

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**UMA ARQUITETURA PARA DISTRIBUIÇÃO DE  
AMBIENTES VIRTUAIS DE REALIDADE AUMENTADA**

**ORIENTADOR: EDGARD LAMOUNIER JÚNIOR, PhD.**

**CO-ORIENTADOR: ALEXANDRE CARDOSO, Dr.**

**ORIENTANDO: WENDER ANTÔNIO DA SILVA**

**JULHO, 2008**

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**UMA ARQUITETURA PARA DISTRIBUIÇÃO DE AMBIENTE  
VIRTUAIS DE REALIDADE AUMENTADA**

Dissertação apresentada por Wender Antônio da Silva  
à Universidade Federal de Uberlândia para obtenção  
do título de Mestre em Ciências.

Professor Edgard Lamounier Jr., PhD – UFU- (Orientador)

Professor Alexandre Cardoso, Dr – UFU - (Co-Orientador)

Professor Marcos Wagner Ribeiro, Dr – ILES/ULBRA-Itumbiara

Professor Antônio Carlos Sementille, Dr –UNESP/Bauru

Uberlândia, 14 de julho de 2008.

Dados Internacionais de Catalogação na Publicação (CIP)

---

S586u Silva, Wender Antônio da, 1977-

Uma arquitetura para distribuição de ambiente virtuais de

realidade aumentada / Wender Antônio da Silva. - 2008.

82 f. : il.

Orientador:.Edgard Lamounier Jr.

Dissertação (mestrado) – Universidade Federal de  
Uberlândia, Pro-  
grama de Pós-Graduação em Engenharia Elétrica.  
Inclui bibliografia.

1. Realidade virtual - Teses. I. Lamounier Júnior, Edgard.  
II.Univer-sidade Federal de Uberlândia. Programa de Pós-  
Graduação em Enge-  
nharia Elétrica. III. Título.

CDU:

---

681.3:007.52

**UMA ARQUITETURA PARA DISTRIBUIÇÃO DE  
AMBIENTES VIRTUAIS DE REALIDADE AUMENTADA**

**WENDER ANTÔNIO DA SILVA**

Dissertação apresentada por Wender Antônio da Silva à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Profº Edgard Lamounier Júnior  
Orientador

Profº. Darizon Alves Andrade  
Coordenador do Curso de Pós-Graduação

## **DEDICATÓRIA**

Aos meus pais José Alves da Silva e Ana Maria da Silva.

Aos meus irmãos Weidiman Alves da Silva e Wallisson Alves da Silva.

A minha mentora e amiga Nadabe Fortes.

Aos meus amigos, que, nas horas mais difíceis me incentivaram.

A todos que contribuíram direta ou indiretamente para a finalização deste trabalho.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, que permitiu a finalização de mais uma etapa em minha vida.

Ao professor e amigo Edgard Lamounier Jr., orientador deste trabalho, pela valiosa orientação e por me mostrar sempre o caminho a seguir.

Ao professor Alexandre Cardoso pelo apoio na co-orientação deste trabalho.

Um agradecimento especial ao professor Marcos Wagner de Souza Ribeiro pela amizade e, pela valiosa co-orientação no desenvolvimento deste trabalho, sempre pronto a ajudar, esteve sempre ao meu lado nos momentos difíceis.

Aos colegas do Laboratório de Realidade Virtual de Goiás, onde foi desenvolvida boa parte desta dissertação, principalmente ao colega Rodrigo Vasconcelos Arruda pela ajuda na modelagem dos objetos virtuais.

Aos colegas do Grupo de Realidade Virtual da Faculdade de Engenharia Elétrica da Universidade de Uberlândia, principalmente ao amigo Ezequiel Roberto Zorral pela primordial ajuda com o ARToolKit.

## RESUMO

SILVA, Wender A. *UMA ARQUITETURA PARA DISTRIBUIÇÃO DE AMBIENTES VIRTUAIS DE REALIDADE AUMENTADA*, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2008, 82p.

Esta dissertação apresenta uma arquitetura para distribuição de ambientes virtuais de Realidade Aumentada como ferramenta de apoio a projetos de ensino. Considerando os trabalhos relacionados com o referido tópico, percebe-se que suas interfaces são pouco flexíveis para suportar interações com os objetos virtuais distribuídos. Diante disto, tem-se como objetivo para esta dissertação, a criação de uma interface de interação com o usuário e uma camada para distribuição das interações realizadas. Assim, para a realização do protótipo de software, a interface de Realidade Aumentada está baseada em um estudo de caso que representa uma área específica do conhecimento, neste caso, a fisiologia vegetal. Desta forma, foi criado uma interface de Realidade Aumentada que possibilita a interação do usuário com o ambiente virtual por meio de menus de interação e também por meio de marcadores. Esta interface é integrada a uma camada de distribuição. Assim, a camada de distribuição foi implementada seguindo a metodologia cliente/servidor baseada em objetos. Com a implementação, as interações feitas em um ambiente são replicadas para os outros ambientes e vice-versa, gerando assim, um sistema fortemente aplicável à educação, pois pode proporcionar múltiplas visualizações dos objetos virtuais sobre os objetos reais. O sistema foi apresentado e avaliado por professores e alunos de uma Instituição de Ensino Superior da cidade de Itumbiara, Goiás. Assim, professores e alunos das áreas de informática e de biologia analisaram o protótipo e responderam um questionário baseado em ISONORMS. Posteriormente, segundo os avaliadores, pode-se constatar que os resultados alcançados permitiram concluir que o protótipo é eficaz e aplicável.

