

# Gerência de Direitos de Acesso de Usuário em Ambientes Virtuais Colaborativos

José Valentim dos S. Filho<sup>1,2</sup>, Aloysio de C. P. Pedroza<sup>1</sup> e Jean-Pierre Courtiat<sup>21</sup>

<sup>1</sup>GTA - Grupo de Teleinformática e Automação - Universidade Federal do Rio de Janeiro (UFRJ)  
Centro de Tecnologia, H301, Ilha do Fundão, Rio de Janeiro, Brasil

<sup>2</sup>LAAS/CNRS - Laboratoire d'Analyse et d'Architecture des Systèmes  
UPS - Université Paul Sabatier, Toulouse, França.

{valentim,courtiat}@laas.fr,{valentim,aloyisio}@gta.ufrj.br

## RESUMO

*Ambientes Virtuais Colaborativos (AVCs) representam uma importante categoria de sistemas CSCW (Computer-Supported Cooperative Work), que utilizam espaços 3D compartilhados a fim de suportar atividades de colaboração. No que diz respeito às políticas de controle de acessos em AVCs, elas são freqüentemente baseadas somente no conceito de role. Neste artigo, propomos um AVC com um controle de acesso adaptável. O nível de acesso que um usuário tem sobre um objeto depende do seu estado e do estado do objeto. Além disso, definimos uma API (Application Program Interface) que torna possível a integração deste AVC com outras aplicações colaborativas.*

## Palavras-chave

*AVC, Awareness, Gestão de Direitos, Integração*

## ABSTRACT

*CVEs (Collaborative Virtual Environments) represent an important category of CSCW (Computer-Supported Cooperative Work) systems that generally make use of a 3D shared space in order to provide collaboration facilities. With respect to policies access control in the CVEs, they are often only role-based. In this paper, we propose a CVE with an adaptive access control. The access level of a user on each object depends of the user and object's state. Moreover, we have defined an API (Application Program Interface) to makes possible the CVE integration with other collaborative applications.*

## Keywords

*CVE, Awareness, Right's Management, Integration*

## 1 INTRODUÇÃO

Um dos problemas mais críticos no emprego de aplicações cooperativas é fornecer ao usuário uma real noção da presença de todos os usuários com os quais ele coopera. A realidade virtual, permitindo representar o mundo real através de metáforas, torna esta noção de presença mais natural, com as possibilidades de interação bem próximas daquelas do mundo real. Neste contexto, os Ambientes Virtuais Colaborativos (AVCs) desempenham um papel bastante importante, pois eles permitem à usuários separados geograficamente se comunicar e interagir em um ambiente 3D compartilhado, freqüentemente chamado de cena virtual. Neste artigo, propomos um AVC que fornece um sistema de controle dos direitos dos usuários baseado no estado do usuário e no estado do objeto. Além disso, definimos uma API que visa expandir as possibilidades de colaboração entre os usuários para além da cena virtual. Este artigo é estruturado da seguinte maneira: Seção 2 apresenta um estado da arte

---

<sup>1</sup>Agradecimentos à CAPES e ao LAAS/CNRS-França pelo financiamento.

sobre ambientes virtuais colaborativos. Na Seção 3, apresentamos uma visão geral do contexto no qual este trabalho está sendo desenvolvido; na Seção 4 apresentamos nosso AVC de maneira mais formal, e tratamos também dos aspectos ligados à configuração de uma sessão virtual colaborativa; na Seção 5 descrevemos a arquitetura e aspectos de implementação do AVC. Na última seção, apresentamos algumas perspectivas sobre trabalhos futuros.

## 2 Estado da Arte

Ambientes virtuais colaborativos têm sido usados em várias aplicações, tais como: *e-learning*, treinamento, simulações de guerra, etc. DIS [1], HLA [2] and NPSNET [3] são exemplos de sistemas voltados para simulações de guerra. Dos sistemas que propuseram mecanismos para implementar ambientes virtuais multiusuários baseados em VRML, destaca-se Living Works [4]. Nos sistemas baseados em agentes, como em InViwo [5], Agile [6] and PAR [7], um conjunto de agentes autônomos percebe o ambiente e reage segundo regras predefinidas. INVITE project<sup>2</sup>, DIVE [8], VREng<sup>3</sup> e VNet<sup>4</sup> são exemplos de AVCs onde os usuários podem compartilhar um ambiente 3D e interagir com outros usuários e objetos. Nossa proposta de AVCs se enquadra nesta categoria.

