

DESENVOLVIMENTO DE FERRAMENTAS EM AMBIENTE LINUX PARA VISUALIZAÇÃO DE FLUXO DE ÁGUA SUBTERRÂNEA

José Eduardo Quaresma¹; Jefferson Nascimento de Oliveira² & Edson Wendland³

Resumo - O presente trabalho apresenta o desenvolvimento de um software visualizador de processos em aquíferos chamado VPA, como módulo integrante de um pacote de programas computacionais, aplicados ao gerenciamento de recursos hídricos subterrâneos, denominado SPA (Simulação de Processos em Aquíferos). Esse visualizador foi desenvolvido em ambiente *KDE 3* integrado ao sistema operacional *SuSE Linux*. A linguagem *Borland Kylix 3* foi escolhida para programação e desenvolvimento do visualizador. Este ambiente de programação é distribuído nos âmbitos de *open source development (GNU – General Public License)* (MANZANO, 2000), para programação em C++, orientada a objetos - “*Object Oriented Programming - OOP*” e serve como complemento da interface gráfica com o usuário - “*Graphic User Interface - GUI*”. Em conjunto com esta linguagem de programação foram usadas as bibliotecas do visualizador científico *Data Explorer IBM OpenDX* (EARNSHAW, 1993), distribuído também nos termos de *open source development (GNU)*, pela *IBM*, como ferramenta para visualização de simulações em duas e três dimensões. O software desenvolvido permite visualizar os resultados de simulação de processos de fluxo de fluidos e transporte de poluentes no subsolo, gerados pelos módulos de *pre and main processing* do pacote SPA, e arquivados na forma de dados binários.

Abstract - This work describes the development of a software for visualization of processes in aquifers (VPA), as integral module of a computational programs package applied to the management of groundwater resources, denominated SPA (Simulation of Processes in Aquifers). The visualizator was developed in the *KDE 3* environment integrated into the operating system *SuSE Linux*. The language *Borland Kylix 3* was chosen for programming and development of the visualizator. This programming environment provides a *Graphic User Interface (GUI)* for development of *Object Oriented Codes (OOC)* in C++. Together with this programming

¹ Departamento de Hidráulica e Saneamento, Escola de Engenharia de São Carlos, Universidade de São Paulo
Caixa Postal 359, São Carlos-SP, 13560-970, e-mail: quaresma@sc.usp.br

² Departamento de Engenharia Civil, Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista
Alameda Bahia N° 550, Ilha Solteira-SP, 15385-000, e-mail: jeffno@dec.feis.unesp.br

³ Departamento de Hidráulica e Saneamento, Escola de Engenharia de São Carlos, Universidade de São Paulo
Caixa Postal 359, São Carlos-SP, 13560-970, e-mail: ew@sc.usp.br

environment, libraries of the scientific visualizator OpenDX (ERNSHAW, 1993), developed by IBM, were used for visualization of simulation in two and three dimensions. The developed software allows the visualization of simulation results of fluid flow and pollutants transport in the underground, generated by the package SPA in the form of binary data.

Palavras-chave - Aquífero, Simulação, Visualização Científica.

INTRODUÇÃO

Os recursos hídricos subterrâneos constituem aproximadamente 97% de toda a água doce disponível no planeta. Sua utilização pelo ser humano é remota, havendo indícios de seu uso, pelos chineses, no ano 5000 a.C., REBOUÇAS (1999), LEAP (1999) e FEITOSA & MANUEL FILHO (2000). O Brasil dispõe de um volume de 112.000 quilômetros cúbicos de água em suas reservas subterrâneas, FEITOSA & MANUEL FILHO (2000). Conforme dados do DEPARTAMENTO DE ÁGUAS E ENERGIA ELÉTRICA – DAEE (1999), 61% da população brasileira é abastecida por água subterrânea: 43% por poços tubulares, 12% por fontes ou nascentes e 6% por poços escavados (tipo Amazonas). No Estado de São Paulo, segundo a COMPANHIA DE TECNOLOGIA DE SANEAMENTO AMBIENTAL – CETESB (1997), 72% das cidades utilizam-se de água subterrânea no sistema de abastecimento público, sendo 48% abastecidas exclusivamente por esse manancial. Além disso, 90% das indústrias também usam o recurso hídrico subterrâneo. De acordo com o DAEE (2000), no Estado de São Paulo, dos 100 bilhões de m³/ano correspondentes ao escoamento total, 41 bilhões, ou 1.285 m³/s, são oriundos do escoamento básico, parcela responsável pela regularização dos rios. Neste contexto, uma preocupação dos órgãos responsáveis pelo gerenciamento da água refere-se à proteção e ao uso racional da água subterrânea, sendo imprescindível a coleta e gerenciamento de dados obtidos em campo. Como enfatizaram ENGEL & NAVULUR (1999) e GOGU et al. (2001), a disponibilidade dos dados de um sistema aquífero torna viável o estudo da variabilidade espacial e temporal desse sistema, podendo ser acoplado a programas de modelação numérica para estimativa de cenários futuros. Modelar numericamente aquíferos e analisar a variabilidade espacial do fluxo vem sendo alvo de muitos estudos nas últimas décadas, quando uma grande quantidade de modelos matemáticos foi criada e simulações foram propostas (KRESIC, 1997).

