nem sempre estes dados estão no formato estruturado. A situação ideal para visualizar dados em grafos é obtida quando os dados já estão em formato estruturado adequado ao uso de grafos para visualização, pois isto simplifica muito o seu pré-processamento. Este é exatamente o caso dos dados armazenados em bancos de dados relacionais.

A construção de um grafo a partir de um banco relacional pode ser realizada em quatro etapas básicas:

- (1) Escolher uma entidade (entidade raiz) do banco que se quer visualizar.
- (2) Descobrir quais são as entidades que se relacionam com a entidade raiz (isto é feito, através de uma consulta que recebe como parâmetro o nome da entidade raiz e é feita na tabela que guarda as informações sobre as relações do banco).
- (3) Selecionar as tuplas da entidade raiz e de cada uma das entidades relacionadas.
- (4) Mapear as tuplas como nós do grafo e as relações entre elas como arestas (note que para cada tupla da entidade raiz, existe uma ou mais tuplas relacionadas).

A Figura 2 exibe um exemplo de construção de um grafo a partir de um banco de dados relacional com três entidades (“filme”, “gênero” e “produtora”). Inicialmente, a entidade “filme” é selecionada como raiz. Depois disso, descobre-se as entidades relacionadas: “gênero” e “produtora”. Para facilitar a visualização, devem-se selecionar atributos visuais que serão associados a cada tipo de entidade. Na Figura 4, “produtoras” são associadas a retângulos, “filmes” a elipses e “gêneros” de filmes a retângulos de bordas arredondadas. As tuplas das entidades de “filme”, “gênero” e “produtora” são selecionadas e mapeadas como nós. Para cada tupla da entidade “filme”, seleciona-se uma tupla relacionada de “gênero” e outra “produtora”. Para cada par de tuplas “filme-gênero” e “filme-produtora”, cria-se uma aresta.

4. A Ferramenta GraphMiner

A GraphMiner é uma ferramenta voltada à visualização de relações entre informações que segue os princípios de uma ferramenta para mineração visual de dados [11]. Ela usa como metáfora visual um grafo e tem como objetivo montar uma representação visual dos relacionamentos de dados contidos em bancos de dados relacionais, permitindo desta forma, a descoberta de informações potencialmente úteis e inéditas até então.

A partir de mecanismos de acesso a uma base de dados, um grafo é construído e apresentado de forma gráfica e navegacional. Na construção da cena visual, os registros das diversas entidades são representados como nós e as relações entre eles, como arestas.

Seguindo a tendência da maioria das ferramentas da área de visualização baseada em grafos (apresentadas na seção 2.2), a GraphMiner foi construída usando os algoritmos de força-dirigida. Esses algoritmos já estão implementados no arcabouço TouchGraph, o qual é usado como base da construção da ferramenta.

4.1. A Arquitetura da Ferramenta

A ferramenta GraphMiner é dividida em quatro módulos: (1) entrada de dados; (2) apresentação visual; (3) controle de consulta e (4) controle de interação. A arquitetura da ferramenta é apresentada pela Figura 3.

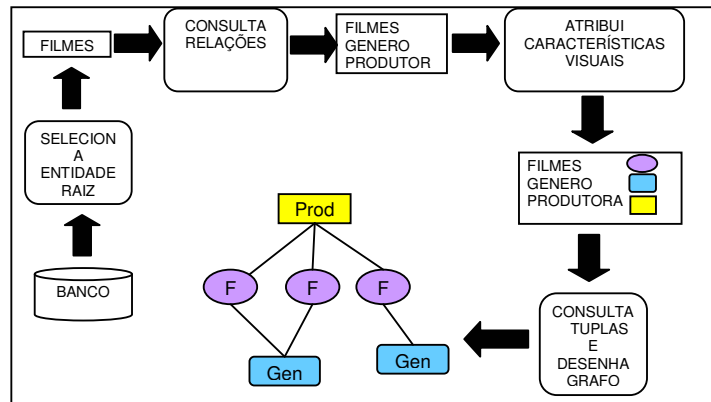


Fig 2. Passos para construção de um grafo a partir de um banco de dados relacional.

