

Visualização de Informação com recursos colaborativos e interativos utilizando o padrão X3D

Alisson Fernando Coelho do Carmo, Milton Hirokazu Shimabukuro
DMC - Departamento de Matemática e Computação
FCT/UNESP – Faculdade de Ciência e Tecnologia
Presidente Prudente, São Paulo, Brasil
alisondocarmo@gmail.com, miltonhs@fct.unesp.br

Resumo—O processo de exploração e análise de dados com técnicas de Visualização de Informação requer recursos de interação e colaboração para ser mais efetivo e eficiente, que juntamente com a representação gráfica tridimensional e o uso de padrões abertos agregam valor a esse processo. Este trabalho apresenta uma solução para a sincronização de visualizações distribuídas construídas e manipuladas dinamicamente, que emprega o padrão X3D, em conjunto com a biblioteca de classes Xj3D, para a visualização 3D interativa de dados. Para resolver a questão da ausência de recurso nativo, foi desenvolvido um *middleware* para gerenciar os aspectos relacionados à manipulação, interação e colaboração na cena X3D. O principal resultado alcançado nesta investigação é a verificação da viabilidade do mapeamento visual para um grafo de cena com a inserção de recursos de colaboração em ambiente de visualização distribuída utilizando o padrão X3D.

Palavras Chaves-Visualização de Informação; Colaboração; Interação; X3D;

Abstract—The process of data exploration and analysis applying Information Visualization techniques requires interactive and collaborative functions in order to be more effective and efficient, and with the three-dimensional graphical representation and the use of open standards add value to this process. This paper presents a solution to the synchronization of distributed views which are dynamically constructed and manipulated that use the X3D standard, together with the classes library Xj3D, for 3D data interactive visualization. Supplying a solution to the absence of native resource, it was developed a middleware to manage the aspects of handling, interaction and collaboration in the X3D scene. The main result achieved in this study is the verification of the feasibility of the visual mapping for the scene graph with the insertion of collaboration features in the distributed visualization environment using the X3D standard.

Keywords-Information Visualization; Collaboration; Interaction; X3D;

I. INTRODUÇÃO

A interação do usuário em ambientes online, alimenta bases de dados com diferentes tipos de informações. Estes dados brutos, armazenados geralmente em SGBD (Sistema Gerenciador de Banco de Dados), trazem consigo outras informações implícitas que podem ser obtidas a partir de processamentos e análises. Por outro lado, a grande oferta de dados, aumenta proporcionalmente a dificuldade presente na análise direta destes dados. Neste sentido, recursos computacionais para extração de conhecimento (KDD - *Knowledge Discovery in Databases*) se tornam ferramentas essenciais para viabilizar

a exploração de grandes volumes de dados. Estes recursos computacionais podem ainda serem potencializados com a utilização de recursos gráficos representativos e interativos, presentes na Visualização de Informação.

Existem diversas técnicas, encontradas em diferentes linhas de pesquisas correlacionadas que buscam se beneficiar da capacidade de cognição visual humana para enriquecer o processo de análise de dados. Dentre estas, existem artifícios da Mineração Visual de Dados (VDM - *Visual Data Mining*), *Visual Analytics* (VA), Visualização de Informação (VI), *Business Intelligence* (BI), entre outros. Nestas técnicas, prevalecem a abstração, sumarização e representação dos dados em elementos gráficos que facilitem a interpretação humana, extração de novas informações e descoberta de conhecimento. Nesta linha, Keim *et al* [1] afirmam que resultados mais eficazes podem ser alcançados através da combinação das capacidades do homem e da máquina, onde os resultados obtidos na análise da base de dados podem ser enriquecidos através de técnicas de visualização.

Heer e Agrawala [2] salientam que a tarefa de análise de informações não é individual e a utilização de recursos de colaboração permite otimizar e enriquecer este processo. Desta forma, a intervenção e interação de agentes humanos durante a análise, por meio de ambientes colaborativos, pode beneficiar a formulação de hipóteses e tarefas envolvidas durante o processo analítico, uma vez que opiniões a partir de diferentes visões contribuem para a composição do raciocínio.

Segundo Polys [3], a manipulação e exploração da visualização, a partir da interatividade com o usuário, aumenta ainda mais o poder de informar. Então, para enriquecer a capacidade de representação e beneficiar a exploração, a interação entre os usuários pode ser expandida para a navegação entre os próprios elementos visuais, imersos em um ambiente tridimensional.