Para visualização dos resultados de simulação desta natureza, foi criado o VPA. Este Visualizador de Processos em Aquíferos (VPA), é resultado de uma pesquisa que estuda estratégias e algoritmos para mapear informações em representações gráficas. Com isso, possibilita uma

melhor compreensão do conteúdo de grandes conjuntos de dados (*Arrays*) e dos fenômenos que os geram.

O desenvolvimento tecnológico dos equipamentos aliado às novas técnicas de simulação, gerou a necessidade de ferramentas para visualização no processo de interpretação das informações auxiliando no gerenciamento de recursos hídricos subterrâneos.

Neste trabalho descreve-se a estrutura do VPA, destacando os conceitos sobre visualização científica, os sistemas disponíveis para visualização e as ferramentas que auxiliaram no desenvolvimento do referido software.

CONCEITOS SOBRE VISUALIZAÇÃO CIENTÍFICA

É importante ressaltar que o objetivo da visualização científica no campo do gerenciamento de recursos hídricos subterrâneos. A técnica de visualização envolve vários passos: a construção do modelo a ser visualizado a partir dos dados coletados em campo; a seleção de algum mecanismo de mapeamento do modelo em um objeto de visualização abstrato, como uma imagem ou um mapa de contornos; e a renderização da imagem no dispositivo de exibição gráfico. A visualização envolve, portanto, a exploração, transformação e mapeamento de dados em objetos gráficos. Esses dados podem estar definidos sobre domínios bidimensionais (2D), tridimensionais (3D) e multidimensionais. Em geral eles estão organizados em uma malha (regular ou irregular), mas podem se constituir de um conjunto de pontos esparsos. As informações associadas a cada ponto podem ser um ou mais valores escalares, vetoriais ou tensoriais, e os dados podem ser estáticos ou variantes no tempo.

A construção de sistemas para visualização é bastante complexa, devido aos requisitos de manipulação de conjuntos de dados volumosos, à implementação de algoritmos geométricos complexos, e à necessidade de interação com objetos multidimensionais. Além disso, a estrutura e a interface dos sistemas precisam considerar as necessidades dos profissionais que o manipulam, admitir orientação a objetos, admitir automação das tarefas que podem ser realizadas sem a interferência do usuário, permitir apresentação de dados em tempo real e incluir técnicas avançadas de interface com o usuário, como o uso de *Front End* (interface com usuário) desenvolvidos em linguagens orientadas a objetos (MINGHIM, 1997).

SISTEMAS DE VISUALIZAÇÃO

Padrões gráficos, como o *GKS* (*Graphical Kernel System*) (ANSI, 1985a) e o *PHIGS* (*Programmer's Hierarchical Interactive Graphics System*) (ANSI, 1985b), tiveram importante

papel na década de 80, inclusive ajudando a estabelecer conceito do uso de padrões mesmo fora da área gráfica, tendo sido implementado em diversas plataformas. Nenhuma destas *APIs* (*Application Programming Interface*), no entanto, conseguiu ter grande aceitação (SEGAL, 1994).

A interface destinada a aplicações gráficas 2D e 3D deve satisfazer diversos critérios como, por exemplo, ser implementável em plataformas com capacidades distintas sem comprometer o desempenho gráfico do hardware e sem sacrificar o controle sobre as operações de hardware (SEGAL, 1996).

A maioria dos sistemas de visualização admite tipos de dados para armazenar malhas regulares e irregulares, bem como a maioria das técnicas clássicas de visualização. A principal diferença entre eles é a disponibilidade de ferramentas para configurar o sistema, sua arquitetura, e a interface. Generalidade e dedicação a uma aplicação específica também são aspectos fundamentais.

A seguir são apresentados alguns dos sistemas mais conhecidos, incluindo bibliotecas, sistemas *turnkey* e geradores de aplicação.

Bibliotecas

As bibliotecas representam conjuntos de rotinas na forma de *APIs* (*Application Programming Interface*) que permitem ao usuário introduzir componentes de visualização nos seus programas.

Alguns exemplos são:

Visual3 (MIT): permite acesso, através de rotinas escritas em FORTRAN, à maioria das principais funções gráficas, de visualização e de exploração de dados volumétricos.

CardinalVision (Wilsonville, EUA): fornece uma biblioteca de objetos escritos em C++ para executar funções de visualização de dados escalares, vetoriais e tensores associados a malhas de elementos finitos.

FOCUS (*Visual Kinematics*, EUA): possui um conjunto de rotinas para várias técnicas gráficas e de visualização de sólidos, fluidos, e outros dados.

VTK (*Visualization Toolkit*): é uma biblioteca completa (SCHROEDER, 1996-97), disponível a baixo custo e com código fonte, implementada para sistemas *Unix*(*Linux*) e *Windows*, possibilitando a programação na linguagem *Tcl/Tk*. Também pode ser classificado como um gerador de aplicação.

Sistemas “Turnkey” (Pacotes Comerciais)

Em sistemas desse tipo não há necessidade de programação. Aqui se incluem sistemas de propósito geral e sistemas específicos para um domínio em particular, por exemplo, escoamento de fluidos. Alguns deles são:

FieldView (Intelligent, EUA): ambiente de visualização para dinâmica dos fluídos, fornece principais funções para planos de corte, superfícies de nível e traçado de partículas, com saídas que incluem animações e vídeo.