Os módulos da ferramenta foram implementados a partir do agrupamento de classes em pacotes de software. Estes componentes são unidades autocontidas, independentes e reutilizáveis que interagem através de suas interfaces públicas. Desta forma, a lógica de funcionamento da ferramenta está dividida entre os componentes dos seus quatro módulos, visando simplificar o seu projeto e implementação. A seguir, é apresentada a descrição detalhada de cada módulo, com destaque às suas principais funcionalidades.

4.1.1. Módulo de Entrada de Dados

O módulo de entrada de dados foi desenvolvido com o objetivo de permitir a integração da GraphMiner aos diversos tipos de bancos de dados relacionais existentes. Esta integração proporciona um aumento na quantidade das fontes de informação que podem ser utilizadas pela ferramenta. Uma das funcionalidades desse módulo é o processo de carga automática de informações provenientes de bases de dados através de uma interface padrão. As classes concretas que cumprem essa interface contêm os requisitos necessários para implementar a lógica de acesso a dados em bases relacionais. Para isto, estas classes utilizam a API JDBC (Java Database Connectivity) e um driver específico disponibilizado pelo banco. Esta API e o driver oferecem um mecanismo de comunicação a partir do qual é possível realizar consultas em SQL cujo resultado fornece informações de entrada do sistema. Estas informações são os registros armazenados em tabelas do banco e os metadados que descrevem esses registros. Na ferramenta estes registros são representados como nós do grafo e os metadados, quando definem as relações entre os dados, são representados como arestas.

O módulo de acesso a dados usa, além dos componentes citados, outros componentes que permitem a configuração, de forma visual, das informações necessárias à obtenção dos dados. O módulo permite visualização do Diagrama de Entidade-Relacionamento (DER) de um banco de dados relacional. A partir desse diagrama, um usuário pode escolher que entidades devem ser visualizadas pela ferramenta e também qual será a entidade raiz. Cada uma das entidades é um nó do

grafo e as relações entre as entidades são representadas graficamente através de arestas do grafo.

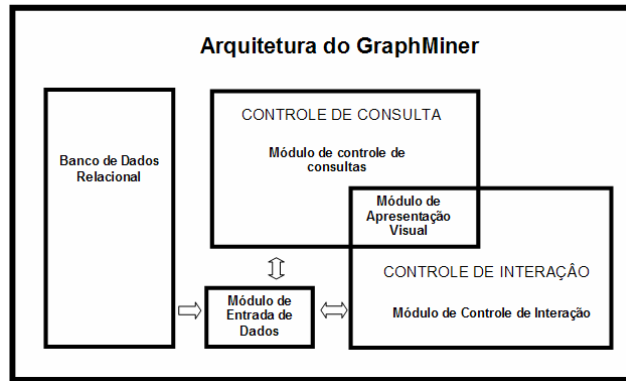


Fig 3. Arquitetura da ferramenta GraphMiner.

4.1.2. Módulo de Apresentação Visual

A visualização de informações em bancos de dados pode revelar relacionamentos ocultos e implícitos entre suas diversas entidades. Na GraphMiner, isto é obtido através da associação do conjunto de dados a uma série de atributos visuais. No módulo de apresentação visual existem dois níveis de visualização: (1) Visualização do DER onde um usuário pode escolher as entidades usadas no processo de exploração visual (ver seção 4.1.1) e, (2) visualização dos dados das entidades selecionadas onde cada registro de uma tabela é representado como um nó do grafo e seus relacionamentos como arestas. Os atributos visuais cor, tamanho, forma e etiquetagem são utilizados para a criação da cena visual. A ferramenta permite ainda que um usuário possa alternar entre os dois tipos de visualização, escolhendo uma nova entidade raiz e, conseqüentemente, obtendo uma nova visão dos dados e suas relações.

