

Uma Proposta de um Ambiente Extensível de Ensino à Distância Aplicado à Área de Engenharia de Software

Daniel Abella C. M. de Souza;
Rodrigo da Cruz Fujioka
Departamento de Informática
Universidade Federal da Paraíba
+55 (83) 3216-7200

{daniel.abella,rodrigo.fujioka}@
di.ufpb.br

Ryan R. de Azevedo; Fred
Freitas; Marcelo Siqueira
Centro de Informática
Universidade Federal de Pernambuco
+55 (81) 2126-8000

{rra2,fred,mjsca}@cin.ufpe.br

César Rocha de Vasconcelos
Coordenação de Informática Aplicada
Centro Federal de Educação
Tecnológica de Urutaí
+55 (64) 3413-2112

cesar@cefeturutai.edu.br

ABSTRACT

Due to the technologies evolution for development of virtual environments, even more realistic user interfaces have been created. In the last years, virtual environments have become an important element adopted for knowledge development in several areas. This paper introduces an extensible environment for distance learning applied to the Software Engineering area, however not limited only for this area. For development of this environment, this work proposes an infrastructure that allows the dynamic association of functionalities to the browser, so that the environment can be easily customized in real time.

Categories and Subject Descriptors

I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism – *Virtual Reality*.

General Terms

Management, Design, Human Factors.

Keywords

Ambientes Virtuais, Realidade Virtual.

1. INTRODUÇÃO

Atuando como um aliado ao ensino presencial, os ambientes de ensino à distância eliminam as fronteiras físicas para o aprendizado, de maneira que o desenvolvimento de habilidades seja realizado através da Internet.

Buscando a redução do impacto na adoção destes tipos de sistemas, ambientes virtuais podem ser aplicados. Nestes ambientes, a realidade pode ser simulada, de maneira a compor um ambiente interativo composto por objetos tridimensionais [1].

Nos últimos anos, ambientes virtuais tornaram-se uma das maneiras adotadas para a composição de ambientes de aprendizagem em diversas áreas do conhecimento [2]. Em decorrência disso, uma diversidade de ambientes de ensino é desenvolvida com características muito próximas da realidade.

Em ambientes virtuais aplicados ao ensino, a capacidade de gerir as funcionalidades associadas aos recursos educacionais é dificilmente considerada. Em virtude disto, podemos ter ambientes com recursos obsoletos e/ou desatualizados, desconsiderando mudanças nos aspectos educacionais.

A proposta deste artigo é a de apresentar características de um *browser* para a representação de ambientes tridimensionais, que permite a associação dinâmica de funcionalidades, devido à infraestrutura desenvolvida, possibilitando assim, a definição de um ambiente extensível de ensino à distância.

O artigo está organizado da seguinte maneira. Na seção 2, é apresentada a arquitetura do *browser* desenvolvido e na seção 3, são apresentados os aspectos relacionados ao funcionamento do sistema desenvolvido. Na seção 4, é realizado um estudo de caso do *browser* desenvolvido, enquanto que na seção 5 são apresentadas as conclusões e trabalhos futuros.

2. ARQUITETURA DO BROWSER

A arquitetura do *browser* proposto dispõe de três componentes conforme apresentado na Figura 1. Nesta aplicação, estes componentes estão organizados em dois módulos, que são os módulos *Task* e *Principal*.

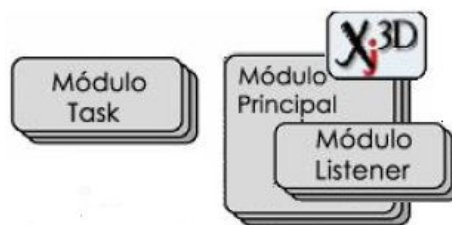


Figura 1. Arquitetura do browser proposto.

A implementação de funcionalidades relacionadas à manipulação de todos os ambientes apresentados está agrupada no módulo *Principal*. Ainda neste módulo, temos um mecanismo de notificação, que nos permite informar qualquer outra aplicação com dados relativos à navegação do usuário com o ambiente em utilização pelo usuário. Tais dados podem auxiliar durante a avaliação do usuário nestes ambientes.

Com o objetivo de facilitar o desenvolvimento das funcionalidades associadas à manipulação dos ambientes do módulo *Principal*, o *toolkit* Xj3d [3] foi utilizado. Este *toolkit* permite a importação de ambientes escritos nas linguagens VRML [4] e X3D [5]. A linguagem utilizada pelo *browser* proposto aqui é a X3D, por dispor de recursos avançados de áudio, vídeo, navegação e interação com o usuário.