Existem tecnologias, principalmente voltadas para Realidade Virtual (RV), que oferecem soluções para a construção, exploração e navegação em ambientes tridimensionais, contribuindo como um aspecto facilitador para a modelagem dos objetos gráficos, além de permitir a definição de comportamentos e interatividade. Dentre estas tecnologias, existe o padrão, internacionalmente reconhecido e aprovado pela ISO (*International Organization for Standardization*), X3D (*eXtensible 3D*). Este padrão representa os elementos em um grafo de

cena (*Scene Graph*) e pode ser manipulado dinamicamente por linguagens de programação que implementam os recursos da SAI (*Scene Access Interface*)¹, como a biblioteca Xj3D para linguagem Java.

A utilização do padrão X3D atende a necessidade da representação e exploração do ambiente tridimensional que, aliado à SAI, oferece recursos para a manipulação dinâmica da cena. No entanto, ambos - padrão X3D e biblioteca Xj3D - não oferecem nenhum recurso nativo voltado à sincronização de cenas em diferentes visualizadores, requisito necessário para a aplicação de recursos de colaboração síncrona. Já o registro (exportação) da cena gerada dinamicamente, destinado ao processo de colaboração assíncrona, pode ser facilitado com a utilização do formato de representação bem definido oferecido pelo padrão X3D, porém a SAI não oferece recurso para exportação desta cena gerada em tempo de execução - situação encontrada no processo de análise e Visualização de Informação.

Neste contexto, este trabalho tem o objetivo de apresentar uma abordagem para a representação de informações através do mapeamento de dados para o grafo de cena do padrão X3D, valendo-se de recursos de colaboração síncrona e assíncrona. Para isto, considera o desenvolvimento de um *middleware* para solucionar o problema imposto pela falta de recurso nativo, no que diz respeito à exportação e sincronização do grafo de cena X3D construído dinamicamente com a biblioteca de classes Xj3D, bem como centralizar - em uma camada intermediária - os recursos e métodos para a representação gráfica de informações para compor o processo de análise e exploração visual dos dados com recursos colaborativos e interativos.

O resultado da utilização do *middleware* demonstra uma representação visual, em forma de árvore, dos usuários de um Ambiente Virtual de Aprendizagem (AVA), diferenciados quanto ao sexo e idade. A visualização gerada pelo *middleware* pode ser sincronizada através da rede em instâncias alocadas em máquinas distintas, além de permitir a exportação em arquivo da visualização gerada. Espera-se que com a utilização do *middleware*, seja possível suprir os requisitos exigidos para a colaboração síncrona e assíncrona no contexto da Visualização de Informação, bem como atender às implementações de métodos de visualização.

As demais Seções descrevem os principais aspectos envolvidos no desenvolvimento do trabalho. Na Seção II são apresentados alguns trabalhos relacionados que abordam outras estratégias de sincronização e cenários de utilização do padrão X3D. A Seção III discute motivos que levaram a construção do *middleware*, bem como apresenta a estrutura proposta e seus principais componentes. Os resultados dos testes realizados com a visualização de estruturas presentes em um banco de dados de um AVA, sobretudo com a representação visual por objetos do grafo de cena, bem como sua exportação

e sincronização, são apresentados na Seção IV para então, subsidiar as conclusões apresentadas na Seção V.

II. TRABALHOS RELACIONADOS

Vucinic *et al* [4] apresentam um *framework* que integra tecnologias como Xj3D, X3D e SOAP (*Simple Object Access Protocol*), focado na visualização e manipulação de conteúdos gráficos em ambiente distribuído, voltados para o suporte à tomada de decisão na gerência de catástrofes. Para isto, utilizam uma arquitetura cliente-servidor, onde existe um *Graphical Middleware* (GM) no servidor, que contém operações básicas de manipulação da base de componentes X3D, armazenados e controlados por um SGBD, além de tratar o gerenciamento da cena 3D - geração, composição e histórico de eventos da cena 3D. Por fim, concluem positivamente que a estruturação da arquitetura aliada com a integração destas diferentes tecnologias, específicas para cada fim, oferecem um ambiente flexível e fracamente acoplado, onde cada componente pode sofrer alterações sem prejudicar as demais funcionalidades.