## 3 Contexto Geral

Esta seção descreve o contexto geral em que o trabalho apresentado neste artigo está sendo desenvolvido. Na Figura 1, podemos ver um possível cenário de integração onde três aplicações colaborativas - um AVC, COLAB [9] and PLATINE [10] - são integradas por LEICA. LEICA é um sistema de integração baseado em regras onde as aplicações interagem através da notificação dos eventos ao seu sistema de eventos.

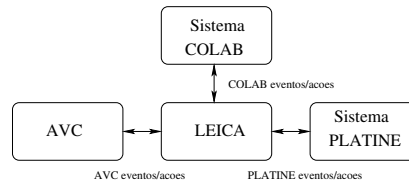


Figura 1: Um Possível Cenário de Integração

O AVC proposto neste artigo objetiva essencialmente ser uma plataforma virtual para colaboração, podendo ser utilizadas em aplicações como: *e-learning*, treinamento e entretenimento.

## 4 Ambiente Virtual Colaborativo - Uma Visão Formal

O AVC deve suportar diferentes sessões de acordo com o propósito da simulação. Formalmente, uma sessão virtual colaborativa é descrita como uma 4-tupla:  $(VScene, Roles_l, Users_l, Attr_l)$ .  $VScene$  é a cena virtual;  $Roles_l$  é um conjunto finito de roles gerais;  $Users_l$  é o conjunto finito de usuários conectados em uma mesma sessão virtual colaborativa; e  $Attr_l$  é a lista de atributos caracterizando a sessão virtual colaborativa.

### 4.1 Cena Virtual

A cena virtual é definida como uma 4-tupla:  $(VSpace, Avatares_l, Objects_l, API)$ .  $VSpace$  é uma representação 3D de um ambiente virtual. Os avatares podem executar ações de navegação e ações

<sup>2</sup><http://invite.fh-joanneum.at/>

<sup>3</sup><http://vreng.enst.fr/html/index.html>

<sup>4</sup><http://www.csclub.uwaterloo.ca/u/sfwhite/vnet>

sobre objetos. As ações adequadas, consideram as funcionalidades dos objetos, reforçam para um usuário a consciência de quais ações estão sendo executadas pelos outros usuários com os quais ele compartilha a cena virtual. Tornando mais claras as atividades de colaboração dos usuários, através de ações dos avatares. Para cada objeto podem ser associadas regras de acessibilidade, levando em consideração o seu estado e o estado do usuário. A definição de uma API permite uma integração aberta do nosso AVC com outras ferramentas colaborativas. Por exemplo, a seleção de um outro avatar por parte do usuário, pode ser interpretado pelo sistema encarregado da integração como sendo um desejo do usuário de realizar uma sessão de áudio conferência; ou um objeto virtual representando um telefone, pode representar o *link* que inicia uma aplicação *VoIP*, como o sistema Skype<sup>5</sup>. Assim como em VHA [11], consideramos um usuário em termos de máquina à estados finitos (MEFs). Em VHA, os autores consideram um determinado estado do usuário como uma condição a ser atingida para que o humanoíde possa realizar um comportamento previamente programado. No nosso sistema, o estado do usuário é um parâmetro utilizado na avaliação dos direitos de acesso sobre um objeto que tal usuário tem.

## 5 Arquitetura do Sistema e Aspectos de Implementação

A Figura 2 descreve a arquitetura do sistema. O módulo *Gerenciador de Eventos* distribue todos os eventos da cena virtual entre os módulos *Gerenciador de Avatares* e *Gerenciador de Objetos*. O módulo *Gerenciador de Avatares* é responsável por atualizar o estado do usuário e o módulo *Gerenciador de Objetos* é responsável por atualizar o estado do objeto. O módulo *Gerenciador de Direitos* avalia quais são as ações estão disponíveis para os objetos. O módulo *Gerenciador de Avatares* é responsável por parametrizar o script definindo a ação. Depois disso, o *Action-Engine* traduz o script que define a ação, já parametrizada, em uma ação executável. As interações entre usuários e objetos seguem um modelo seleção-condição/seleção-ação e podem ser naturalmente descritos por processos CCS de Milner [12].