O módulo utiliza o arcabouço TouchGraph (ver seção 2.2) para implementar a representação visual de grafos. No desenvolvimento deste trabalho foi necessária uma adequação do TouchGraph para que o grafo pudesse exibir informações específicas dos dados explorados. Estas modificações permitem selecionar as tuplas de cada entidade e as relações entre elas e, compô-las em objetos que representam os nós e as arestas do grafo, respectivamente. Uma vez que a estrutura de nós e arestas das entidades selecionadas está montada, pode-se realizar a construção do grafo. Para isto, foi feita uma modificação na classe do TouchGraph, que contém um painel cujo objetivo é permitir o desenho de figuras geométricas como círculos, retângulos e elipses. É nesse novo painel que o grafo é desenhado. Esse desenho é feito com a simples inserção dos objetos contidos na estrutura de entidades mencionada.

Para cada conjunto de nós que representa as tuplas de uma entidade do banco de dados, existe um conjunto de atributos visuais associados a forma, cor e rótulo exibido. Um dos nós da entidade raiz é colocado no centro do painel de visualização e tem a borda colorida de vermelho para diferenciá-lo dos demais. Cada atributo visual pode ser associado com o valor de um atributo das entidades da base. Desta forma, o

usuário da ferramenta GraphMiner pode identificar os dados e suas relações através de atributos visuais que direcionam sua atenção de forma a enfatizar informações importantes associadas a estes dados. A Figura 4 mostra a visualização de um conjunto de dados e de suas relações através da ferramenta GraphMiner.

4.1.3. Módulo de Controle de Consulta

Os controles de consulta são componentes visuais que permitem a execução de operações de seleção sobre o conjunto de dados manipulado. Desta forma, os controles de consulta permitem a filtragem dos dados a serem mostrados na tela [11]. Um controle é criado para cada atributo de uma das entidades do banco.

Associada ao controle, é criada uma estrutura de dados em árvore. Essa estrutura representa os atributos, os valores que estes podem assumir e quais registros estes valores estão referenciando. Na GraphMiner, o controle de consulta está baseado apenas em barras de intervalo. Neste controle pode-se selecionar uma faixa de valores para o atributo representado. Pode-se pressionar a seta da direita para se ajustar o limite superior da seleção, a seta da esquerda para se ajustar o limite inferior da seleção e a barra central para se ajustar os dois limites ao mesmo tempo.

O controle baseado em barras de intervalo torna-se particularmente útil na manipulação de dados numéricos. Existem dois tipos de dados com os quais o GraphMiner trabalha: os dados categóricos e os dados numéricos. Dados numéricos são aqueles que possuem sentido de unidade. Dados categóricos possuem apenas sentido de ordem ou categoria (ex: tipos de veículos “Passeio”, “Carga”, “Luxo”) [14]. Ao se utilizarem os controles baseados em barras de intervalo nos dados, os itens que não fazem parte da seleção desaparecem da tela quando a opção “Esconder nós após filtragem” está selecionada. Se esta opção não estiver marcada, os nós serão coloridos de forma diferente dos demais.

4.1.4. Módulo de Controle de Interação

Com o objetivo de facilitar a exploração visual dos dados, a ferramenta GraphMiner oferece controles que permitem ao usuário interagir com a representação gráfica criada.

O controle de aproximação (zoom) permite aumentar e diminuir as distâncias entre os nós através da modificação do tamanho das arestas. Essa funcionalidade facilita a visualização de informações amenizando problemas como o cruzamento de arestas e a sobreposição de nós. Através de uma barra de rolagem um usuário pode modificar o grau de aproximação aplicado ao grafo.

A ferramenta GraphMiner disponibiliza também um controle de interação que permite ao usuário girar o grafo em torno do nó centralizado denominado controle de rotação. Através de uma barra de rolagem similar a usada no controle de aproximação pode-se girar e modificar as posições dos nós atribuindo valores que vão de 0 a 359 graus.