Data Visualizer (Wavefront Technologies, EUA): ambiente para visualização de dados de elementos finitos generalizados. Permite a utilização simultânea de várias técnicas de visualização.

SSV (Sterling Software, EUA): fornece um grande conjunto de produtos e serviços de visualização, inclusive o *software FAST (Flow Analysis Software Toolkit, University of Geórgia, EUA)*, que possui vários módulos para visualização escalar e vetorial em malhas híbridas, sob o comando do usuário, através de interfaces gráficas.

Outros pacotes que adotam a abordagem *turnkey* são *SciAn (Supercomputer Computations Res. Inst., gratuito)*, *Xdataslice (NSCA, gratuito)*, *Volvis (Stonybook, New York, gratuito)*, *GLVware (Army High Performance Comp. Res. Center, gratuito)*, *Vis-5D (Bill Hibbard, visualizador de mapas de previsão do tempo, gratuito)*, além de muitos outros, de aplicação específica ou de propósito geral (ELVINS, 1994).

Geradores de Aplicação (Application Builders)

Os geradores de aplicação permitem que o usuário crie um sistema de visualização “personalizado”. Além disso, eles admitem que usuários escrevam código em módulos próprios para tarefas de visualização, incluindo-os no contexto dos demais módulos pré-programados. Alguns dos mais conhecidos são:

AVS (Advanced Visual Systems, EUA): dividido em quatro módulos (um visualizador de imagens, um de geometria, um de volume e um editor de rede), possui interface visual para o projeto da rede de objetos (UPSON, 1992). Tem sido progressivamente transportado para muitas plataformas (HERNDON et al., 1994). Seu único problema é o custo elevado.

IBM Visualization Data Explorer (EARNSHAW, 1993): era um sistema comercializado pela *IBM*, agora é de distribuição gratuita, (*GNU*). Segue a mesma filosofia de fluxo de dados (saídas de um módulo são entradas de outros, esses recursos são chamados de *pipeline*) dos demais nessa classe.

Possui um editor visual de programas e várias ferramentas interativas para exploração dos dados. Também permite ao usuário a criação dos seus próprios módulos, que podem ser compartilhados em rede. Suas bibliotecas podem ser usadas para programação, porque possui código aberto. A Figura 1, apresenta um exemplo de visualização de cargas hidráulicas em um aquífero:

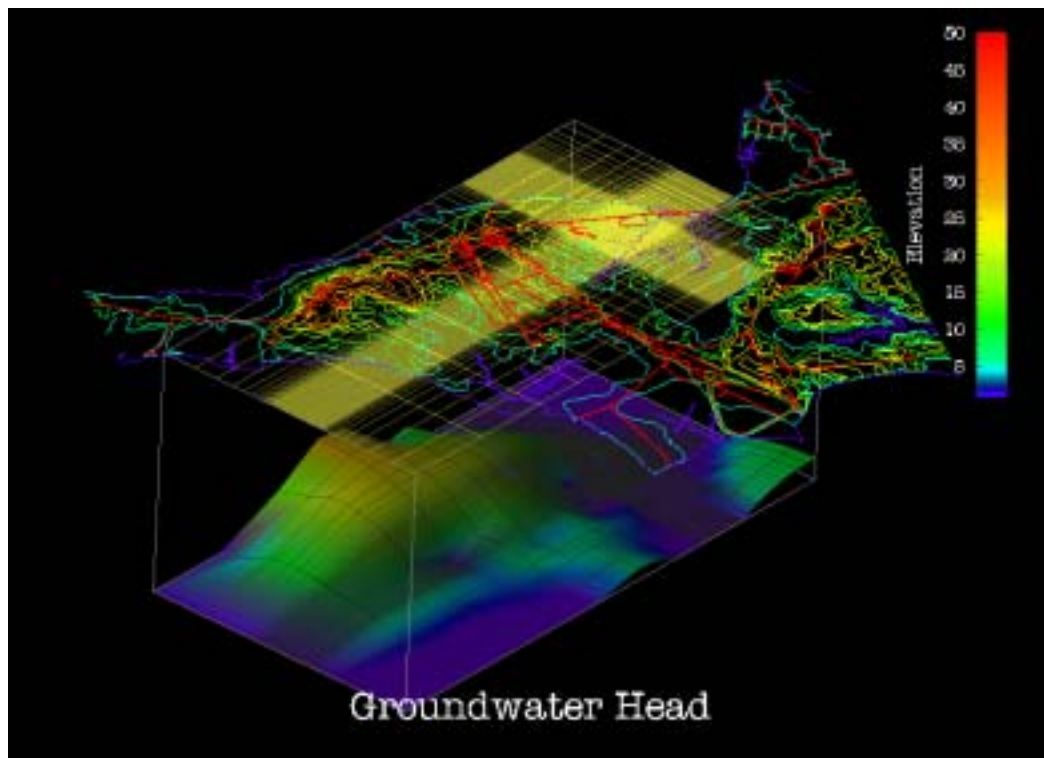


Figura 1 – Visualização gerada pelo OpenDX (Distribuição de Carga Hidráulica)

IRIS Explorer: permite selecionar módulos pré-compilados numa abordagem de seleção visual simples através do mouse. Módulos foram construídos em C e FORTRAN, e são selecionados através de um dos componentes do *software*: o *Librarian*. Os outros dois componentes são um Editor de Mapas, que permite editar a rede de módulos, e o *DataScribe*, responsável pela conversão de formatos entre o Explorer e outros programas. Era fornecido gratuitamente com máquinas *Silicon Graphics*, mas atualmente é comercializado pela NAG (NAG's IRIS Explorer, 2003).