Para a criação de ambientes multiusuários com VRML (*Virtual Reality Modeling Language*)- antecessor do X3D, Araki e Inc. [5] propõem a arquitetura VSPLUS, baseando-se na utilização de NetNodes - Nós intermediários, localizados na rota de propagação de eventos da cena. Nesta arquitetura, os eventos ocorridos no ambiente são capturados pelos NetNodes e propagados a todas as conexões, através da própria cascata de eventos gerada pelo VRML, a partir de rotas pré-definidas. Esta arquitetura depende de um servidor específico, nomeado de *Consistence Manager*, para receber os eventos de entrada e retransmitir para os respectivos NetNodes das outras instâncias conectadas. Ao receber o evento, o NetNode receptor origina a cascata de eventos.

Com o objetivo de evitar a definição explícita de características do protocolo de comunicação no próprio NetNode, Vieira *et al* [6] estendem a definição de NetNodes e inserem um novo conceito: *NetNodeManager*. Neste cenário, cada NetNode possui seu próprio *NetNodeManager*, contendo a implementação do protocolo e informações sobre as instâncias conectadas, removendo a necessidade de um servidor para a propagação dos eventos, pois a comunicação ocorre entre cada *NetNodeManager* e não mais entre NetNode e *Consistence Manager*.

Heer e Agrawla [2] discutem cada fase presente no ciclo de construção do conhecimento (*Sensemaking Cycle*), definido por Card, Mackinlay e Shneiderman [7], no processo de análise visual e apresentam vários aspectos a serem considerados para a construção de ambientes para a análise visual colaborativa assíncrona, como: compartilhamento de artefatos, exportação do conteúdo, técnicas de seleção/apontamentos, entre outros.

Existem diversos trabalhos ([8], [9], [10]) que se beneficiam dos recursos oferecidos pelo padrão X3D, dentre eles, Anslow *et al* [11] apresentam a VARED3D (*Visualization Architecture for Reuse*), uma arquitetura que aplica o padrão X3D para a visualização de rotas na execução de software. VARE3D

¹A SAI está definida na especificação do padrão X3D e compreende um conjunto de funcionalidades e definições abstratas que devem ser implementadas por bibliotecas que buscam ser capazes de manipular este padrão

utiliza estruturas hierárquicas e se beneficia dos recursos de exploração e navegação no ambiente tridimensional oferecido pelos *browsers*² X3D, bem como portabilidade e facilidade na implementação do X3D, sem depender da codificação de grandes *scripts*.

III. GERENCIAMENTO DO AMBIENTE DE VISUALIZAÇÃO 3D

No contexto de Visualização de Informação em ambientes colaborativos síncronos, existem três considerações básicas, que se fazem presentes no processo de criação e distribuição da cena:

- *Construção dinâmica do conteúdo gráfico*: Não existe, a priori, um modelo gráfico a ser considerado antes do processamento dos dados, pois é o resultado do processamento que origina o objeto gráfico;
- *Compartilhamento do conteúdo gráfico*: Após a geração do conteúdo gráfico, este deve ser distribuído a todos os usuários conectados;
- *Sincronização de eventos*: Uma vez que todos os usuários possuem o mesmo conteúdo gráfico, estes devem ser mantidos sincronizados.

Considerando estas etapas, a manipulação e construção dinâmica da cena X3D pode ser efetuada diretamente através da biblioteca Xj3D. No entanto, o compartilhamento da cena X3D construída a partir da biblioteca Xj3D não é uma tarefa imediata, pois, conforme apresentado por Carmo e Shimabukuro [12], o objeto *X3DScene* que contém a cena não pode ser transmitido diretamente pela rede por conter componentes não serializáveis.

Outra estratégia é a recuperação de todos os atributos da cena X3D para permitir a reconstrução do arquivo X3D, em formato XML (*eXtensible Markup Language*), e então enviá-lo. Porém após a criação dinâmica de um nó na *X3DScene*, alguns atributos não podem ser recuperados, inviabilizando a reconstrução do arquivo XML.

Diante deste cenário, e com os resultados obtidos por Carmo e Shimabukuro [12], com relação ao compartilhamento da cena X3D, é proposto a criação de um *middleware* para gerenciar o grafo de cena. Desta forma, todas as operações efetuadas na cena, são primeiramente conhecidas pelo *middleware*, permitindo tanto o compartilhamento da cena inicial com os demais clientes, como também a sincronização de alterações e eventos dinâmicos que ocorrem na cena.