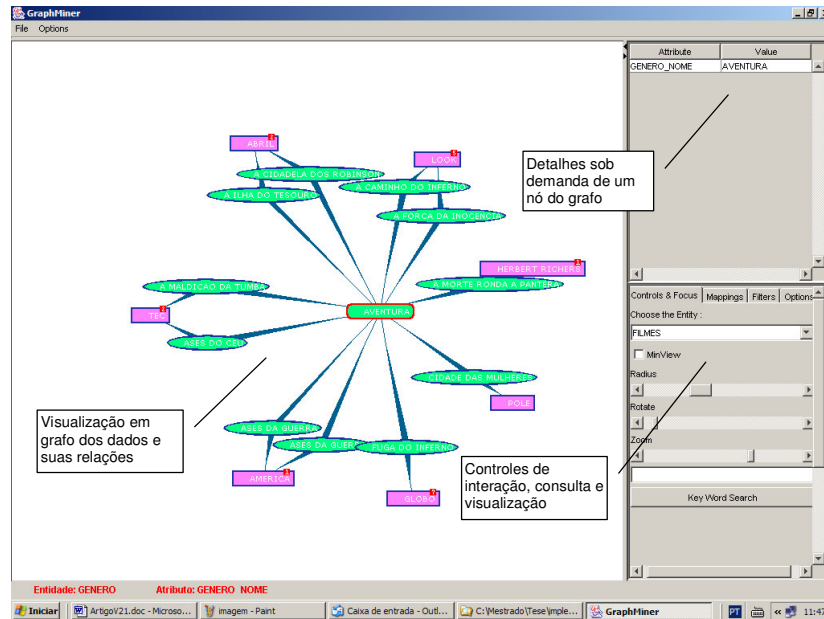


Fig 4. Visualização de dados e suas relações através da GraphMiner.

Outra funcionalidade que segue a mesma linha do controle e aproximação é o controle do raio de exibição. Ele permite controlar a quantidade de nós exibidos na tela usando como parâmetro para mostrar ou ocultar um nó a sua distância do nó selecionado. Neste caso, a distância entre dois nós corresponde à quantidade de arestas entre o nó de origem selecionado e nó destino. Quando o nível de visualização do grafo é zero, somente o nó selecionado é mostrado. O controle do raio de exibição só vai até o valor quatro, a partir desse valor todos os nós do grafo são exibidos. O número de nós visíveis na tela do computador pode ajudar a melhorar a visibilidade do grafo.

A ferramenta disponibiliza ainda os controles de interação que permitem o mapeamento de atributos visuais exibidos no grafo como a cor, o formato e o rótulo dos nós e arestas com qualquer um dos atributos do banco de dados relacional. A escolha do atributo visual é feita através de uma caixa de seleção que contém a lista de atributos de uma das entidades do banco. Cada caixa de seleção está associada a um dos quatro atributos visuais do grafo. O atributo selecionado a partir de uma das caixas é classificado como categórico ou numérico e, a partir dessa classificação, a modificação de um atributo visual é executada. Por exemplo, associando-se um atributo categórico (“Bom”, “Ruim” ou “Regular”) com o controle de cor do nó, uma cor (ex: Azul, Vermelha ou Preta) é associada a cada nó (“Bom” -Azul, “Regular” -Vermelha, “Ruim” -Preta). A utilização desses controles é exibida na Figura 5.

Um outro mapeamento que pode ser realizado é a associação de uma imagem (ícone) a uma das entidades selecionadas. Essa funcionalidade facilita a identificação dos nós que representam as tuplas da entidade selecionada. Por exemplo, a Figura 6 exhibe um grafo cujos nós exibem mapeamentos de atributos visuais com atributos do

banco de dados relacional da seção 3. Os nós no formato de elipse e com uma imagem associada são do tipo filme e o nó retangular no centro é do tipo gênero e os nós do tipo produtora estão ocultos pelo controle do raio de exibição mencionado anteriormente. A cor de cada nó do tipo filme está mapeada com atributo do banco de dados produtora. Isso significa que nós com as mesmas cores têm as mesmas produtoras. Pode-se observar pela figura que os filmes “A Ilha do Tesouro” e “A Cidadela dos Robinson” são da mesma produtora, pois tem a mesma cor.

5. Conclusões

Este artigo apresentou uma abordagem para mapeamento de dados estruturados armazenados em bases relacionais para grafos. Esta abordagem engloba a definição de uma ferramenta para a visualização de relacionamentos entre informações de diferentes domínios. A ferramenta se destaca pelo fato de permitir o mapeamento e a visualização de dados de maneira interativa e intuitiva, apoiando o processo de descoberta de conhecimento a partir de dados relacionais.

