

Integrando múltiplas visões em uma ferramenta de visualização de grafos

Lucas Nowaczyk Seadi, Andre Suslik Spritzer, Carla M.D.S. Freitas
Instituto de Informatica
Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
{ Inseadi , spritzer , carla }@inf.ufrgs.br

Resumo—Grafos são a forma mais intuitiva de visualizar características relacionais e hierárquicas de conjuntos de dados. A visualização de grafos se popularizou nos últimos anos em função da ampla e intensa utilização de redes sociais. Este artigo apresenta a integração de múltiplas visualizações simultâneas para explorar relações em redes sociais complexas. São utilizadas visualizações dirigida à força, radial e radial com meta-nodos. As técnicas foram integradas a uma ferramenta de visualização de grafos disponível.

Abstract—Graphs are the most intuitive way of visualizing the relational and hierarchical aspects of a dataset. Graph visualization has recently grown in popularity due to the widespread and intense use of social networks. This paper presents the integration of multiple simultaneous visualizations in order to explore relationships in complex social networks. Force-directed layouts, radial layouts, and rayal layouts with meta-nodes are employed. The techniques have been integrated to an available graph visualization tool.

Keywords—Visualização de Grafos; Grafos Dirigidos à Força; Grafos Radiais;

I. INTRODUÇÃO E MOTIVAÇÃO

Na última década, a popularidade de aplicações de *networking* aumentou dramaticamente. Atualmente, as pessoas frequentemente fazem parte de muitas redes sociais: eles se comunicam em grupos ou fóruns em tópicos de interesse, trocam e-mails com seus amigos e colegas, expressam suas ideias em blogs, compartilham vídeos no YouTube, trocam e comentam fotos no Flickr ou no Facebook.

Sistemas de *networking* recentes com crescente aceitação, tais como LinkedIn ou Facebook, são dedicados exclusivamente a gerenciar e estender a rede social de um indivíduo. Dessa maneira, estas comunidades online crescem rapidamente e sem esforço graças a um efeito de bola de neve. Comparadas a dados coletados através de pesquisas e entrevistas, redes online são muito maiores e contêm informações muito mais ricas. Essa avalanche de novas e vastas bases de dados geram novos desafios: as ferramentas precisam suportar a análise de uma quantidade muito grande de dados, geralmente evoluindo com o tempo.

Analisar como as pessoas se comunicam, colaboram, que informações elas trocam e que papéis elas desempenham no grupo social está se tornando um ponto de interesse de uma grande variedade de organizações, transcendendo o uso

peçoal. Não há modelos confiáveis de tais redes; portanto, há uma necessidade por ferramentas que forneçam a análise exploratória da estrutura de redes sociais reais. Para este fim, grafos são uma ferramenta poderosa.

Normalmente, os usuários finais têm que lidar diretamente com grafos e por isso, geralmente, é de seu interesse visualizá-los e interagir com eles de maneira a chegar a uma melhor compreensão. A representação visual natural de um grafo é o diagrama nodo-aresta, onde os nodos são exibidos como objetos visuais, conectados uns aos outros por linhas que representam as arestas [1].

O problema de computar um nodo-aresta pode ser descrito simplesmente como encontrar as posições geométricas dos nodos que são mais interessantes esteticamente para a melhor compreensão do grafo e sua estrutura. Resolver este problema, porém, provou-se uma tarefa muito complexa, tornando-se o foco do campo do desenho de grafos.

Há uma multiplicidade de algoritmos para desenho de grafos, com a escolha de qual usar dependendo do tipo de grafo que se queira desenhar e dos critérios estéticos escolhidos para o *layout* gerado [1], [2]. Embora haja algumas linhas gerais sobre o que torna um grafo melhor compreendido por um humano, a escolha de um algoritmo depende fortemente da aplicação pretendida. Assim, é interessante para uma ferramenta de visualização de grafos oferecer a possibilidade de gerar visualizações com diferentes algoritmos.

O presente trabalho tem como objetivo estender a ferramenta denominada MagnetViz, com layouts diferentes daquele gerado pelo algoritmo default. O restante do texto está organizado como segue. A seção ??, a seguir, descreve os trabalhos relacionados, incluindo o MagnetViz. A seção III detalha a extensão do MagnetViz para múltiplas visões, incluindo *layout* radial e exibição de subgrafos como metanodos. Finalmente, a seção IV mostra as conclusões do trabalho e direções para sua continuidade.