Khoros (YOUNG, 2000): inclui algumas facilidades de programação visual. Possui geradores para estender a linguagem e para adicionar novos pacotes de aplicação ao sistema. É executado em sistemas *Unix (Linux)* e é gratuito.

PROCESSO DE VISUALIZAÇÃO

Os sistemas de visualização devem ser usados como parte integrante do processo de investigação científica. Assim, o processo de investigação requer a criação de um modelo empírico da “realidade”, a aquisição de dados que sirvam como subsídios para verificar ou refinar o modelo, e o uso desse modelo para fazer previsões. No ciclo de investigação o cientista usa o seu discernimento e conhecimento (*insight*) sobre o fenômeno para formular uma hipótese, coleta de dados, controlar um processo de computação, visualizar e interpretar os resultados e, em seguida, repetir e refinar o ciclo, como ilustrado na Figura 2.



Figura 2 – Análise científica e *feedback* com o auxílio da visualização

Os sistemas de visualização também devem ser caracterizados por um procedimento interativo iniciado pelo usuário, que continua o processo de investigação à medida que recebe *feedback* do sistema. A interação realizada pelo usuário, através da interface ao qual é apresentado, é transformada logicamente de volta aos termos do fenômeno científico sob investigação, e resultados atualizados são gerados pelo sistema de visualização e apresentados ao usuário. Dessa forma, o usuário aumenta o seu conhecimento sobre a validade da hipótese originalmente formulada e sobre o modelo adotado para a sua investigação.

Dados podem ser obtidos de duas maneiras: medições experimentais através da amostragem de processos empíricos, ou simulações computacionais. A consequência é que os dados são conhecidos apenas em pontos discretos. Isso cria um sério problema, pois uma das atividades relevantes em visualização é justamente determinar valores dos dados em pontos arbitrários.

Outro problema típico do processo de visualização é que os dados de entrada geralmente contêm erros aleatórios introduzidos pelo próprio processo de aquisição. Os dados precisam ser “filtrados” para que o erro seja removido, mas o processo de filtragem pode “distorcê-los”, removendo características de interesse ou incluindo informações espúrias. Finalmente, os processos experimentais podem ser conduzidos erradamente, gerando dados errados. Esses problemas relacionados à amostragem e à interpolação, que devem ser considerados no contexto do processo científico, refletem-se como limitações a serem tratadas também nas técnicas e sistemas de visualização. É importante ter em mente que esses sistemas trabalham com modelos computacionais empíricos dos fenômenos em estudo, e não com modelos matemáticos ideais.

DESENVOLVIMENTO DO VISUALIZADOR

O Visualizador de Processos em Aquíferos (VPA), é modulo integrante de um pacote computacional denominado SPA - Simulação de Processos em Aquíferos (WENDLAND, 2003), aplicado ao gerenciamento de recursos hídricos subterrâneos.

Esse visualizador foi desenvolvido com auxílio de técnicas computacionais avançadas. Para execução deste software, foram usados os sistemas operacionais *SuSE Linux* (SuSE User Guide, 2002) ou *FreeBSD (Unix)*, para interface *KDE 3* e *GNOME*, ambos disponibilizados gratuitamente no âmbito de *open source development*.

O *Front End* (interface com usuário), os códigos e as rotinas do visualizador, foram desenvolvidos com a linguagem de programação *Kylix 3* (SONNINO, 2001) que é uma ferramenta *RAD (Rapid Application Development)*. Este pacote desenvolvido pela *Borland*, integra as linguagens de programação *Delphi* (ALVES, 2002) e *C++* (BRUCE ECKEL, 2000) para o desenvolvimento de aplicações de banco de dados, *GUI*, *Web* e *Web Services* no *Linux*. A versão do *Kylix 3* inclui suporte expandido para o desenvolvimento de *Web Services* e compatibilidade com os últimos padrões dessa tecnologia. As soluções *RAD* da *Borland*, como linguagem de desenvolvimento *C++* é adotada pela maioria dos desenvolvedores *Linux*. O *Kylix 3* (SONNINO, 2001) é distribuído nos termos de *open source development*, e pode ser combinado com o *Borland C++ Builder* ou *Delphi* para criar aplicações *Linux* ou *Windows*, tornando-o assim multiplataforma e de fonte única. Os componentes para pré-construção *CLX (Component Library for Cross-platform development)* são personalizáveis e extensíveis, para o desenvolvimento de aplicações sofisticadas que simplificam a migração de aplicações baseadas em *Windows* para a plataforma *Linux* (SuSE Administration Guide, 2002) ou vice-versa. A figura 3 apresenta o *Front End* do *Kylix 3 (C++)*.