No módulo *Task* estão relacionados todos os recursos que permitem a associação dinâmica de funcionalidades. Este módulo é composto pelo componente chamado *Task Manager*. De maneira não tão abrangente, este componente, após a leitura de um arquivo de configuração chamado descritor de implantação (DI) nomeado como *browser.xml*, que é implementado na linguagem XML, efetuará a associação das funcionalidades ao *browser* com base em regras previamente definidas no DI. Este processo de associação é apresentado na Figura 2.

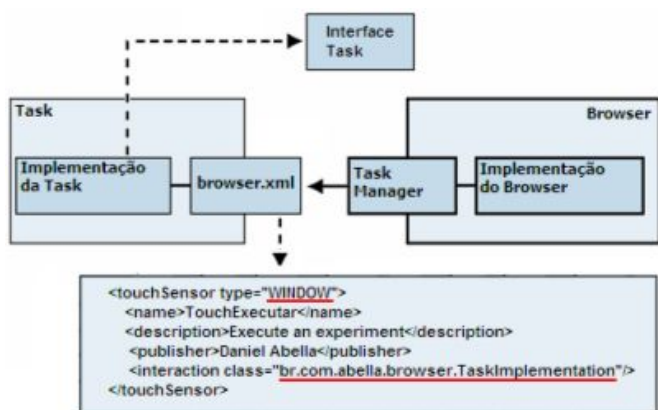


Figura 2. Representação do processo de associação de uma *task* ao *browser*.

A maneira com que estas funcionalidades são adicionadas dinamicamente ao *browser* é apresentada na próxima seção.

3. ANÁLISE DO BROWSER

Com o objetivo de associar dinamicamente as funcionalidades ao *browser*, o DI, geralmente nomeado como *browser.xml*, relaciona o uso de *touchSensors* para descrever o momento em que uma determinada funcionalidade deve ser associada (nomeada como *task*, nesta aplicação).

O *touchSensor* é uma forma de interação do usuário com os ambientes virtuais, onde temos a disposição de sensores que acionam um evento com base nas ações deste usuário. As ações detectadas podem variar desde um clique do mouse, até movimentos menos perceptíveis, como a aproximação de um objeto.

Entretanto, não são todos os *touchSensors* que necessitam ter funcionalidade associadas, de maneira que se torna necessário diferenciar os *touchSensors* que possuem funcionalidade associada e quais não possuem.

Com o intuito de satisfazer esta necessidade, é realizada a categorização de *touchSensors*, podendo ter um dos três tipos definidos nesta aplicação. De acordo com a sua aplicabilidade, o *touchSensor* pode ser classificado como:

- **NOTIFY**: notifica todos as aplicações associadas ao *browser* com dados referentes a interação do usuário com o *touchSensor* assinalado com este tipo;

- **NONE**: nenhuma ação é desempenhada. Este é o padrão para todos os *touchSensors* que não foram associados no descritor de implantação;
- **WINDOW**: sinaliza ao *Task Manager* que a ação desempenhada com o *touchSensor* associado com este tipo, desencadeará a associação de uma nova *task*, permitindo assim a inclusão em tempo de execução de um recurso educacional (*task*) mediante interação com um *touchSensor* assinalado com este tipo.

Conforme dito anteriormente, o DI deste *browser* é descrito através da linguagem XML, cuja estrutura deste documento está descrita através de um *Schema*. É apresentada, na Figura 3, a representação gráfica da estrutura deste documento, enquanto que, na Figura 4, há um exemplo válido de um descritor de implantação para o *browser*.

De maneira a complementar a representação gráfica da estrutura do descritor de implantação apresentada, as seguintes informações podem ser relacionadas:

- A *tag browser* descreve todos os *touchSensors* e seus tipos, através da *tag touchSensor* e do atributo *type*, respectivamente.
- A *tag touchSensor* possui as *tags name*, *description* e *task*, que descrevem respectivamente o nome do *touchSensor*, descrição do mesmo e o nome qualificado da classe que contém a funcionalidade (*task*) que será associada dinamicamente.
- A *tag task* é opcional, devendo apenas ser associada às *tags touchSensor* que possuam o atributo *type* configurado com **WINDOW**.
- O atributo *class* da *tag task* conterá o nome qualificado da classe que provê a implementação da interface *Task*. Nesta classe, temos todas as funcionalidades da tarefa que será associada ao *browser*.