II. TRABALHOS RELACIONADOS

A. Algoritmos baseados em força e MagnetViz

Como ponto de partida para este trabalho é usada MagnetViz [3], uma técnica para visualização de grafos baseados em força.

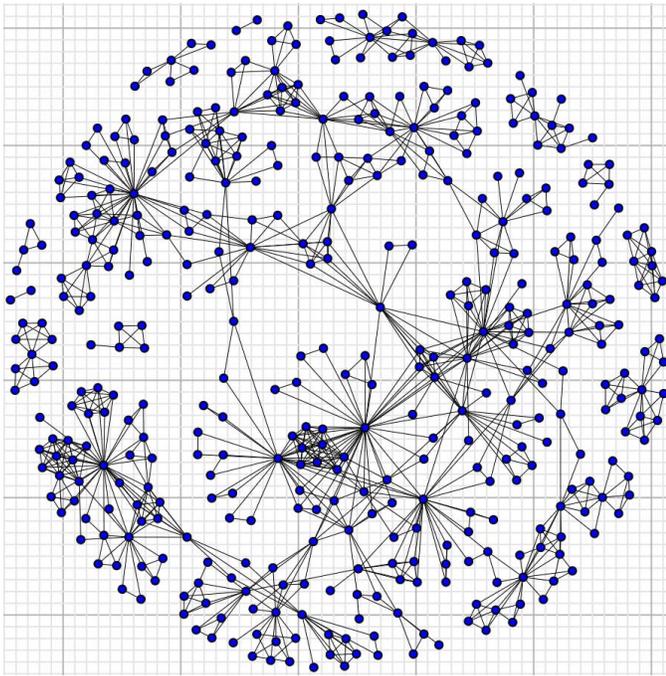


Figura 1. Uma rede de co-autorias visualizada com MagnetViz.

Os algoritmos baseados em força constituem uma das famílias clássicas de algoritmos de desenho de grafos. Sendo uma subclasse dos *layouts* baseados em otimização, esses algoritmos tratam o grafo como um sistema físico, atribuindo forças aos nodos e arestas e minimizando a energia até que um layout estável seja alcançado.

Já que os *layouts* dirigidos à força são essencialmente simulações físicas, eles são muito flexíveis na aplicação de restrições – tudo que puder ser expresso por forças pode ser integrado ao algoritmo. Esta flexibilidade e o fato de que eles geralmente produzem *layouts* limpos e de aparência orgânica os fez serem adotados por designers de aplicação apesar das desvantagens de não serem determinísticos e serem computacionalmente pesados.

Outra desvantagem dos algoritmos dirigidos à força é que, como a maioria das outras técnicas de desenho de grafos, eles consideram apenas a topologia do grafo. Isso pode por vezes deixar de fora dados importantes na computação do layout, ao ignorar a informação semântica contida nos atributos dos nodos e arestas. Esta informação é muito dependente da aplicação e pode ser tão importante para os usuários quanto os relacionamentos que são expressos explicitamente pelo diagrama de nodos e arestas.

Enquanto a maior parte das técnicas visualizam um *layout* de grafo estático pré-computado, MagnetViz permite que usuários dinamicamente alterem o *layout* de um grafo de forma a melhor satisfazer suas necessidades. Isso é feito estendendo a metáfora de física dos algoritmos dirigidos à força (Figura 1) para proporcionar aos usuários ímãs virtuais, que podem atrair nodos que satisfazem um conjunto de restrições associadas a eles (Figura 2).

