

Propondo uma Arquitetura para Ambientes Interativos baseados em Múltiplas Visões

Arleson Nunes Silva, Glauco de Figueiredo Carneiro

Programa de Pós Graduação em Sistemas e Computação
Universidade Salvador (UNIFACS) – Salvador, BA – Brasil

arleson.nunes@unifacs.br, glauco.carneiro@unifacs.br

***Resumo.** Este artigo propõe uma arquitetura para ambientes interativos baseados em múltiplas visões (AIMV). Para esta finalidade são apresentados os objetivos e são discutidos os principais conceitos relacionados aos AIMVs. O artigo também relata o desenvolvimento de duas aplicações a partir do AIMV proposto. A primeira com a finalidade de apoio à compreensão de software. A segunda com a finalidade de apoio à análise do desempenho de redes de computadores. O desenvolvimento destas duas aplicações teve o objetivo de analisar a viabilidade da arquitetura proposta.*

1. Introdução

A visualização é um importante meio de compreensão e é fundamental para apoiar a construção de um modelo mental a respeito de uma determinada situação ou realidade (Spence, 2007). A visualização de informação pode ser mapeada como parte de um processo de compreensão com o objetivo de atingir um conhecimento aprofundado a respeito de um tema a partir de um conjunto de dados (Jacobson, 1999). Para cada tipo ou conjunto de dados podem ser selecionadas uma ou mais metáforas visuais para a sua análise. Estas metáforas podem ser então disponibilizadas em um ambiente interativo baseado em múltiplas visões (AIMV) (Baldonado, Woodruff e Kuchinsky, 2000).

Sistemas baseados em múltiplas visões têm sido propostos e utilizados para a análise de um vasto conjunto de tipos de dados. Na maioria das vezes esta análise requer mais de uma metáfora visual para que seja efetiva (Wang, Woodruff e Kuchinsky, 2000). Isto ocorre em função da dificuldade da representação de determinadas características e propriedades inerentemente complexas em uma única visão (Boukhelifa e Rodgers, 2003; Roberts, 2000; Becks e Seeling, 2004). Além disso, múltiplas visões incentivam a construção de conhecimento mais aprofundado a respeito dos dados analisados e evitam interpretações distorcidas que poderiam emergir de uma única visão (Ainsworth, 1999).

As próximas seções deste trabalho estão estruturadas da seguinte forma. A seção 2 descreve ambientes interativos baseados em múltiplas visões. A seção 3 apresenta a proposta de uma arquitetura para um AIMV. A seção 4 apresenta um relato do desenvolvimento de duas aplicações a partir da arquitetura do AIMV proposto. A seção 5 apresenta as considerações finais.

2. Ambientes Interativos baseados em Múltiplas Visões

Ambientes interativos baseados em múltiplas visões (AIMV) oferecem recursos e mecanismos de visualização para apoiar a análise de dados e também a descoberta e o reconhecimento de informação relevante (Spence, 2007; Shneiderman e Plaisant, 2009).

Estes ambientes utilizam mecanismos de coordenação entre as diversas visões que o compõem. As visões podem ser utilizadas de forma combinada para possibilitar a análise de um conjunto de dados através de diferentes perspectivas. Além disso, um AIMV oferece mecanismos de interação que permitem selecionar e ajustar como os dados serão analisados.

2.1. Objetivos de um AIMV

O objetivo de um AIMV é oferecer um grau de liberdade aos usuários para a execução de tarefas diferentes através de recursos de interação e também do uso combinado das representações visuais disponíveis e coordenadas entre si. Um AIMV deve ser flexível o bastante para permitir: (i) selecionar itens individuais ou subconjuntos de interesses a eles relacionados; (ii) manter o foco em determinados atributos; (iii) utilizar uma visão panorâmica ou detalhada de conjuntos de dados; (iv) navegar pelo conjunto de dados; e, por último, (v) controlar o mapeamento dos atributos reais dos dados nas representações visuais (North e Shneiderman, 2000).