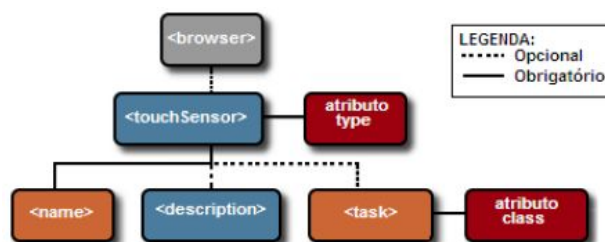


Figura 3. Representação Gráfica da Estrutura do DI.

```

<browser xmlns="http://www.abella.com.br/schema/browser"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://abella.com.br/schema/browser
http://abella.com.br/schema/browser/browser.xsd"/>
  <touchSensor type="WINDOW">
    <name>TouchExecutar</name>
    <description>Execute an experiment</description>
    <task class="com.abella.browser.TaskImplementation"/>
  </touchSensor>
  <touchSensor type="NOTIFY">
    <name>TestDrivenDevelopmentExperiment</name>
    <description>TDD Learning Resources</description>
  </touchSensor>
</browser>

```

Figura 4. Exemplo válido de um DI para o browser desenvolvido.

O resultado da associação de uma *task* é apresentado na Figura 5. Neste exemplo, uma janela que contém informações em formato texto é apresentada em complementação ao *browser*. Entretanto, vale lembrar que *tasks* complexas podem ser desenvolvidas, conforme veremos na próxima seção.

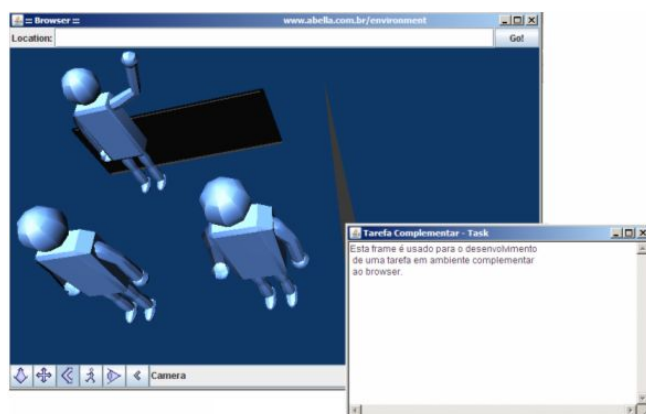


Figura 5. Demonstração da execução de uma *task*.

4. ESTUDO DE CASO: ANÁLISE DO *BROWSER* DESENVOLVIDO NO ENSINO DA METODOLOGIA *EXTREME PROGRAMMING*

Na área de Engenharia de Software, podem ser desenvolvidas *tasks* que possibilitem, por exemplo, a execução de programas simples desenvolvidos com a linguagem Java, execução de *scripts*

e até definição de testes. Além disso, *tasks* podem ainda ser criadas para a customização do ambiente proposto visando à aplicabilidade em outras áreas do conhecimento.

É apresentado, nesta seção, um estudo do caso, onde o *browser* é utilizado para o ensino de alguns tópicos da metodologia de desenvolvimento *Extreme Programming* (XP), que dispõe de uma série de práticas a serem estudadas.

Buscando o entendimento de cada uma destas práticas, um corredor (*hall* de entrada) com diversas portas é apresentado ao usuário quando o *browser* é inicializado, conforme ilustrado na Figura 6. Em cada uma destas portas, é garantido o acesso a uma sala em que umas das práticas da metodologia XP são apresentadas.

Cada uma destas salas é composta por objetos e atores. Os atores realizam atividades com o intuito de fornecer ao usuário, o entendimento de alguma prática da metodologia XP, enquanto os objetos representam algum conceito sobre a metodologia.

É apresentada na Figura 7, uma das práticas da metodologia XP, chamada *Stand up meeting*. Esta prática se refere a uma breve reunião diária, em pé, entre a equipe de desenvolvimento com o objetivo de compartilhar informações sobre o projeto. Quando o usuário entra na sala em que esta prática é apresentada, os atores deste cenário se movimentam com o objetivo de transmitir a sensação de que uma reunião está em andamento. As cadeiras que são apresentadas estão propositalmente desocupadas para que o usuário compreenda que as reuniões da prática *Stand up meeting* são realizadas de fato em pé.

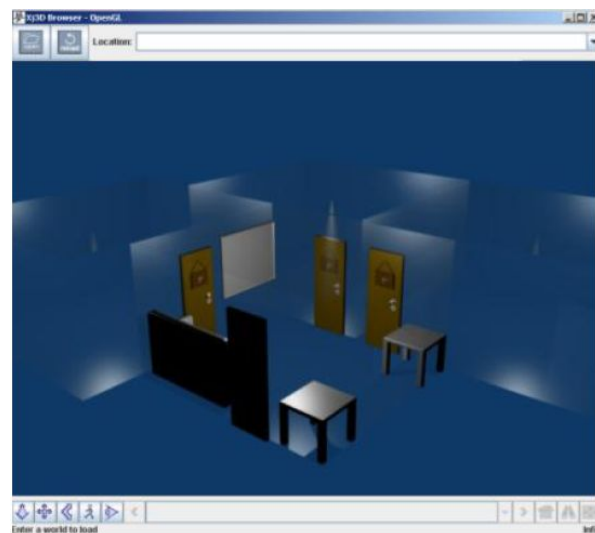


Figura 6. Parte inicial do ambiente: *hall* de entrada.