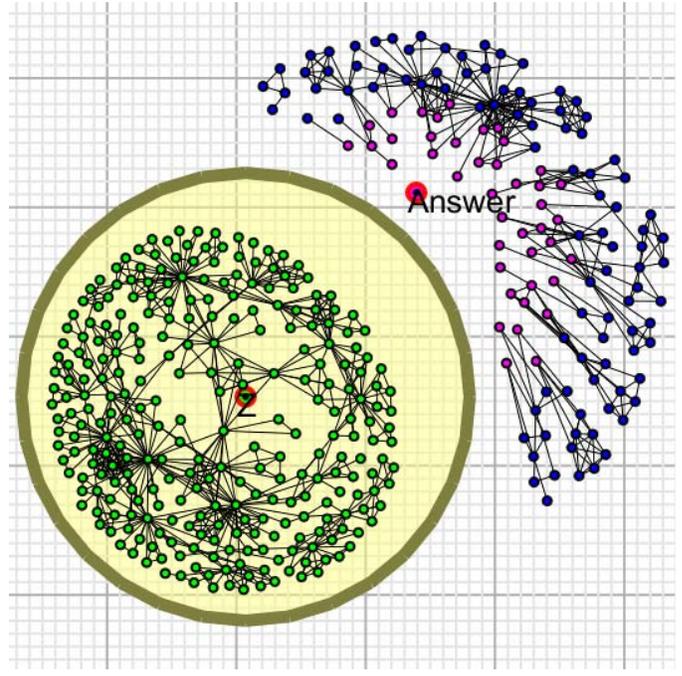


Figura 2. Ímã virtual isolando um sub-grafo exibido com uma *boundary shape*.

Restrições podem ser baseadas na topologia ou semântica do grafo. Através de *boundary shapes*, que são simples formas geométricas que podem ser colocadas ao redor de ímãs, usuários podem também definir regiões na cena onde os nodos atraídos devem permanecer (Figura 2).

Os grafos são descritos para o MagnetViz usando GraphML, uma linguagem baseada em XML, que permite a especificação dos nodos e arestas e de atributos para essas entidades. Após a submissão de um grafo como entrada, MagnetViz o exibe utilizando uma versão modificada do algoritmo clássico de Fruchterman e Rheingold [4], e permite que o usuário, a seguir, insira ímãs na cena. Usuários podem construir as condições associadas aos ímãs utilizando os atributos dos nodos e arestas, além de atributos topológicos próprios de grafos. Ímãs também podem ser criados baseados em outros ímãs, permitindo operações de conjunto sobre os dados.

No presente trabalho, o objetivo é integrar outras formas de visualização dos grafos no MagnetViz, de modo a proporcionar melhor visualização dependendo da topologia do grafo. A próxima subseção aborda outros leiautes de grafos.

B. *Layouts Alternativos*

Além dos algoritmos baseados em forças, há inúmeras técnicas de visualização de grafos, baseadas em princípios diferentes, descritas na literatura [1], [2]. A escolha de um *layout* alternativo para integração ao MagnetViz levou em conta a agilidade na geração da visualização e as possibilidades de interação previstas.

Dentre as formas de visualização de grafos pesquisadas, uma das técnicas é a do *layout* radial [5], a qual foi escolhida para implementação. Nessa técnica, um nodo é posicionado no

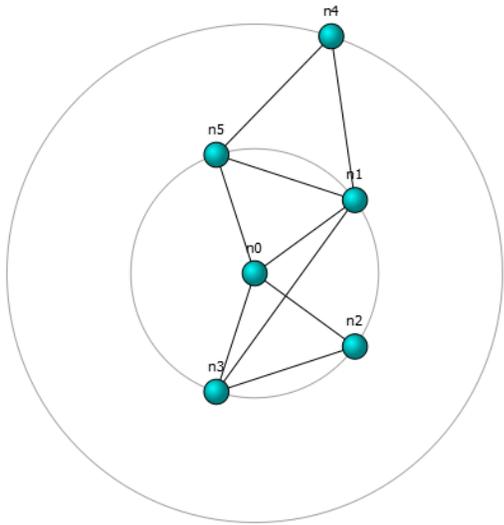


Figura 3. Grafo em layout radial.

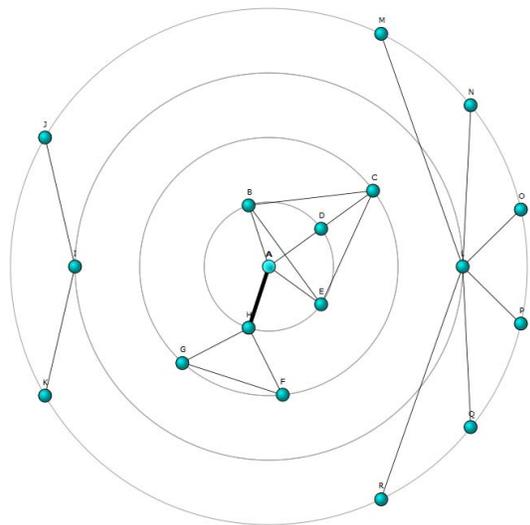


Figura 4. Grafo com componentes não conectados ao nodo-foco.