A. O Middleware

Em razão do *middleware* estar presente em uma camada intermediária, entre a manipulação com a biblioteca Xj3D e o padrão X3D, permite que suas funcionalidades não sejam restritas apenas à sincronização da cena, mas também estendidas aos outros aspectos envolvidos, como a visualização e interação. Partindo deste princípio, a estrutura proposta para este componente pode ser observada na Figura 1.

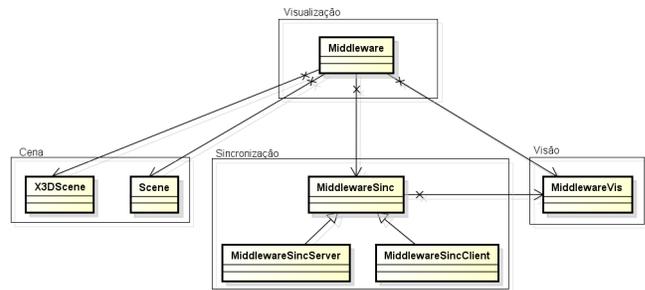


Figura 1. Estrutura do *middleware*

Como pode ser visto no diagrama da Figura 1, o *middleware* pode ser segmentado em quatro principais componentes: Componentes da Cena; Componente de Sincronização; Componente de Visão e Componente de Visualização. Estes componentes são descritos a seguir.

1) *Componentes da Cena*: Estes componentes foram criados de modo a definir uma cena abstrata, e armazenar os atributos dos nós que a compõem, para permitir que o *middleware* gerencie e transmita a cena X3D às outras instâncias sincronizadas. Deste modo, a representação dos nós e suas relações hierárquicas, origina um novo grafo de cena, com os dados necessários para reconstrução, transmissão e manipulação dos atributos da cena X3D. Na Figura 2 são apresentados os principais nós implementados, considerando as definições da própria especificação X3D.

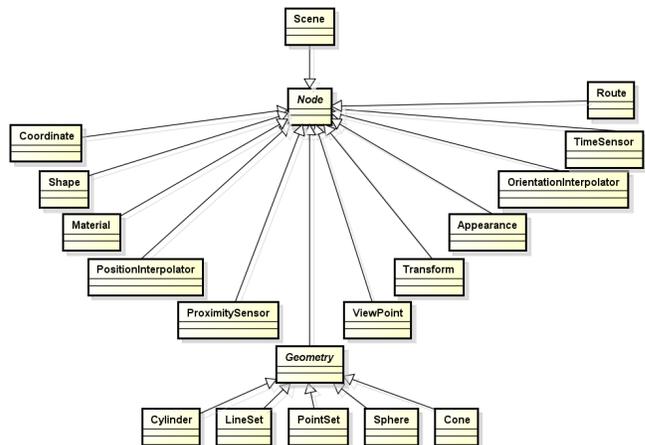


Figura 2. Nós abstratos definidos na cena genérica

Cada nó da nova cena é responsável por definir duas funcionalidades necessárias para a aplicação:

- *writeX3dScene*: método responsável por transformar o referido nó em um *X3DNode*³, adicionando-o à cena X3D. De forma geral, este método realiza a conversão entre o nó abstrato e o *X3DNode*;
- *toXml*: a partir dos atributos e características definidos no referido nó, este método é capaz de codificá-lo no formato

²Browsers X3D são aplicações responsáveis por interpretar o grafo de cena X3D, visualizar o ambiente X3D e oferecer recursos para a navegação na cena

³*X3DNode* é o objeto definido na biblioteca Xj3D que representa cada nó da cena

Índice	Identificador Descrição	Conteúdo
0	Nova Cena	Novo objeto contendo todos os nós da cena, construídos com os Componentes de Cena do <i>middleware</i>
1	Novo Nó	Novo objeto contendo um nó da cena definido nos Componentes de Cena do <i>middleware</i>
2	Remover Nó	Identificador do nó a ser removido
3	Novo Servidor	Novo endereço do servidor
4	Alterar Atributo	Nome do atributo e novo valor
5	Solicitar Comando	Mensagem do usuário para justificar a solicitação (para alteração do servidor)
6	Atualização de Câmera	Nova posição da exploração do ambiente, a partir do Componente Visão

Tabela I
PRINCIPAIS TIPOS DE MENSAGENS UTILIZADAS NO *middleware*

XML do padrão X3D, agindo recursivamente em todos os seus nós internos;

2) *Sincronização*: Este componente do *middleware* é responsável por conter todas as funcionalidades necessárias para a sincronização da cena. Como espera-se que cada usuário/analista possa ser capaz de investigar sua própria base de dados local, a comunicação entre as instâncias da aplicação será efetuada baseada em um servidor temporário de dados.