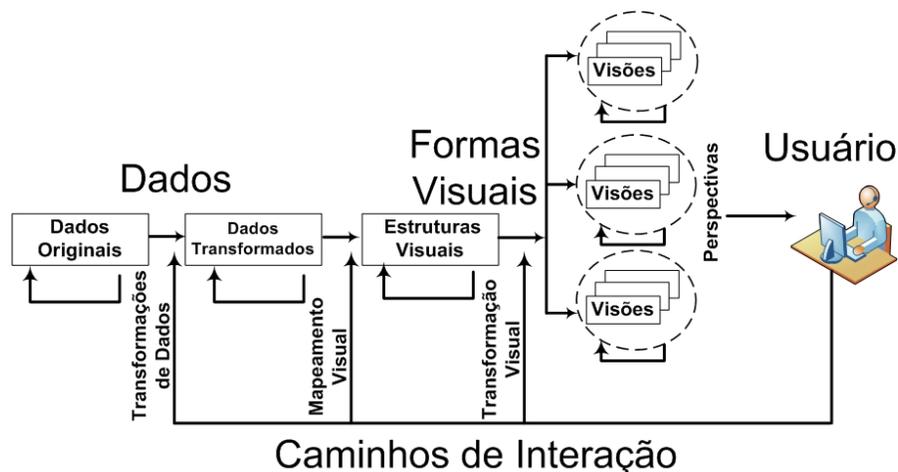


Figura 1: Um Modelo de Referência para AIMVs (Carneiro, 2011)

2.2. Um Modelo de Referência para AIMVs

Um modelo de referência para visualização de informação bastante conhecido foi proposto por Card, Mackinlay e Shneiderman (1999). De acordo com este modelo, a criação de visões ocorre através das seguintes etapas: transformação de dados, mapeamento visual e criação da visão (Card, Mackinlay e Shneiderman, 1999). Carneiro (2011) estendeu este modelo para adaptá-lo ao contexto de múltiplas visões e perspectivas (Figura 1). Uma perspectiva é um conjunto de visões coordenadas que representam um grupo de propriedades ou características específicas de uma situação, contexto ou entidade, geralmente utilizando diversas metáforas visuais (Carneiro, 2011). Este modelo adaptado possibilita o uso combinado dos recursos de um ambiente interativo e baseado em múltiplas visões. O modelo inicia-se com os dados originais obtidos de um repositório. Os dados então passam por um conjunto de transformações para serem organizados em estruturas de dados apropriadas para a exploração. Esta etapa corresponde à transformação dos dados conforme mostrado na Figura 1. Em seguida, estes dados são usados para montar as estruturas de dados visuais. Estas estruturas organizam as propriedades dos dados e propriedades visuais tais como

formas, cores e posições de forma a facilitar a construção das metáforas visuais. Esta etapa corresponde ao mapeamento visual mostrado na Figura 1. A última etapa, a transformação visual, tem o objetivo de representar na tela do computador as informações organizadas na etapa anterior.

2.3. Aplicações Desenvolvidas a partir de AIMVs

Atualmente existem diversas aplicações desenvolvidas a partir de AIMVs. Na área de Engenharia de Software (ES) existe, por exemplo, o Shrimp (Shrimp, 2012), utilizado para representação visual da arquitetura de software. Outro exemplo também em ES é o Tarantula (Tarantula, 2012), utilizado para representação visual de um conjunto de testes para uma determinada aplicação. Existem também exemplos de aplicações baseadas em AIMVs para a análise de dados em outras áreas. Por exemplo, aplicações como InfoZoom (Spenke e Beilken, 2000) e Polaris (Stolte, Tang e Hanrahan, 2002) podem ser utilizadas para a representação visual de dados tabulares com a finalidade de análise de padrões de distribuição.

Entretanto, o que se verifica na prática para a maioria destas aplicações é a dificuldade na inclusão de novos recursos tais como visões ou filtros de interação. Isto poderia auxiliar na configuração dos cenários visuais de forma adequada e compatível às necessidades de análise de um determinado conjunto de dados. Para atender a esta importante demanda, este artigo propõe uma arquitetura de AIMV que viabilize a expansão e customização de aplicações desenvolvidas a partir dela.

3. Propondo uma Arquitetura para um AIMV

Este artigo apresenta uma arquitetura com o objetivo de fornecer uma estrutura básica para o desenvolvimento de aplicações baseadas em múltiplas visões. O principal objetivo desta arquitetura é possibilitar a expansão e customização destas aplicações através da inclusão de novos recursos. Como exemplo, pode-se citar uma aplicação que possui somente uma metáfora visual que representa um conjunto de dados hierárquicos, como, por exemplo, a visão polimétrica (Lanza e Ducasse, 2003). Caso ela tenha sido desenvolvida a partir do AIMV proposto, há a possibilidade de inclusão de outras metáforas visuais para a representação do relacionamento de acoplamento entre os dados, como por exemplo uma visão baseada em grafos (Balzer e Deussen, 2007).