A possibilidade de mapear diversos atributos visuais com os dados diferencia a GraphMiner das demais ferramentas analisadas. Além disso, um usuário da ferramenta pode interagir com a cena produzida através de diversos controles de interação modificando os atributos visuais e navegando sobre os dados. Isso é feito com intuito de aumentar a percepção sobre dados visualizados.

Um ponto positivo da GraphMiner é o componente de acesso a dados. Esse componente permite o acesso de forma transparente a qualquer banco de dados relacional, sem necessidade de re-programar a ferramenta. Além disso, outro fato positivo é mapeamento dos dados armazenados em um banco de dados relacional para grafos, que é feito de forma automática. Além disso, vale ressaltar que a construção da cena de visualização é feita de maneira interativa.

Uma limitação apresentada pela maioria das ferramentas de mineração visual baseada em grafo é a quantidade de dados usados no processo de mineração. Esse problema é tratado parcialmente pela GraphMiner através do controle do raio de exibição. A utilização desse controle resolve a questão da sobreposição dos elementos do grafo. Porém, não resolve o problema da insuficiência de memória para comportar a quantidade nós criados para a representação visual quando o número de registros é muito alto. Resta ainda dizer que existem limites na capacidade humana em compreender cenas visuais com um número elevado de objetos. Assim, o fato da ferramenta não dar suporte a visualização de uma grande quantidade de nós não é tão relevante.

Para avaliar a versão apresentada da ferramenta GraphMiner, foram realizados dois exemplos de uso, envolvendo uma base de dados dos projetos realizados no Núcleo Interdepartamental de Pesquisas em Redes de Computadores (NUPERC) da Universidade Salvador (UNIFACS) e o relatório sobre o ranking de publicações em visualização de informação (InfoVis).

Além das limitações citadas, os exemplos de uso realizados apontaram novas funcionalidades que podem vir a ser acrescentadas à ferramenta em trabalhos futuros. A principal extensão sugerida foi que o aplicativo permita salvar os objetos que

compõem o grafo na cena visual. Esta funcionalidade é especialmente útil para situações que envolvam consultas complexas sobre os dados, pois permitiria ao usuário salvar consultas e mapeamentos visuais complexos e reusá-las quando necessário. Outra necessidade apontada durante os exemplos de uso foi a disponibilização de um controle que permita a seleção e o agrupamento ou expansão de múltiplos nós do grafo. O usuário através destes controles, poderia focar a visualização em uma determinada área do grafo amenizando o problema da visualização de grafos com muitos nós. Finalmente, a adoção de tabelas de cores dependentes do domínio de aplicação está também sendo estudada.