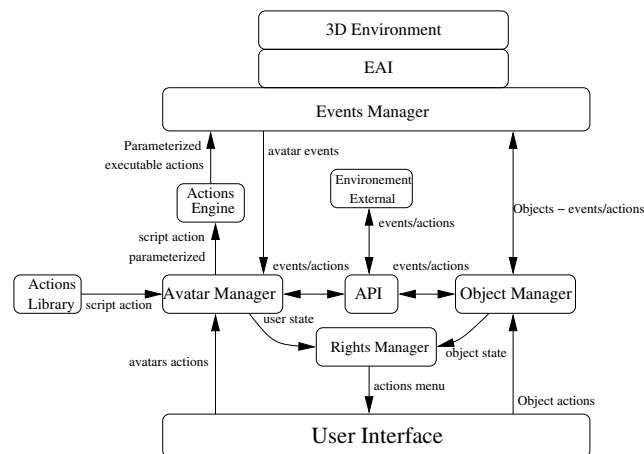


Figura 2: Arquitetura do Sistema

### 5.1 Aspectos de Implementação

Nosso protótipo de AVC é baseado no sistema VNet. VNet utiliza o protocolo *VIP - VRML Interchange Protocol* para garantir a comunicação entre um cliente e o servidor; o cliente é composto de uma interface JAVA e um browser Xj3D. O servidor garante a consistência da base de dados. Nosso modelo de avatar segue o padrão *H-Anim* (Humanoid Animation Working Group)<sup>6</sup>. *H-Anim* é um padrão de descrição de humanoídes animados. Mas, especificar uma ação seguindo o padrão *H-Anim* é uma tarefa

<sup>5</sup>www.skype.com

<sup>6</sup>http://www.h-anim.org

complicada e nada intuitiva por conta da distância entre *H-Anim*, uma linguagem de animação, e a linguagem natural. Para tornar a composição de ações uma tarefa mais fácil, nós adotamos *STEP* (*Script Technology for Embodied Persona*) [13]. STEP dispõe de uma interface com um rico conjunto de operadores de ação prontos e disponíveis para serem utilizados. O núcleo de STEP traduz o script definindo a ação em comandos H-Anim que serão executados através da EAI/VRML.

## 6 Perspectivas para Trabalhos Futuros

Neste artigo, propomos um AVC que fornece um módulo de gerenciamento direitos de acesso dos usuários sobre os objetos. A nossa abordagem, baseada no estado do usuário e no estado do objeto, proporciona um controle de acesso adaptativo. Além disso, definimos uma API que permite a colaboração além do contexto da cena virtual. Entre as perspectivas para trabalhos futuros estão a definição de uma arquitetura distribuída e a definição de um esquema de especificação das regras de acessibilidade.

## Referências

- [1] “DIS (IEEE Std 1278.1), Distributed Interactive Simulation -Application Protocols”, 1995.
- [2] “HLA (IEEE Standard for Modeling and Simulation (M&S)), High Level Architecture Framrwork and Rules - IEEE 1516-2000”, 2000.
- [3] M. Capps, D. McGregor, D. Brutzman e M. Zyda, “Npsnet-v: A new beginning for dynamically extensible virtual environments”, in *IEEE Computer Graphics and Applications*, vol. 20, pp. 12–15, 2000.
- [4] “Web3d Consortium - Living Worlds Working Groups”, 1997.
- [5] N. Richard, “InViWo agents: Write once, display everywhere”, in *Proceedings of the Web3D Conference*, 2003.
- [6] Y. Zhang, L. Guo e N. D. Georganas, “AGILE: An architecture for agent-based collaborative and interactive virtual environments”, in *Proc. Workshop on Application Virtual Reality Technologies for Future Telecommunication System, IEEE Globecom’2000 Conference*, 2000.
- [7] N. I. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao e M. Palmer, “Parameterized action representation for virtual human agents”, *Embodied conversational agents*, pp. 256–284, 2000.
- [8] E. Frecon e M. Stenius, “DIVE: A scalable network architecture for distributed virtual environments”, in *Distributed Systems Engineering Journal (special issue on Distributed Virtual Environments)*, vol. 5, pp. 91–100, 1998.
- [9] G. de Jesús Hoyos Rivera, *COLAB: Conception et mise en oeuvre d’un outil pour la navigation coopérative sur le WEB*. Tese de Doutorado, Université Paul Sabatier - Toulouse - France, 2005.
- [10] V. Baudin, “Supporting distributed experts in e-meetings for synchronous collaboration”, in *Proc. IEEE International Conference on Systems, Man and Cybernetics, SMC’02, Tunisie*, 2002.
- [11] L. Ieronutti e L. Chittaro, “A virtual human architecture that integrates kinematic, physical and behavioral aspects to control h-anim characters”, in *Web3D ’05: Proceedings of the tenth international conference on 3D Web technology*, New York, NY, USA, pp. 75–83, ACM Press, 2005.
- [12] R. Milner, “A calculus of communicating systems - CCS”, 1980.
- [13] Z. Huang, A. Eliens e C. Visser, “STEP: A Script Language for Embodied Agents”, in *PRICAI-02 International Workshop on Lifelike Animated Agents. Tools, Agents and Applications* (H. Prendiger, ed.), pp. 46–51, 2002.