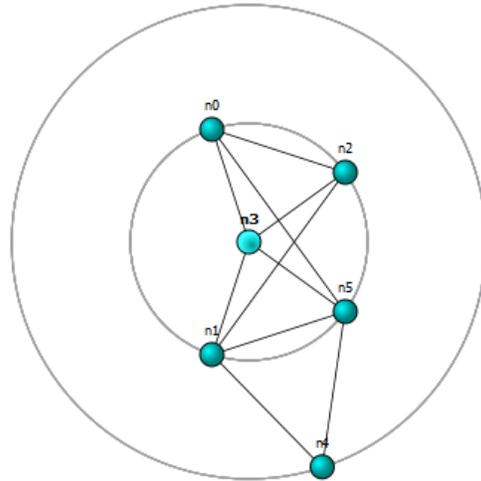
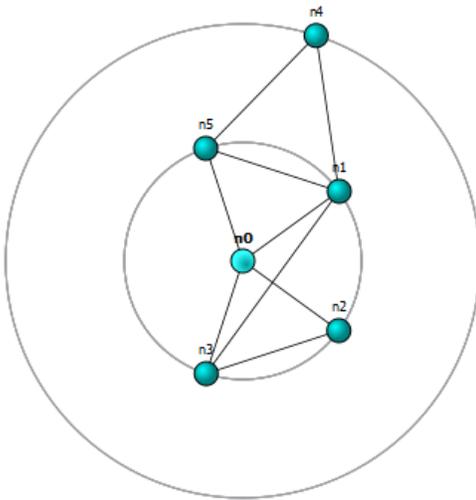


Figura 5. Mudança de foco do nodo n0 (esquerda) para o nodo n3 (direita).

centro e os demais são distribuídos ao redor dele, dispostos em anéis concêntricos em torno do nodo em foco (Figura 3). Cada nodo reside no anel correspondente à sua menor distância do foco. Vizinhos imediatos do foco residem no anel mais inferior, seus vizinhos residem no segundo menor anel, e assim por diante. Os anéis são desenhados explicitamente para tornar a distância aparente. A posição angular do nodo em seu anel é determinada pelo setor do anel alocado a ele. A cada nodo é alocado um setor dentro do setor designado a seu pai, com tamanho proporcional à largura angular da sub-árvore daquele nodo.

Componentes não conectados ao nodo em foco são exibidos em novas árvores cujas raízes estão distribuídas no nível seguinte ao último da árvore principal, com os setores iniciais distribuídos igualmente entre elas (Figura 4). Ao selecionar um nodo, ele se torna o novo foco, deslocando-se para o centro, com o restante do grafo se rearranjando em torno dele (Figura

5).

Para reduzir a desorientação que pode ser causada durante o rearranjo, é usada animação para executar uma transição suave, além de algumas restrições para manter o novo leiaute similar ao atual, tornando a transição mais fácil de acompanhar.

Um aspecto importante na obtenção de visualizações compreensíveis, mais organizadas visualmente, é o tratamento dado a grafos onde existem subgrafos densos, fortemente conectados, que podem atrapalhar a análise de relações entre subgrafos. Para melhorar a apresentação do grafo num nível de abstração mais alto, algumas ferramentas permitem apresentar subgrafos como nodos especiais (metanodos) [6]–[8]. Assim, um grafo composto por diversos subgrafos pode ser representado como um grafo de metanodos. Arestas entre metanodos indicam que existe pelo menos uma aresta entre dois nodos pertencentes aos dois subgrafos.

A incorporação de visualização de subgrafos como me-

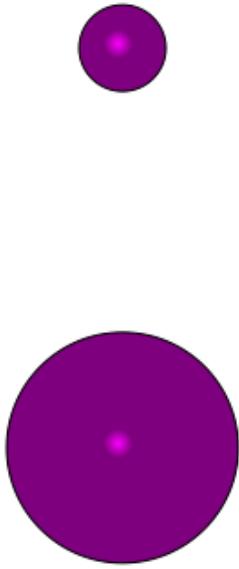


Figura 6. Estado inicial do layout radial com meta-nodos.