3.1.A Arquitetura Proposta

A arquitetura proposta tem como ponto de partida a arquitetura proposta por Carneiro (2011). Porém, ajustes foram feitos para que pudesse ser utilizada em diversos domínios de aplicações. A arquitetura tem como referência o modelo de desenvolvimento de Software MVC (Modelo, Visão e Controle), cujo objetivo é separar a lógica de negócio da apresentação ou representação visual (Gamma, *et al.*, 2000). Ela é composta por três camadas (Figura 2). Cada camada tem uma função específica dentro da arquitetura e utiliza um conjunto de componentes independentes para realizá-la.

A primeira camada é a Camada de Modelagem. Essa camada é responsável por importar e modelar os dados em estruturas que facilitem e auxiliem o seu uso pelos recursos da aplicação AIMV. Ela contém os módulos de importação e as estruturas de dados. Os módulos de importação são os componentes responsáveis pela importação e modelagem dos dados nas estruturas de dados, como ilustrado na Figura 2. Eles

recebem o conjunto original dos dados (seta 1) e realizam uma série de transformações para modelá-los (seta 2) de acordo com o objetivo da aplicação. As estruturas de dados são os componentes responsáveis pelo armazenamento dos dados tratados da fonte original. Elas são utilizadas para facilitar e auxiliar o uso dos dados pelos demais componentes da aplicação.

A segunda camada é a Camada de Controle. Ela é responsável pelo controle da aplicação e pela interação entre as estruturas visuais e as estruturas de dados. Ela contém os controladores, os filtros e as ferramentas. Os controladores são componentes responsáveis pelo controle dos eventos gerados pelos demais componentes da aplicação. Eles recebem os eventos (seta 3) e disparam ações de acordo com o tipo de evento recebido e também de acordo com o componente que o enviou. Por exemplo, quando um filtro é aplicado, ele envia um evento para o controlador responsável. Esse controlador vai receber este evento e vai solicitar a atualização (seta 4) das metáforas visuais. Os filtros são os recursos que permitem a filtragem dos dados. Eles atuam diretamente nas estruturas de dados (seta 5) e possibilitam selecionar interativamente um subconjunto de interesse do conjunto de dados. As ferramentas são os recursos que permitem a implementação das funcionalidades da aplicação. Elas também atuam diretamente nas estruturas de dados e podem ser utilizadas para interagir sobre os dados. Tanto os filtros quanto as ferramentas enviam seus eventos para os controladores.

A última camada é a Camada de Visualização. Essa camada fornece os recursos visuais do AIMV. Ela contém os paradigmas (metáforas visuais), as visões de filtragem e a barra de ferramentas. Os paradigmas (parte B da Figura 2) são representações visuais de um conjunto de dados. Eles são resultados da aplicação de uma metáfora visual para representar os dados armazenados nas estruturas de dados em estruturas visuais. As visões de filtragem (parte A da Figura 2) são as estruturas responsáveis pela criação dos componentes visuais que serão utilizados para aplicação dos filtros (seta 6). Elas fornecem um conjunto de controles para a execução das operações de filtragem. A barra de ferramentas (parte C da Figura 2) é responsável por fornecer os componentes visuais para aplicar ou ativar as ferramentas (seta 7). Ela armazena os atalhos que dão acesso às ferramentas.