Um servidor temporário é a instância da aplicação na qual se encontra o analista principal, que comanda a sincronização, e a base de dados a ser explorada. O termo temporário é utilizado, pois o analista tem a opção de ceder o controle para outro analista, efetuando uma alteração do servidor. Neste sentido, o *middleware* representa uma arquitetura cliente-servidor.

A variedade de elementos enviados durante a sincronização, exige a definição de um segmento capaz de encapsular e identificar todas as mensagens que trafegam entre as instâncias das aplicações para garantir a correta comunicação. Basicamente, este segmento é composto de um identificador e um objeto associado, que é interpretado e convertido de acordo com o significado do identificador. Na Tabela I são apresentados os principais tipos de mensagens utilizadas.

Ao detectar um evento, que deve ser transmitido aos outros clientes, o servidor cria uma nova mensagem, define o identificador, encapsula o valor a ser transmitido e envia para todos os clientes. Dependendo do identificador da mensagem, pode ser necessário realizar mais de uma transmissão em várias iterações, como é o caso do envio inicial da cena. Na Figura 3 é exibido um diagrama de estados resumido que define o comportamento da sincronização no servidor.

Após a sincronização inicial, disparada a partir de cada cliente, é definido um ouvinte que aguarda o recebimento de novas mensagens com atualizações. Ao receber uma mensagem, o valor é interpretado de acordo com o identificador definido. Na Figura 4 é exibido um diagrama de estados que define o comportamento da sincronização no cliente.

3) *Visão*: Uma importante informação durante o processo de análise e exploração do conteúdo gráfico por parte do analista é a sua própria visão, isto é, a maneira com que

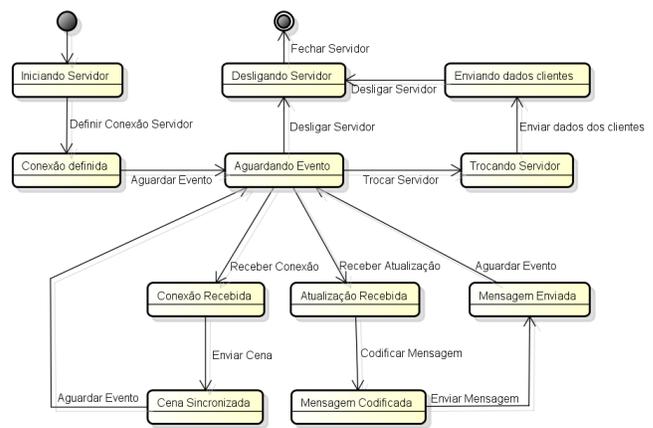


Figura 3. Diagrama de estados do servidor para a manipulação de mensagens

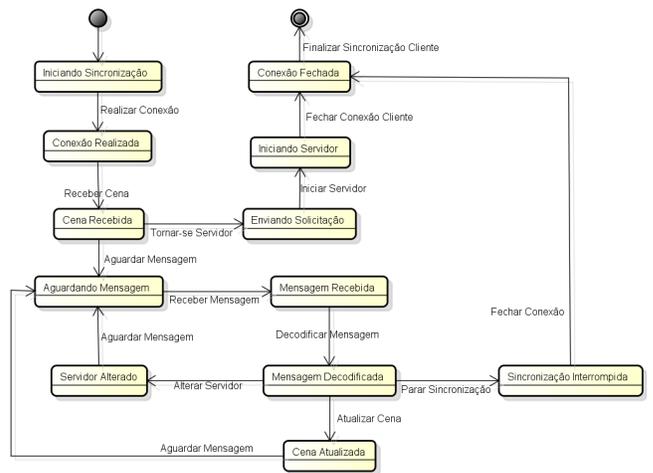


Figura 4. Diagrama de estados resumido do cliente para a manipulação de mensagens

ele está posicionado na cena, influenciando seu campo de visão na mesma. Partindo deste princípio, este componente é responsável pelo tratamento da exploração da cena. Enquanto o servidor realiza a exploração do ambiente (conteúdo gráfico), uma das funções deste componente é capturar os movimentos para serem transmitidos aos clientes e sincronizados.