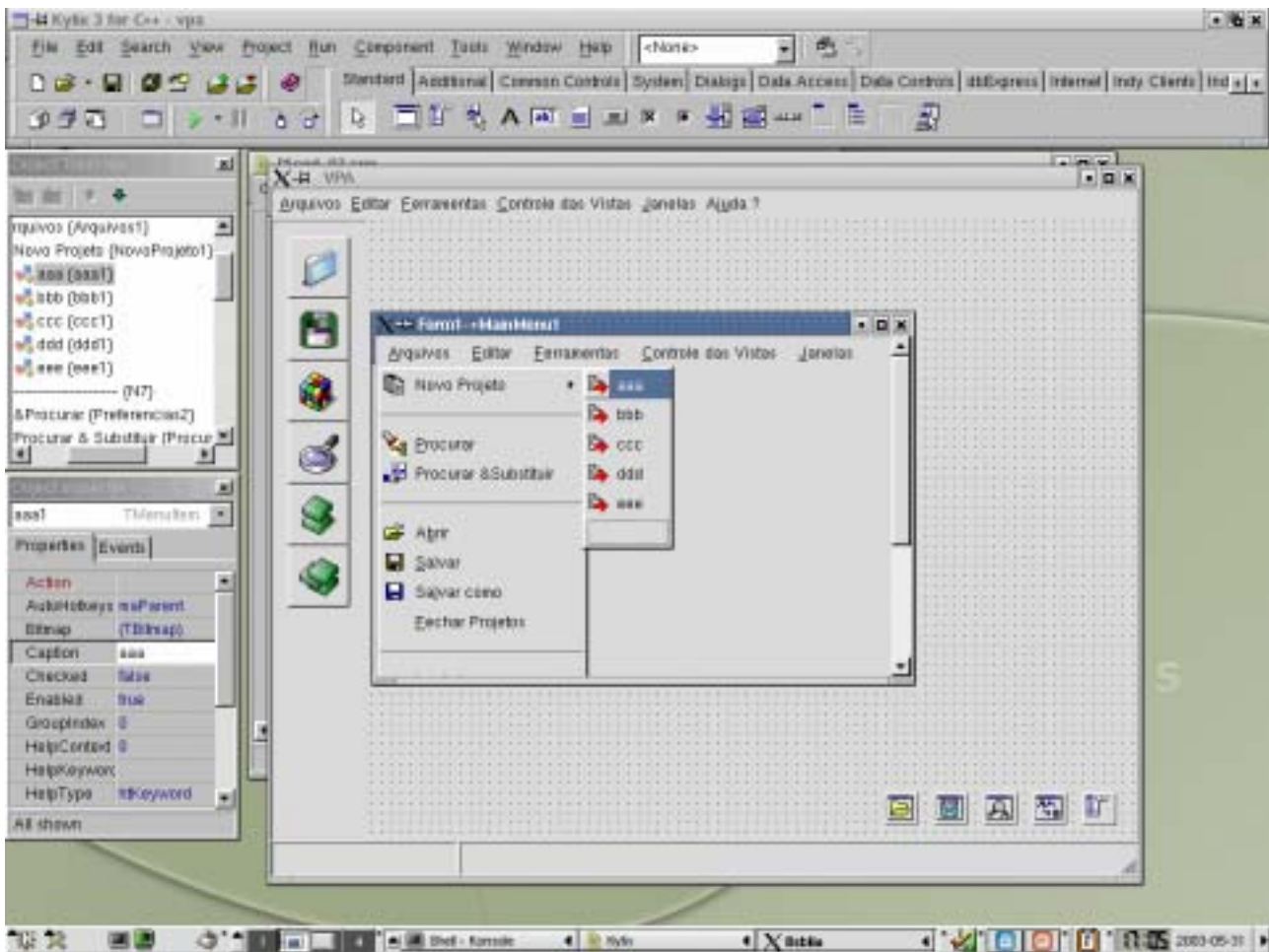


Figura 3 – *Front End* da Linguagem de Programação *Kylix 3* usada no desenvolvimento do VPA

A integração do SPA com VPA foi realizada por uma interface, desenvolvida em C++ (*Kylix 3*), com o auxílio de bibliotecas do SPA e do *OpenDX*. As simulações geradas pelo SPA (Simulação de Processos em Aquíferos) (WENDLAND, 2003), são convertidas em arquivos binários pela biblioteca *DtnLib*, para serem visualizadas no VPA (Visualizador de Processos em Aquíferos) com auxílio das bibliotecas do *OpenDX*. A figura 4 exemplifica com maior clareza a estrutura do visualizador.

RESULTADOS DO VPA

Os testes para avaliação do software VPA foram baseados na visualização das simulações do modelo de gerenciamento para o Aquífero Bauru (OLIVEIRA, 2002), capaz de reproduzir a complexidade e peculiaridades do sistema hidrogeológico, e pode ser considerado um avanço no campo da modelação regional.