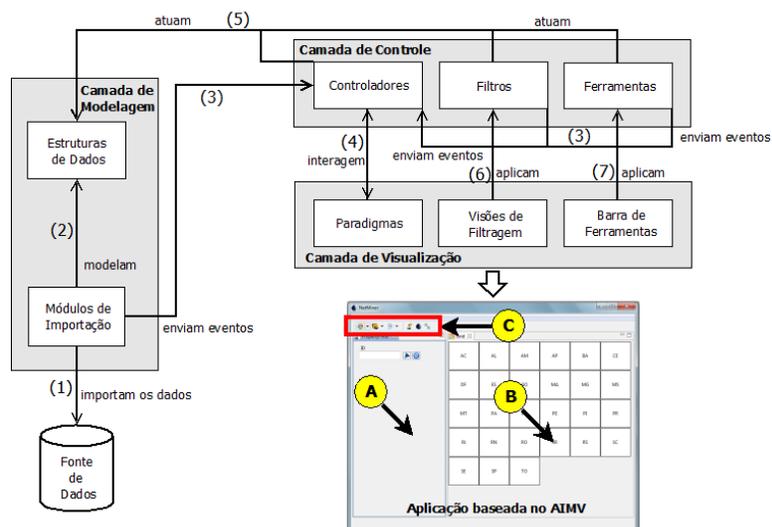


Figura 2: Arquitetura Proposta em Três Camadas para o AIMV

3.2. Estratégias de Desenvolvimento e Evolução

A arquitetura proposta foi desenvolvida como um plug-in do ambiente de desenvolvimento de software (ADS) Eclipse (Eclipse, 2012) usando a linguagem Java. Na concepção da arquitetura foram utilizados recursos de extensão e pontos de extensão do Eclipse (Eclipse, 2012). O objetivo é permitir que os componentes da aplicação possam ser instanciados em plug-ins diferentes. Com isso, aplicações desenvolvidas com base na arquitetura podem ser adaptadas para outros contextos utilizando apenas um conjunto de plug-ins já desenvolvidos. Além disso, os componentes podem ser reutilizados para diferentes objetivos. A Figura 3 representa a arquitetura do plug-in AIMV. Ele contém um conjunto de pacotes, classes e recursos internos que são responsáveis por fornecer as funcionalidades da arquitetura. Além disso, este plug-in oferece cinco pontos de extensão, como mostra a Figura 3. Estes pontos de extensão possuem um conjunto de regras e propriedades para a criação dos componentes da aplicação e facilitam a inclusão de novas funcionalidades.

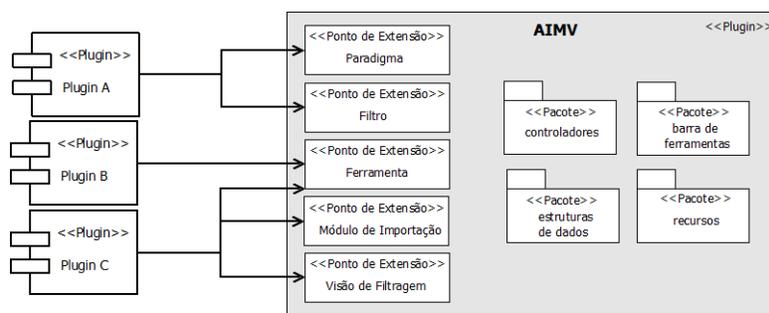


Figura 3: Estrutura do plug-in AIMV

4. Relatando o Desenvolvimento de Duas Aplicações a partir do AIMV Proposto

O objetivo desta seção é analisar a viabilidade do desenvolvimento de aplicações a partir da arquitetura proposta e utilizando a estrutura básica disponibilizada pelo plug-in AIMV mostrado na Figura 3. Para esta finalidade, será apresentado um relato do desenvolvimento pelo primeiro autor deste artigo de duas aplicações: o SourceMiner e o NetMiner conforme descrito a seguir.

4.1. SourceMiner

O SourceMiner¹ é um aplicação do tipo AIMV para representar através de metáforas visuais informações consideradas relevantes de uma aplicação Java (Carneiro, 2011). A aplicação foi desenvolvida como um plug-in do Eclipse com a finalidade de apoio a atividades de compreensão de software. As primeiras versões do SourceMiner tinham alguns de seus componentes com forte acoplamento, fato que tornava mais onerosa a inclusão de novas funcionalidades tais como novas metáforas visuais. Em alguns casos, isto implicou na necessidade de diversas modificações em cascata para que fosse possível a alteração da implementação de determinados componentes. Além disso, para o desenvolvimento de novas funcionalidades era necessário o acesso e uso do código fonte do SourceMiner.

¹ O SourceMiner está disponível em <http://www.sourceminor.org>.

A nova versão do SourceMiner desenvolvida a partir da arquitetura proposta do AIMV contornou estas questões. Além da implementação das funcionalidades já existentes, também foram tratadas oportunidades de melhorias identificadas em uma avaliação de usabilidade realizada por 15 usuários do SourceMiner. A Figura 4 mostra o relacionamento entre os plug-ins que compõem a nova versão do SourceMiner desenvolvido a partir da arquitetura proposta do AIMV.