A identificação do posicionamento na cena é baseada em um sensor X3D inserido no ambiente. Este sensor é atualizado de acordo com a proximidade no processo de exploração. A partir da proximidade, recupera-se a posição, orientação e a rotação em relação ao sensor e com estes dados é possível obter o posicionamento da visão do analista no ambiente. Uma vez que estes dados sejam conhecidos, é necessário alterar o ponto de visão para sincronizar a movimentação no ambiente.

Outra função presente neste componente é a possibilidade de gravação da exploração do ambiente, criando o efeito de animação dos objetos através da movimentação do ponto de visão. Para isto, são criados interpoladores de movimento, para cada atributo da visão: posição, orientação e rotação. Com as coordenadas capturadas, é preciso armazená-las e inseri-las em uma rota de eventos da cena X3D, transformando cada

um dos valores obtidos em entrada para o ponto de visão. Atenção especial é necessária para garantir a sincronia entre os movimentos e o tempo em que ocorrem.

Ao exportar a cena, a partir do *middleware*, contendo as rotas de exploração definidas, a cena manterá a capacidade de reproduzir a animação definida pela exploração do ambiente, a ser interpretada posteriormente em um *browser* X3D.

4) *Visualização*: Por fim, as técnicas de visualização podem ser investigadas para consumir os recursos disponíveis no ambiente tridimensional e colaborativo, e se beneficiar da interatividade e capacidade de representação oferecida pelo padrão X3D.

Buscando representar a estrutura relacional/hierárquica de um banco de dados, uma visualização em forma de árvore tridimensional foi implementada. Cada nó esférico da árvore representa um elemento que pode agregar informações visuais relacionadas ao seu diâmetro, cor, opacidade, posição, e outros atributos visuais. Além disso, cada aresta, que vincula os nós, pode representar algum outro atributo através da sua cor.

Esta representação, é útil para a exibição de camadas hierárquicas em uma ambiente, evidenciando certas características em atributos gráficos de cada elemento da árvore.

IV. RESULTADOS E DISCUSSÕES

Para avaliar a representação visual, os recursos de colaboração síncrona e assíncrona, e a interatividade do ambiente, o protótipo foi testado utilizando a base de dados de um AVA. Entre outras informações, este AVA contém características de diferentes cursos e dados relativos aos perfis dos usuários. O teste foi realizado considerando a disposição dos usuários de seis cursos registrados no AVA, apresentando informações sobre o sexo, representado pela cor do nó e aresta (azul: masculino, vermelho: feminino, branco: não informado) e a idade relacionada ao diâmetro do nó (quanto menor, mais novo). Cada nó do curso (verde), representa a quantidade de inscrições abertas como o diâmetro da esfera.

O resultado desta visualização pode ser observado na Figura 5, onde a visualização se encontra sincronizada em três máquinas distintas, tanto no sistema operacional Linux, como Windows.

Vale ressaltar que a utilização do padrão X3D para a representação permite que os aspectos relacionados a exibição/renderização da cena -como escala, proporção, nível de detalhe- sejam manipulados internamente pelo *browser* X3D utilizado, tornando-se transparente ao desenvolvedor.

A mesma visualização foi exportada a partir do *middleware*, tanto pelo cliente como pelo servidor, e executada corretamente em outro *browser* X3D, SwirlX3D (versão *trial*), conforme pode ser observado na Figura 6, demonstrando que, durante o processo de análise, as cenas geradas podem ser exportadas para visualização posterior de forma independente ao *middleware*. Tal artifício pode ser utilizado para registrar a cena gerada, para uma possível posterior colaboração assíncrona.

Vale ressaltar que por se tratar de um protótipo, a definição dos atributos a serem representados visualmente ocorre de

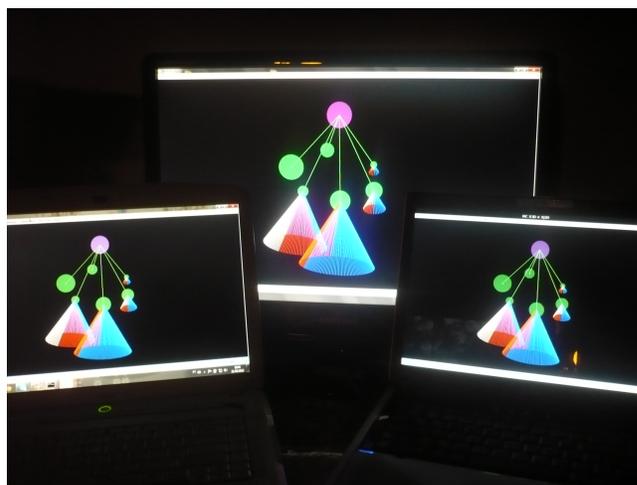


Figura 5. Visualização sincronizada em forma de árvore, da estrutura de usuários de um AVA, considerando o sexo e a idade dos usuários