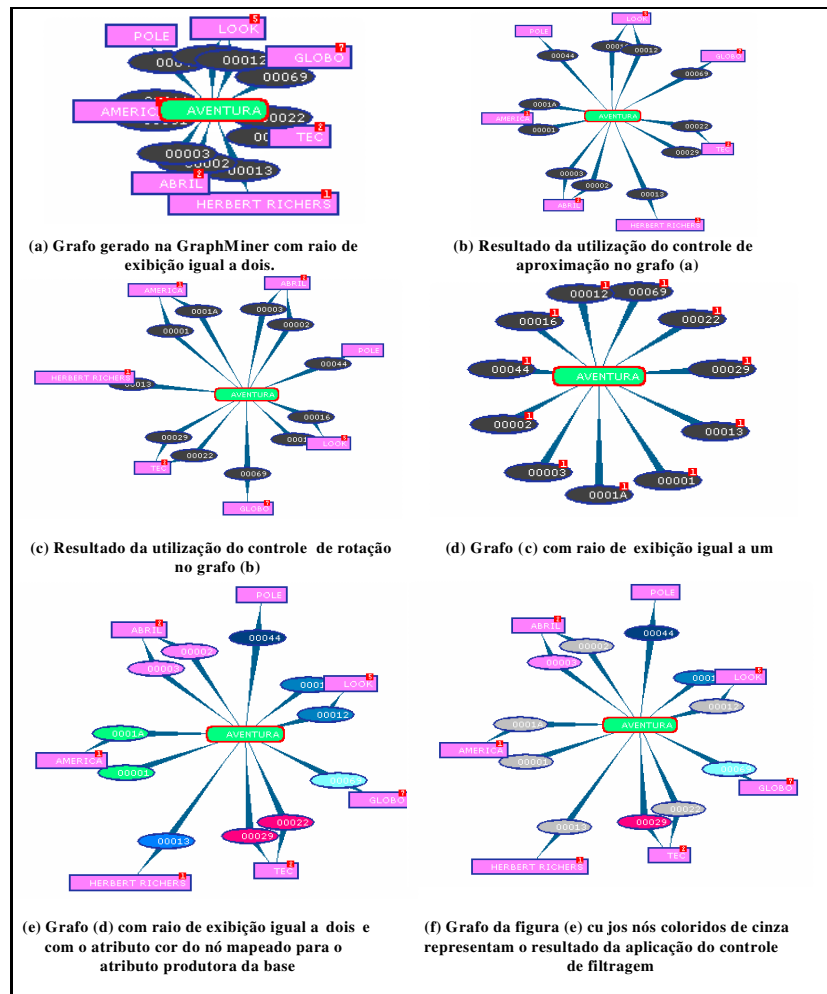


Fig 5. Utilização dos Controles de Interação.

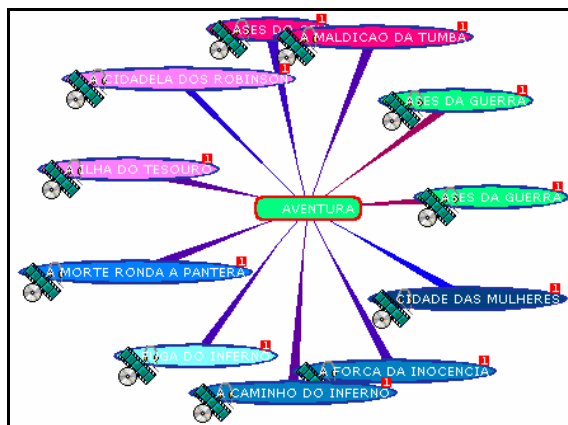


Fig 6. Grafo com mapeamentos de atributos visuais.

Referências

1. Adar, E., Tyler, J., Zoomgraph. Information Dynamics Lab, HP Laboratories CA – USA, 2003.
2. Batagelj, V., Mrvar, A., Pajek – Program for Large Analysis, Connections, 21:47-47, 1998.
3. Borgatti, S., Everett, M. G., Freeman, L.C., UCINET, Analytic Technologies, Disponível em: <http://www.analytictech.com>, Acesso em Janeiro de 2004.
4. Erlingsson, U., Krishnamoorthy, M., Interactive Graph Drawing on the World Wide Web, Rensselaer Polytechnic Institute, NY, 1996.
5. Fayyad U., Mining Databases: Towards Algorithms for Knowledge Discovery,” Bulletin of Tech. Committee on Data Engineering, vol. 21, pp. 29-48, 1998.
6. Gansner, R., North, S., An open graph visualization system and its applications to software engineering. AT&T Labs - Research, Shannon Laboratory, NJ- USA, 1999.
7. Guha, S., Rastogi, R., Shim, K., Cure: An efficient clustering algorithm for large databases. Proceedings of ACM SIGMOD International Conference on Management of Data, 1998.
8. Knuth, D. E., Computer Drawn Flowchart, Commun.. ACM, 6, 1963.
9. Krackhardt, D., Blythe, J., Mcgrath, C., KrackPlot 3.0 User’s Manual, Analytic Technologies, 1995.
10. Marshall, M.S., Herman, I., Malancon, G., An object-oriented design for graph visualization, 2000, CWI Technical Report, INS-R0001.
11. Oliveira M.C.F. de, Levkowitz, H., From Visualization to Visual Data Mining: A Survey. Accepted for publication in IEEE Transactions on Visualization and Computer Graphics.
12. Niggemann, O, Visual Data Mining of Graph-Based Data, Ph.D. Thesis, University of Paderborn, Germany, 2001.
13. Sugiyama, S. T. K., Toda, M., Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics, 11(2), 1981.
14. TOUCHGRAPH, TouchGraph LLC. Disponível em: <http://www.touchgraph.com>. Acesso em Janeiro de 2004.