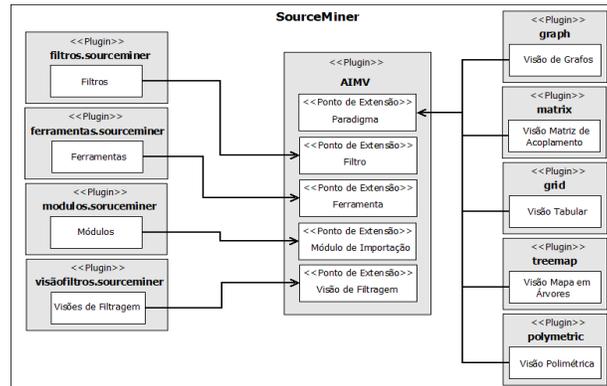


Figura 4: Nova Arquitetura do SourceMiner

O plug-in central *AIMV* da Figura 4 representa o mesmo plug-in indicado na Figura 3. Ele fornece a estrutura básica do AIMV e os pontos de extensão para instanciar os componentes da aplicação, nesse caso o SourceMiner. A partir daí, foram desenvolvidos outros plug-ins para a criação dos componentes. Cada plug-in foi responsável pela criação de um conjunto de componentes específicos do SourceMiner. Neste caso, um plug-in para criação dos filtros, outro para criação das ferramentas, outro dos módulos e por último um plug-in para as visões de filtragem, como mostra a parte esquerda da Figura 4.

Os demais plug-ins, apresentados à direita da figura, foram desenvolvidos para criação das visões. Foram implementadas as mesmas visões presentes na versão anterior do SourceMiner, porém com base na nova arquitetura. A visão de grafos (Balzer e Deussen, 2007), a visão polimétrica (Lanza e Ducasse, 2003), a visão mapa em árvores (Shneiderman, 1992), a visão matriz de acoplamento (Sangal, *et al.*, 2005) e a visão tabular (Carneiro, 2011), representadas respectivamente pelos plug-ins: *Graph*, *Polymetric*, *TreeMap*, *Matrix* e *Grid*. Como pode ser constatado, o SourceMiner é a aplicação resultante da integração de todos estes plug-ins ilustrados na Figura 4. Além disso, o plug-in AIMV fornece a estrutura básica da aplicação enquanto que os demais implementam as funcionalidades e instanciam os componentes do SourceMiner.

4.2.NetMiner

O NetMiner é um ambiente integrado baseado em múltiplas visões (AIMV) utilizado para apresentar visualmente as métricas de desempenho de redes de computadores. Este ambiente pode permitir a monitoração e a análise de forma visual dos principais atributos que afetam o desempenho das redes. Ele foi desenvolvido a partir da arquitetura proposta, utilizando o plug-in *AIMV*. A Figura 5 mostra a estrutura do NetMiner.

O plug-in central *AIMV* da Figura 5 representa o mesmo plug-in da arquitetura proposta mostrado na Figura 3. Ele fornece a estrutura básica do AIMV e os pontos de

extensão para instanciar os componentes da aplicação, nesse caso o NetMiner. De forma similar ao SourceMiner, foram desenvolvidos outros plug-ins para a criação dos componentes. Cada plug-in foi responsável pela criação de um conjunto de componentes específicos do NetMiner. Além disso, o NetMiner utiliza as mesmas visões utilizadas no SourceMiner. A diferença é que no NetMiner estas visões são utilizadas com um objetivo diferente. O foco agora é no uso de um ambiente interativo baseado em múltiplas visões para a análise das métricas de desempenho de redes de computadores. O NetMiner consiste então na integração de todos esses plug-ins mostrados de acordo com o que é apresentado na Figura 5.