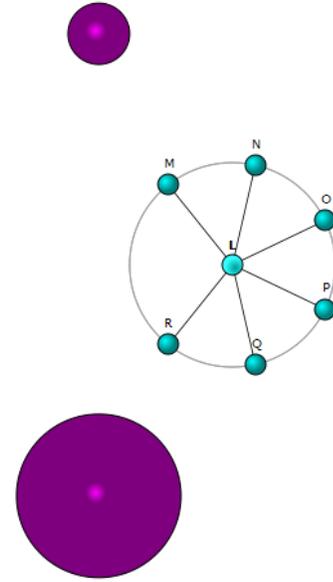


Figura 7. Componente correspondente ao meta-nodo expandido.

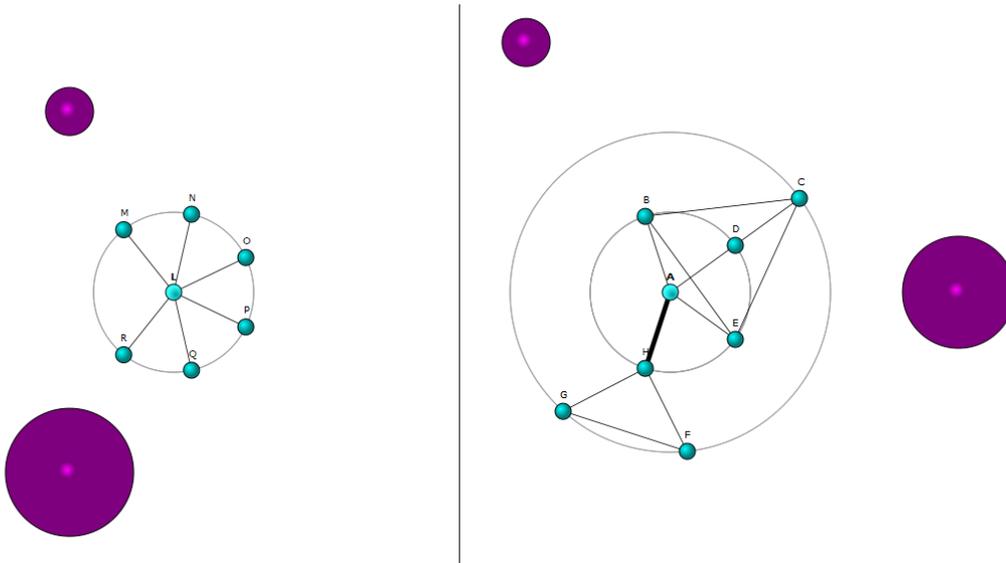


Figura 8. Mudança de componente em exibição radial.

tanodos num técnica como a do leiaute radial ou leiaute baseado em forças leva à necessidade de prover ao usuário a possibilidade de indicar o subgrafo a ser aglutinado e exibido como um metanodo e, é claro, sua consequente expansão, ou seja, indicar o metanodo que deve ser reexibido como um subgrafo seguinte as regras do *layout* geral.

III. ESTENDENDO O MAGNETVIZ COM LAYOUT RADIAL E MÚLTIPLAS VISÕES

A integração de leiautes alternativos ao MagnetViz levou à utilização do conceito de múltiplas visões coordenadas [9]. Assim, a interface básica do MagnetViz foi modificada com a introdução de duas visões adicionais àquela original do *layout*

baseado em forças. A nova interface é apresentada adiante, nesta seção (Figura 9).

O usuário pode iniciar a visualização do grafo com o *layout* baseado em forças, e pode optar pelo leiaute radial na segunda visão.

O *layout* radial se mostrou satisfatório para o componente do nodo seleccionado como foco. Entretanto, os componentes desconexos a ele nem sempre são dispostos de maneira facilmente distinguível – especialmente nos casos em que há muitos deles ou uma grande quantidade de nodos irmãos, resultando em nodos demais dispostos num mesmo nível, dificultando uma visualização prática.