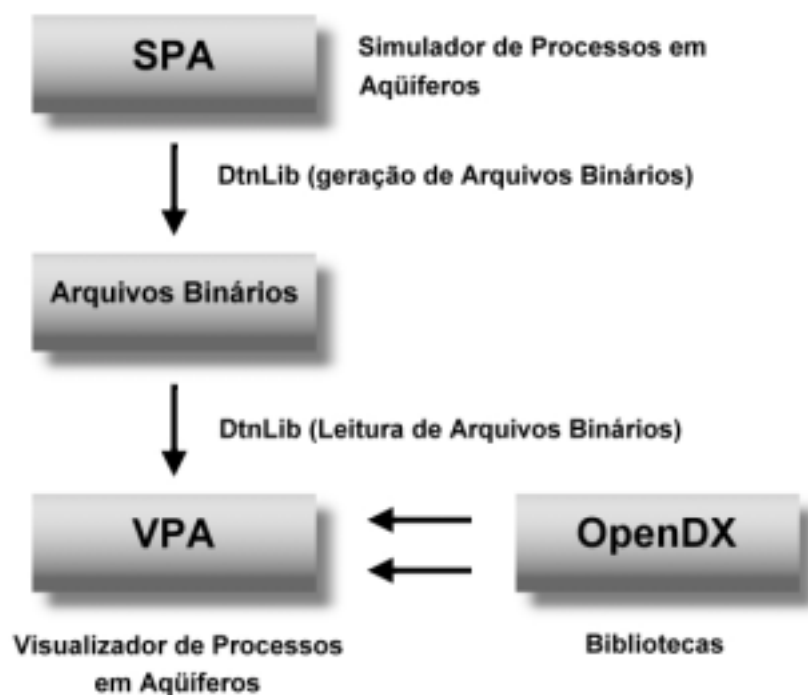


Figura 4 – Estrutura do Visualizador de Processos em Aquíferos

As visualizações do comportamento do modelo quanto à carga hidráulica aparecem como contribuição do presente trabalho ao setor de gestão da água subterrânea, para que se possa inferir a respeito do comportamento hidráulico do modelo desenvolvido.

O visualizador desenvolvido demonstrou ser útil ao gerenciamento de recursos hídricos, tendo incrementado o conhecimento das condições do fluxo na região.

A figura 5 apresenta o resultado final da visualização do potencial de carga hidráulica.

DISCUSSÃO

A visualização científica é uma área de pesquisa que apresentou uma grande evolução nos últimos anos. No entanto, muitos problemas importantes ainda não foram resolvidos. Do ponto de vista dos conjuntos de dados, são necessárias estratégias para manipular esses conjuntos cada vez maiores, já atingindo a dimensão de *terabytes*. Novas estratégias de interface com o usuário são necessárias para permitir a navegação no campo visual de informação dos conjuntos de dados, incluindo interfaces multimodais, que usam mais de uma forma de representação. A geração de imagens em tempo real que permite rápida interpretação ainda apresenta sérios problemas no seu processamento.

A perspectiva de desenvolvimento da área é bastante promissora para os próximos anos. Do ponto de vista científico e tecnológico, pode representar uma possibilidade de independência

tecnológica para futuras gerações que se interessem em investigar a área de simulação e visualização de processos físicos em aquíferos. Quanto ao aspecto econômico, a disponibilidade de um visualizador com tecnologia nacional, evitaria a transferência de divisas para o mercado internacional.