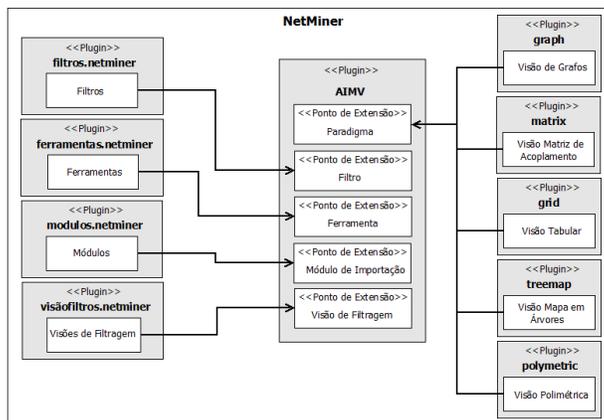


Figura 5: Arquitetura do NetMiner

5. Considerações Finais

Sistemas baseados em múltiplas visões podem ser utilizados para a análise de um vasto conjunto de tipos de dados. Atualmente existem diversas aplicações desenvolvidas a partir de AIMVs em diversas áreas de conhecimento. Entretanto, o que se verifica na prática para a maioria destas aplicações é a dificuldade na inclusão de novos recursos. Neste contexto, o artigo apresentou uma arquitetura que tem como objetivo fornecer uma estrutura básica para o desenvolvimento de aplicações baseadas em múltiplas visões. Esta arquitetura visa possibilitar a expansão e customização destas aplicações através da inclusão de novos recursos. O estudo de caso relatado ilustra o desenvolvimento de duas aplicações para a análise de dois tipos de conjuntos de dados diferentes a partir da arquitetura proposta de AIMV. Os resultados preliminares indicam evidências iniciais que a arquitetura é efetiva para o desenvolvimento de aplicações interativas baseadas em múltiplas visões. Neste momento os autores estão planejando dois estudos. O primeiro com foco na avaliação de usabilidade das versões atuais do SourceMiner e NetMiner. O segundo estudo terá como foco a efetividade do uso destas aplicações para atividades nas áreas de engenharia de software e redes de computadores.

Referências

- Spence, R. Information Visualization: Design for Interaction (2nd Edition). 2. ed. Prentice Hall, 2007.
- Jacobson, I.; Booch, G.; Rumbaugh, J. The Unified Software Development Process. Addison-Wesley Professional, 1999.
- Baldonado, M., Woodruff, A., Kuhinsky, A. Guidelines for Using Multiple Views in Information Visualization, In ACM AVI 2000; Palermo, Italy. 110-119.

- Wang, M. Q.; Woodruff, A.; Kuchinsky, A. Guidelines for using multiple views in information visualization. Proceedings of the working conference on Advanced visual interfaces. ACM. 2000. p. 110-119.
- Boukhelifa, N.; Rodgers, P. J. A model and software system for coordinated and multiple views in exploratory visualization. Information Visualization, v. 2, p. 258-269, 2003.
- Roberts, J. C. Multiple-View and Multiform Visualization. IS&T and SPIE Visual data exploration and analysis. 2000. Vol. 3960 p. 176-185.
- Becks, A.; Seeling, C. SWAPit: a multiple views paradigm for exploring associations of texts and structured data. Proceedings of the Working Conference on Advanced Visual Interfaces. ACM. 2004. p. 193-196.
- Ainsworth, S. The functions of multiple representations. Comput. Educ., v. 33, p. 131-152, 1999.
- Shneiderman, B.; Plaisant, C. Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition), 5th ed. Addison Wesley, 2009.
- North, C.; Shneiderman, B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. Proceedings of the Working Conference on Advanced Visual Interfaces ACM. 2000. p. 128-135.
- Card, S. K.; Mackinlay, J. D.; Shneiderman, B. (Eds.). Readings in information visualization: using vision to think. Morgan Kaufmann Publishers Inc., 1999.
- Carneiro, G. F. Sourceminer: um ambiente integrado para Visualização multi-perspectiva de software. 2011. 230 f. Tese (doutorado) – Universidade Federal da Bahia, Instituto de Matemática, Doutorado Multiinstitucional em Ciência da Computação.
- Tarantula software. <http://pleuma.cc.gatech.edu/aristotle/Tools/tarantula/index.html>, Maio. 2012.
- Shrimp software. <http://www.thechiselgroup.org/shrimp>), Maio. 2012.
- Spenke, M.; Beilken, C. InfoZoom: Analysing Formula One racing results with an interactive data mining and visualisation tool. In Proceedings of 2nd International Conference on Data Mining, pages 455–464, Cambridge University, UK, 2000.
- Stolte, C.; Tang, D.; Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. IEEE Transactions on Visualization and Computer Graphics, pages 52–65, 2002.
- Eclipse. <http://www.eclipse.org/>, Maio. 2012.
- Balzer, M.; Deussen, O. Hierarchy Based 3D Visualization of Large Software Structures. Proceedings of the Conference on Visualization. 2004. p. 4-14.
- Lanza, M.; Ducasse, S. Polymetric Views-A Lightweight Visual Approach to Reverse Engineering. IEEE Trans. Softw. Eng., v. 29, p. 782-795, 2003.
- Shneiderman, B. Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph., v. 11, p. 92-99, 1992.
- Sangal, N. et al. Using Dependency Models to Manage Complex Software Architecture. Proceedings of the 20th Annual ACM SIGPLAN Conference on Object Oriented Programming, Systems, Languages, and Applications. 2005. p. 167-176.
- Gamma, Erich et al. *Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos*. Porto Alegre: Bookman, 2000.