Para contornar esse problema – e confiando na noção

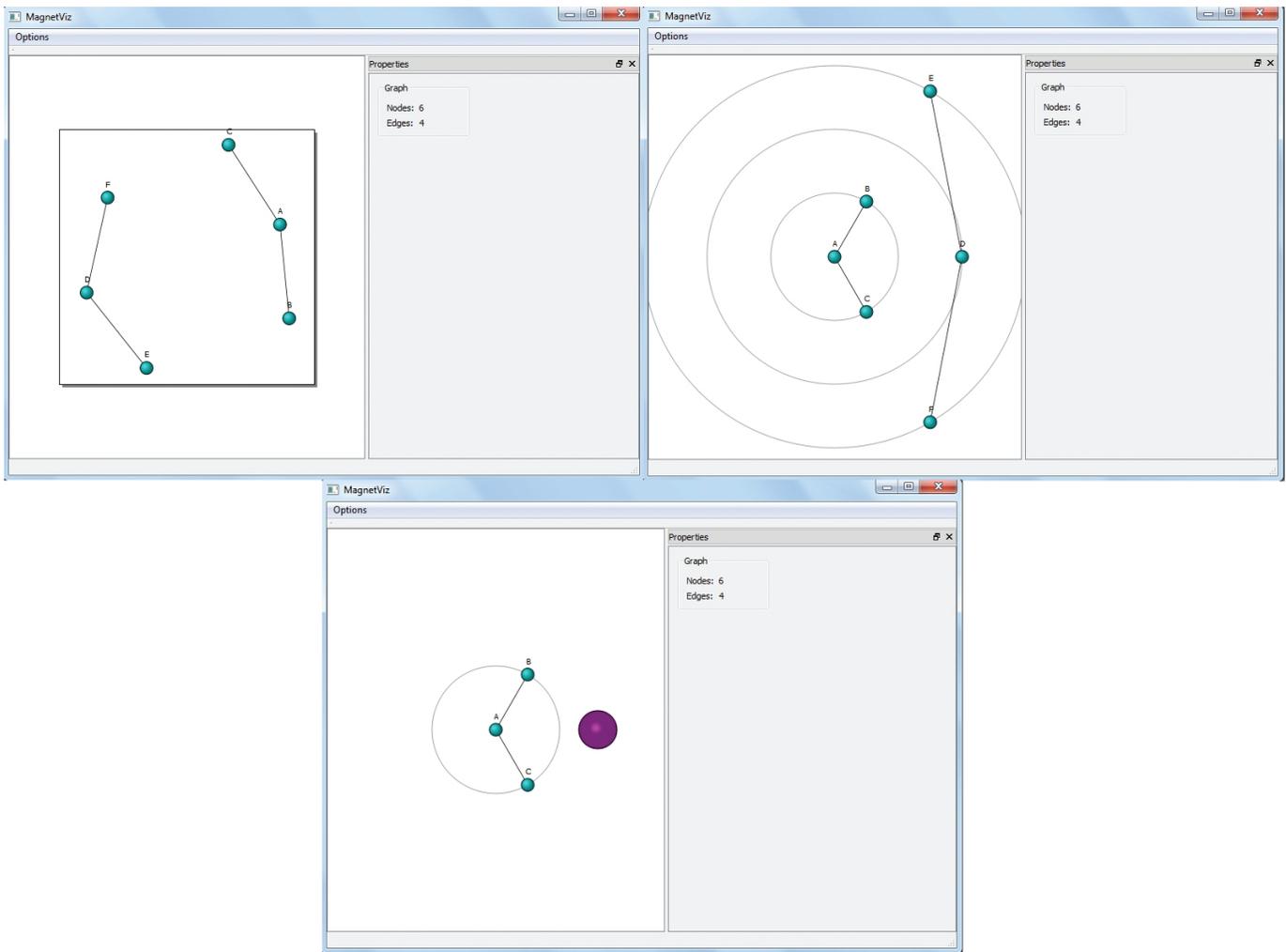


Figura 9. Interface com as três janelas.

intuitiva de que o usuário na maioria das situações estaria interessado em analisar a disposição de um componente por vez – chega-se ao terceiro *layout*, que se trata de uma versão do *layout* radial, com a adição de meta-nodos para representar os componentes conexos.