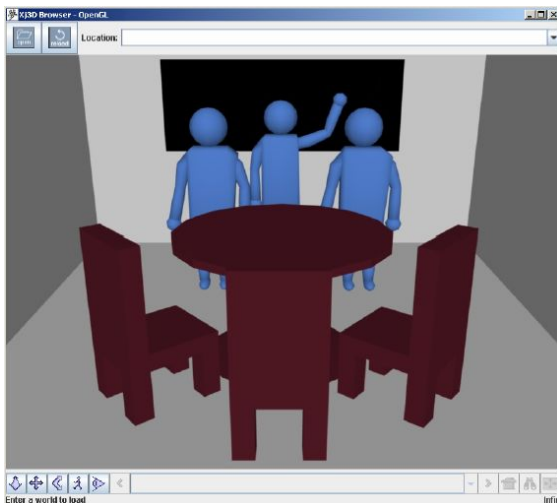


Figura 7. Demonstração da prática *Stand up meeting*.

Em outra sala também acessível através do corredor, temos a prática *Test-Driven Development* (TDD), conforme ilustrado na Figura 8. Nesta sala, temos uma sala com computadores, onde uma funcionalidade (*Task*) é executada caso um destes computadores seja clicado. Note que, nesta *task* podemos desenvolver atividades complementares ao que fazemos no *browser*. Neste caso, podemos digitar um trecho de código que se refere a um teste da prática TDD.

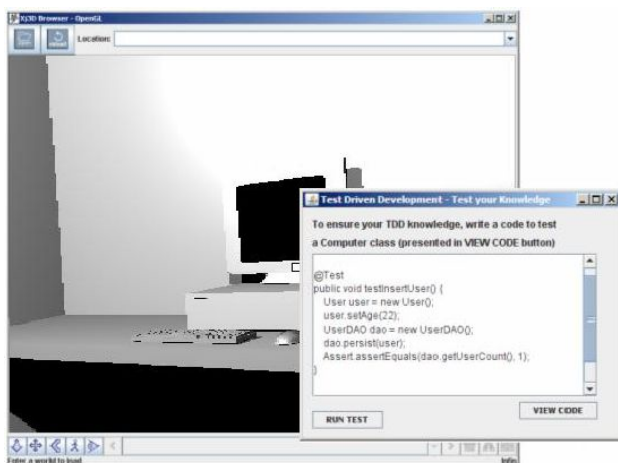


Figura 8. Demonstração da prática *TDD* com uma *task*.

5. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um *browser*, que dispõe de uma infraestrutura que permite a associação dinâmica de funcionalidades aplicadas à área de Engenharia de Software, entretanto não limitado apenas a esta área.

O sistema proposto possui o objetivo principal de permitir que o processo de definição e associação de funcionalidades ao *browser* seja realizado em caráter evolutivo, tornando este sistema um ambiente extensível de EAD.

Como trabalho futuro, é proposto o desenvolvimento de um repositório público para o armazenamento de funcionalidades desenvolvidas, permitindo a reciclagem e o compartilhamento de soluções encontradas para o desenvolvimento de funcionalidades em diversas áreas do conhecimento.

6. REFERÊNCIAS

- [1] Kirner, C., Siscoutto, R., "Fundamentos de Realidade Virtual e Aumentada". Proceeding of IX Symposium of Virtual Reality, Petrópolis – RJ, 2007.
- [2] Aquino, M. S.; Souza, F. F.; Frery, A. C.; Souza, D. A. C. M.; Fujioka, R. C. (2007), "Supporting Adaptive Virtual Environments with Intelligent Agents". Proceeding of 7th International Conference on Intelligent Systems Design and Applications, Rio de Janeiro - RJ, 2007.
- [3] The Web3D Consortium Specifications: Extensible 3D (X3D), Virtual Reality Modeling Language (VRML-ISO/IEC 14772:1997)
http://www.web3d.org/fs_specifications.htm
- [4] G. Bell, R. Carey, and C. Marrin. The Virtual Reality Modeling Language Especificação da Versão 2.0.
<http://vag.vrml.org/VRML2.0/FINAL>, 1996.
- [5] Johannes Behr , Patrick Dähne , Marcus Roth, Utilizing X3D for immersive environments, Proceeding of Ninth International Conference on 3D Web technology, April, 2004, Monterey, California. DOI =
<http://doi.acm.org/10.1145/985040.985051>