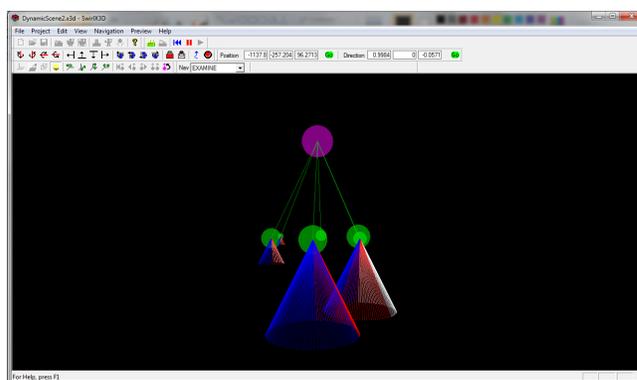


Figura 6. Visualização exportada a partir do *middleware*, para o formato XML do padrão X3D, e executada em outro *browser* X3D, SwirlX3D

forma estática. Em estágio posterior, o ideal é que o próprio analista tenha o poder de definir a relação entre os dados e os atributos visuais.

A arquitetura de sincronização de eventos definida por Araki e Inc. [5] tem grande utilidade em situações na qual todas as instâncias possuem a mesma cena predefinida, ou seja, o cenário de execução é previamente conhecido e todos possuem os mesmos elementos gráficos para a propagação dos eventos. No entanto, no contexto de Visualização de Informação, a cena contendo os elementos gráficos é conhecida somente após o processamento dos dados, e não exige apenas a sincronização dos eventos, mas também a sincronização da cena construída dinamicamente e outros objetos gráficos que podem ser adicionados em tempo de execução.

A proposta de Vieira *et al* [6] mantém o mesmo conceito anterior, que, por se tratar de uma definição abstrata baseada apenas no padrão de representação (X3D ou VRML), não pode ser aplicada diretamente em situações onde os eventos não podem ser definidos a partir de rotas de eventos do padrão VRML/X3D.

Por outro lado, Vucinic *et al* [4] definem uma arquitetura

baseada em um *middleware* gráfico onde a cena 3D é construída e armazenada no servidor, a partir de um conjunto de componentes X3D, e enviada para o cliente. No entanto, esta estratégia trata de cenas construídas individualmente pelos clientes ou de forma assíncrona (armazenar no banco de dados para posterior recuperação), onde os eventos da cena também são armazenados no banco de dados.

Já este trabalho introduz uma nova estratégia para a implementação de recursos de colaboração, tanto síncrona como assíncrona, no contexto de Visualização de Informação, onde a cena (objeto gráfico) é construída dinamicamente a partir de processamento/mineração dos dados. Em razão de utilizar o padrão X3D, beneficia-se também de todos os recursos de interatividade e exploração do ambiente tridimensional oferecido pelos *browsers* X3D.

V. CONCLUSÃO

Com os resultados obtidos, apresentados na Seção anterior, pôde-se observar a relevância e viabilidade da utilização do *middleware* proposto para suportar a manipulação da cena, agregando recursos inexistentes na própria biblioteca Xj3D. Estes resultados têm demonstrado a utilidade do *middleware* no contexto de Visualização de Informação, conforme pode ser observado com o protótipo de visualização em forma de árvore. O ambiente construído demonstrou ser capaz de atender recursos de interatividade e colaboração para enriquecer a representação visual de informações, onde técnicas de visualização podem ser investigadas para serem empregadas neste ambiente.

No decorrer do trabalho, foram priorizadas ferramentas e tecnologias livres e de código aberto. A adoção do padrão X3D tem se mostrado uma opção acertada, pois oferece um ambiente estável e atual, que condiz com a proposta do próprio Web3D Consortium de favorecer e incentivar a utilização de tecnologias que facilitem sua integração com ambientes na internet, como é o caso do XML. Isso confirma a conclusão de Anslow *et al* [13], que após análise considerando os aspectos positivos e negativos do padrão X3D voltado para a visualização de software, incentiva os desenvolvedores pela adoção do padrão.