Focando principalmente no problema da escalabilidade (ou seja, exibir grafos muito grandes), foram propostas técnicas que utilizam um *layout* de grafo pré-computado e o utilizam para construir uma hierarquia que será explorada interativamente navegando através de meta-nodos recursivos, sendo os meta-nodos um agrupamento de nodos do grafo. Schaffer [10] utiliza uma abordagem multifocal onde meta-nodos podem ser expandidos e visualizados no contexto do grafo inteiro. Uma solução similar foi aplicada ao nosso *layout* radial.

Cada componente do grafo possui um meta-nodo, cujo tamanho é proporcional à sua quantidade de nodos. Inicialmente são exibidos apenas os meta-nodos dispostos em um círculo (Figura 6). Ao clicar em um deles, ele é expandido para o centro do grafo, onde o componente é exibido na forma radial, inclusive podendo-se escolher o nodo em foco (Figura 7).

Clicando-se em outro meta-nodo, o componente em exibição é recolhido para seu meta-nodo enquanto o novo entra em seu lugar, e assim por diante (Figura 8). Caso o grafo possua apenas um componente, ele estará sempre no centro, sem a exibição de nenhum meta-nodo.

Na implementação destas visualizações, a interface da ferramenta apresenta três janelas, cada uma com sua visualização (Figura 9). Na subseção *Properties* é exibida a contagem de nodos e arestas do grafo. Ao selecionar um nodo, são exibidos seus atributos, tais como seu rótulo, seu grau e sua quantidade de vizinhos. Ao selecionar uma aresta, são exibidos os nodos que ela liga e seu valor, se houver. No menu *Options* é possível inserir ímãs (na primeira janela), abrir arquivos de grafos, editar preferências de exibição e exportar a visualização para um arquivo de imagem. Da mesma maneira que a original, esta implementação é feita na linguagem C++ com o auxílio do *framework* Qt 4 versão 4.7.0.

IV. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma extensão da ferramenta MagnetViz, descrevendo os conceitos e o protótipo.

Embora tenha sido constatado que MagnetViz dispõe de técnicas poderosas para a exploração de grafos, especialmente com grande potencial no contexto de redes sociais, ainda há trabalho a ser feito. Já foram realizadas validações para as técnicas utilizadas em separado, mas não para todas integradas. Além disso, novas características podem ser acrescentadas para tornar a ferramenta mais poderosa.

Entre as ideias de trabalhos futuros estão a sincronização completa das visualizações, a escolha de com quais delas deve ser exibido um grafo a ser aberto, fusão das visualizações (substituindo meta-nodos por outro tipo de grafo, por exemplo), expansão das funcionalidades dos meta-nodos (como permitir que mais de um seja expandido ao mesmo tempo), além de acrescentar outras alternativas de visualização.

REFERÊNCIAS

- [1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing*. Upper Saddle River, NJ: Prentice Hall, 1999.
- [2] I. Herman, I. C. Society, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 24–43, 2000.
- [3] A. S. Spritzer and C. M. D. S. Freitas, "A physics-based approach for interactive manipulation of graph visualizations," in *AVI '08: Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA: ACM, 2008, pp. 271–278.
- [4] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
- [5] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, 2001, pp. 43–50.
- [6] J. Abello, F. van Ham, and N. Krishnan, "Ask-graphview: A large scale graph visualization system," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 669–676, Sept.-Oct. 2006.
- [7] D. Archambault, H. c. Purchase, and B. Pinaud, "The readability of path-preserving clusterings of graphs," *Computer Graphics Forum (Proc. EuroVis 2010)*, vol. 29, no. 3, pp. 1173–1182, 2010.
- [8] J. F. Rodrigues Jr., H. Tong, A. J. M. Traina, C. Faloutsos, and J. Leskovec, "Gmine: a system for scalable, interactive graph visualization and mining," in *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 1195–1198. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1164242>
- [9] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," in *Working Conference on Advanced Visual Interfaces*, 2000, pp. 110–119.
- [10] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, and M. Roseman, "Navigating hierarchically clustered networks through fisheye and full-zoom methods," *ACM Transactions on Computer-Human Interaction*, vol. 3, pp. 162–188, 1998.