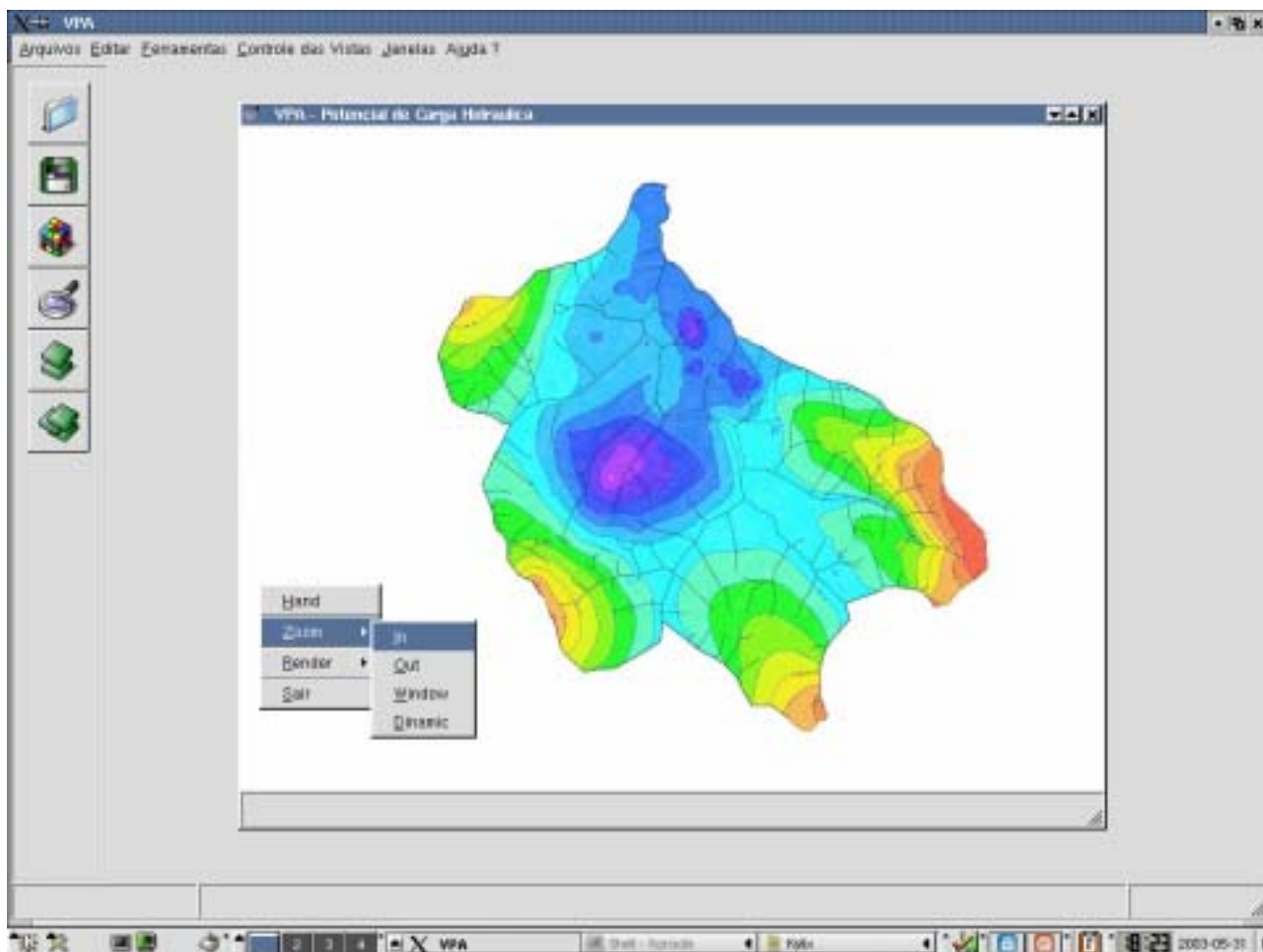


Figura 5 – Visualização do Potencial de Carga Hidráulica do Aquífero Bauru

CONCLUSÃO

Este trabalho descreve o desenvolvimento de um software visualizador de processos em aquíferos chamado VPA, para ambiente *KDE 3* ou *GNOME* integrado ao sistema operacional *SuSE Linux*. Sua interface foi desenvolvida no ambiente de programação *Kylix 3* (C++) com auxílio das bibliotecas *DtmLib* do SPA e também do *OpenDX*, para visualização em duas e três dimensões. Este visualizador é módulo integrante de um pacote de programas computacionais, aplicados ao gerenciamento de recursos hídricos subterrâneos, denominado SPA (Simulação de Processos em Aquíferos).

Os testes de avaliação do VPA apresentaram bons resultados quando aplicados na visualização do potencial de carga hidráulica do aquífero Bauru. A partir desses resultados pode-se afirmar que o visualizador desenvolvido apresenta um grande avanço como ferramenta de visualização para o gerenciamento de recursos hídricos subterrâneos.

AGRADECIMENTOS

Agradecimento ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) pela bolsa de estudo concedida, e ao Programa de Pós-Graduação do Departamento de Hidráulica e Saneamento da Escola de Engenharia de São Carlos da Universidade de São Paulo.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, W. P.: “*Programação Gráfica em Delphi 6*”, Editora Érica Ltda, São Paulo, 244 p., 2002.
- ANSI: “*Computer Graphics - Graphical Kernel System (GKS) Function Description*”, Technical Report, American National Standards Institute for Information Processing Systems, USA, 1985a.
- ANSI: “*Computer Graphics - Graphical Kernel System (PHIGS) Function Description*”, Technical Report, American National Standards Institute for Information Processing Systems, USA, 1985b.
- BODAMMER, FRANK et al.: “*SuSE Linux 8.1 – Administration Guide*”, SuSE Linux AG., Germany, 360 p., 2002.
- BODAMMER, FRANK et al.: “*SuSE Linux 8.1 – User Guide*”, SuSE Linux AG., Germany, 360 p., 2002.
- COMPANHIA DE TECNOLOGIA DE SANEAMENTO AMBIENTAL – CETESB: “*Uso das Águas Subterrâneas para o Abastecimento Público no Estado de São Paulo*”, São Paulo, 48 p., 1997.
- CONNER, D. B.; SNIBBE, S. S.; HERNDON, K. P.; ROBBINS, D. C.; ZELEZNIK, A. VAN DAM; “*Three - Dimensional Widgets, Proceedings of the 1992 Workshop on Interactive 3D Graphics*”, 183-188 p., 1992.
- DEPARTAMENTO DE ÁGUAS E ENERGIA ELÉTRICA – DAEE: “*Água Subterrânea: Uma Riqueza de São Paulo*”, Revista Água e Energia, 75-80 p., Abril 1999.
- DEPARTAMENTO DE ÁGUAS E ENERGIA ELÉTRICA – DAEE: “*Relatório de Situação dos Recursos Hídricos*”, Secretaria de Recursos Hídricos Saneamento e Obras, Governo do Estado de São Paulo, CD-ROM.