Estudos estão sendo direcionados para a generalização da definição do *middleware*, evitando a necessidade de implementação de cada nó X3D que se pretenda utilizar, limitação atual do *middleware*. Para isto, está sendo considerado a utilização da funcionalidade *Reflection*⁴ da linguagem Java, para a otimização da estruturação do *middleware*. Desta forma, seria possível definir todos os nós X3D, seus atributos e relacionamentos em uma estrutura externa, como um arquivo XML, para a permitir a manipulação e construção em tempo de execução.

Espera-se que o *middleware* seja capaz de atender as exigências para a colaboração no contexto da Visualização de Informação, bem como atender a futuras implementações

de métodos de visualização que consomem seus recursos de interação e colaboração, síncrona e assíncrona. De modo geral, que este trabalho possa incentivar a utilização de ambientes tridimensionais ricos e interativos, sobretudo que ofereça a possibilidade de colaboração entre usuários para alcançar um resultado em comum, como o processo de análise de dados.

AGRADECIMENTOS

Os autores agradecem o Departamento de Matemática e Computação (DMC) da Faculdade de Ciências e Tecnologia/Unesp (FCT/UNESP)- Campus de Presidente Prudente - por ter propiciado o início do desenvolvimento desta investigação através do Programa de Formação Complementar; e a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo prosseguimento da pesquisa como projeto de Iniciação Científica.

REFERÊNCIAS

- [1] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, Jörn, and G. Melançon, "Information visualization," A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Visual Analytics: Definition, Process, and Challenges, pp. 154–175.
- [2] J. Heer and M. Agrawala, "Design considerations for collaborative visual analytics," in *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 171–178.
- [3] N. Polys, *Publishing Paradigms for X3D*. New York: Springer, 2005, pp. 153–180.
- [4] D. Vucinic, D. Deen, E. Oanta, Z. Batarilo, and C. Lacor, "Distributed 3d information visualization - towards integration of the dynamic 3d graphics and web services," in *GRAPP 2006: Proceedings of the First International Conference on Computer Graphics Theory and Applications, Setúbal, Portugal, February 25-28, 2006*, J. Braz, J. A. Jorge, M. Dias, and A. Marcos, Eds. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2006, pp. 251–258.
- [5] Y. Araki and S. M. E. J. Inc., "Vspplus: A high-level multi-user extension library for interactive vrml worlds," in *Proceedings of the third symposium on Virtual reality modeling language*, ser. VRML '98. New York, NY, USA: ACM, 1998, pp. 39–47. [Online]. Available: <http://doi.acm.org/10.1145/271897.274369>
- [6] F. Vieira, M. Degiovani, A. C. Colombini, J. Hattori, M. Luppe, C. V. Amadeu, A. F. Traina, and D. F. Pires, "Sincronização de ambientes virtuais x3d," in *Terceiras Jornadas de Engenharia de Electrónica e Telecomunicações e de Computadores*, 2005.
- [7] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in information visualization: using vision to think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [8] T. Hopf, G. A. M. Falkembach, and F. V. d. Araújo, "O uso da tecnologia x3d para o desenvolvimento de jogos educacionais," *RENOTE - Revista Novas Tecnologias na Educação*, vol. 5, no. 2, 2007.
- [9] D. Brutzman, "Computer graphics teaching support using x3d: extensible 3d graphics for web authors," in *ACM SIGGRAPH ASIA 2008 courses*, ser. SIGGRAPH Asia '08. New York, NY, USA: ACM, 2008, pp. 23:1–23:335. [Online]. Available: <http://doi.acm.org/10.1145/1508044.1508067>
- [10] A. Havele, "Augmented reality roadmap for x3d," 2011.
- [11] C. Anslow, S. Marshall, J. Noble, and R. Biddle, "X3d software visualisation," in *In Proc. of NZCSRSC*, 2007.
- [12] A. F. C. do Carmo and M. H. Shimabukuro, "Estudo sobre o uso do x3d e java para a análise visual 3d interativa e colaborativa de dados via internet," in *[Anais do] WRVA'2011 8 Workshop de Realidade Virtual e Aumentada*, Uberaba, MG, Brasil, 2011.
- [13] C. Anslow, K. Jackson, R. Khaled, A. Martin, M. Duignan, H. Smith, B. Chawner, J. Evermann, and R. Morrison, "Evaluating extensible 3d (x3d) graphics for use in software visualization," Master's thesis, School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, New Zealand, 2008.

⁴Reflection é um recurso da linguagem Java que permite a definição, acesso e manipulação dinâmica de classes em tempo de execução