- EARNSHAW, R. A.; WATSON, D.: “*Animation and Scientific Visualization – Tools and Applications*”, Academic Press Harcourt Brace & Company Publishers, 285 p., 1993.
- ECKEL, BRUCE: “*Thinking in C++ - Second Edition*”, Upper Saddle River - New Jersey, Prentice Hall, 636 p., 2000, *e-Book* nos formatos de arquivo .html, .pdf e .doc, disponível no site: <http://mindview.net/Books>. Acesso em: 07 mar. 2003.
- ELVINS, T. T.: “*KIntroduction to Volume Visualization: Imaging Multidimensional Scientific Data*”, SIGGRAPH’94, Course Notes.
- ENGEL, B. A.; NAVULUR, K. C. S.: “*The Role of Geographical Information Systems in Groundwater Engineering*”, In: DELLEUR, J. W. (ed.). *The Handbook of Groundwater Engineering*. CRC Press, Cap. 21, 21/1 – 20/40 p.
- FEITOSA, F. A. C.; MANUEL FILHO, J.: “*Hidrogeologia: Conceitos e Aplicações*”, Recife, CPRM, 392 p., 2000.
- GOGU, R. C.; CARABIAN, G.; HALLET, V.: “*GIS – Based Hydrogeological Databases and Groundwater Modeling*”, *Hydrogeology Journal*, v. 9, 555-569 p., 2001.
- HEINY, LOREN: “*Programação gráfica para Windows com Borland C++*”, Tradução: Renata Barbosa da Silva Neto e Marcelo Pettengill, Rio de Janeiro, Editora LTC Livros Técnicos e Científicos, 490 p., 1993.
- HERNDON, K. P.; DAM, A. VAN; GLEICHER, M.: “*Workshop on the Challenges of 3D Interaction*”, *SIGHI Bulletin*, vol. 26, no. 4, 1-9 p., 1994.
- KRESIC, N.: “*Quantitative Solutions in Hydrogeology and Groundwater Modeling*”, Boca Raton, Florida, CRC Press, 1997.
- LEAP, D. I.: “*Geological occurrence of groundwater*”, In: DELLEUR, J. W. (ed.). *The Handbook of Groundwater Engineering*. CRC Press. Cap. 1. p. 1/1-1/52, 1999.
- MANZANO, JOSÉ AUGUSTO N. G.: “*SuSE Linux 6.4 – Guia Prático de Orientação*”, São Paulo, Editora Érica Ltda., 240 p., 2000.
- MINGHIM, ROSANE; OLIVEIRA, MARIA C. F.: “*Uma Introdução à Visualização Computacional*”, XVI Jornada de Atualização em Informática, Anais / XVII Congresso da Sociedade de Computação, Editado por Cláudia Maria Bauzer Medeiros – Brasília, UnB, 494 p., 1997.
- NAG’s IRIS EXPLORER: “*Visualization Technical Reports*”, Disponível em: <http://www.nag.co.uk/doc/TechRep/ietr.html>. Acesso em: 08 mar. 2003.
- OLIVEIRA, J. N.: “*Ferramental de gestão de águas subterrâneas para a cidade de São José do Rio Preto*”, São Paulo, São Carlos, 117 p., Tese (Doutorado) – Escola de Engenharia de São Carlos – Universidade de São Paulo, 2002.

- REBOUÇAS, A. C.: “Águas Subterrâneas. In: Águas Doces no Brasil: capital ecológico, uso e conservação”, Escrituras Editora, São Paulo, Cap. 4, 117-151 p., 1999.
- SEGAL, M. AND AKELEY, K.: “*The OpenGL Graphics Interface*”, Technical report, Silicon Graphics Inc., 1994. Disponível em:
http://www.opengl.org/developers/documentation/white_papers/oglGraphSys/opengl.html.
Acesso em: 07 mar. 2003.
- SEGAL, M. AND AKELEY, K.: “*The OpenGL Graphics Interface*”, Technical report, Silicon Graphics Inc., 1996. Disponível em:
http://www.opengl.org/developers/documentation/white_papers/opengl/index.html
Acesso em: 07 mar. 2003.
- SCHROEDER, WILL; MARTIN, KEN; LORENSEN, BILL: “*The Visualization Toolkit*”, New Jersey, Prentice Hall, 1996, 826 p.
- SCHROEDER, WILL; MARTIN, KEN; LORENSEN, BILL: “*The Visualization Toolkit - 2nd Edition – An Object-Oriented Approach to 3D Graphics*”, Prentice-Hall International (UK) Limited (London), 645 p., 1997.
- SONNINO, BRUNO: “*Kylix - Delphi para Linux - Guia Prático de Programação*”, Makron Books Ltda., São Paulo, 425 p., 2001.
- UPSON, C.; FAULHABER, T.; KAMMINS, D.; LAIDLAW, D.; SCHLEGERT, D.; VROOM, J.; GURWITZ, R.; DAM, A. VAN: Student Member IEEE Computer Society, and MARKUS GROSS, Member IEEE: “*The Application Visualization System: A Computational Environment for Scientific Visualization*” - IEEE Computer Graphics and Applications, vol. 09, no. 07, pp. 30-42, 1992.
- YOUNG, MARK; ARGIRO, DANIELLE; KUBICA, STEVEN: “*Cantata: The Visual Programming Environment for the Khoros System*”, Khoral Research, Inc., USA, 9 p., 2000. Disponível em: <http://www.khoral.com/khoros/cantata/>. Acesso em: 07 mar. 2003.
- WENDLAND, E. C.: Desenvolvimento do Pacote Computacional - SPA (Simulação de Processos em Aquíferos), Projeto de Pesquisa em Desenvolvimento, Processo Fapesp 02/09696-3.

MANUAIS DO OPENDX

IBM Visualization Data Explorer – “*User’s Guide*” - **SC38-0496**

Disponível em: <http://www.opendx.org/support.html> - Acesso em: 07 mar. 2003.

IBM Visualization Data Explorer – “*User’s Reference*” - **SC38-0486**

Disponível em: <http://www.opendx.org/support.html> - Acesso em: 07 mar. 2003.

IBM Visualization Data Explorer – “*Programmer’s Reference*” - **SC38-0497**

Disponível em: <http://www.opendx.org/support.html> - Acesso em: 07 mar. 2003.