**Palavras-Chave:** Ambientes Virtuais Distribuídos, CORBA, Realidade Aumentada.

## **ABSTRACT**

SILVA, Wender A. ***AN ARCHITECTURE FOR DISTRIBUTION OF AUGMENTED VIRTUAL ENVIRONMENTS***, Uberlândia, Faculty of Electrical Engineering - UFU, 2008, 82p.

This thesis presents an architecture for distribution of virtual environments in Augmented Reality as a tool to support projects in education. Subsequently, according to the evaluators, you can see that the results showed that the prototype is effective and enforceable. In face of this, this work aims creating an interface for interaction with the user and a layer for distribution of made interactions. Thus, for the creation of the software prototype, the interface of Augmented Reality is based on a study case that represents a specific area of knowledge, in this case the plant physiology. Thus was created an interface of Augmented Reality that enables user interaction with the virtual environment through menus for interaction and also by means of labels. This interface is integrated with a layer of distribution. Thus, the layer of distribution was implemented following the methodology client / server-based objects. With the implementation, the interactions made in an environment are replicated for other environments and vice versa, thereby generating a strong system applicable to education as it can provide multiple views of virtual objects on the real objects. The system was submitted and evaluated by teachers and students of a Educational Institution, a Faculty of Itumbiara city, Goiás. Thus teachers and students of information technology and biology areas examined the prototype and answered a questionnaire based on ISONORM. Later, according to the evaluators, it could be seen that the results showed the prototype is effective and enforceable.

**Key-Words:** Distributed Virtual Environments, CORBA, Augmented Reality.

## PUBLICAÇÕES

A seguir são apresentadas as publicações resultantes desse trabalho:

SILVA, Wender A.; LAMOUNIER JR, Edgard.; CARDOSO, Alexandre; RIBEIRO, Marcos Wagner de Souza. *Arquitetura para distribuição de Ambiente de Realidade Aumentada.* SVR 2008 X Symposium on Virtual Reality, 10, 2008, João Pessoa. *Proceedings...* João Pessoa: UFPB, 2008.

SILVA, Wender A.; LAMOUNIER JR, Edgard; CARDOSO, Alexandre; RIBEIRO, Marcos Wagner de Souza. *Interface para distribuição e integração de Realidade Aumentada com Realidade Virtual por meio da plataforma CORBA, tendo como estudo de caso ambientes multidisciplinares de biologia e química.* MT Digital 2008. Mato Grosso Digital 2008, 1, 2008, Cuiabá. *Proceedings...* Cuiabá: SUCESU, 2008.

SILVA, Wender A.; LAMOUNIER JR, Edgard; CARDOSO, Alexandre; RIBEIRO, Marcos Wagner de Souza. *Ambientes Virtuais Distribuídos Multidisciplinares com Integração na Realidade Aumentada.* SVR 2007 IX Symposium on Virtual Reality, 9, 2007, Petrópolis. *Proceedings...* Petrópolis: LNCC, 2007.

SILVA, Wender A.; RIBEIRO, Marcos Wagner de Souza. *Ambientes Virtuais Distribuídos multidisciplinares usando Realidade Aumentada.* Revista Práxis, Canoas-RS, n. 6, p49-56, jan/jul, 2006.

SILVA, Wender Antônio da; RIBEIRO, Marcos Wagner de Souza. *Visualização de Ambientes Virtuais Distribuídos Multidisciplinares por meio de Realidade Aumentada.* SVR 2006 VIII Symposium on Virtual Reality, 6, 2006, Belém. *Proceedings...* Belém: CESUPA, 2006.

SILVA, Wender A.; LAMOUNIER JR, Edgard; CARDOSO, Alexandre; RIBEIRO, Marcos Wagner de Souza. *Ambientes Virtuais Distribuídos multidisciplinares com integração na Realidade Aumentada.* WARV 2006 II Workshop de Aplicações de Realidade Virtual, 2, 2006, Recife. *Proceedings...* Recife: UFPE, 2006.

SILVA, Wender A.; LAMOUNIER JR, Edgard; CARDOSO, Alexandre; RIBEIRO, Marcos Wagner de Souza. *Arquitetura para Distribuição de Ambientes Virtuais utilizando Realidade Aumentada.* WARV 2006 II Workshop de Aplicações de Realidade Virtual, 2, 2006, Recife. *Proceedings...* Recife: UFPE, 2006.

# SUMÁRIO

<b>CAPÍTULO I .....</b>	<b>1</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1 CONSIDERAÇÕES INICIAIS.....	1
1.2 OBJETIVOS.....	2
1.3 CONTRIBUIÇÕES DESSE TRABALHO.....	2
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO.....	3
<b>CAPÍTULO II.....</b>	<b>4</b>
<b>2. FUNDAMENTOS TECNOLÓGICOS .....</b>	<b>4</b>
2.1 INTRODUÇÃO .....	4
2.2 REALIDADE VIRTUAL .....	4
2.3 AMBIENTES VIRTUAIS.....	5
2.4 REALIDADE AUMENTADA.....	5
2.4.1 <i>Aplicações Educacionais em Ambientes de Realidade Aumentada</i> .....	7
2.4.2 <i>Intervenções Educacionais com Realidade Aumentada</i> .....	7
2.5. SISTEMAS DISTRIBUÍDOS DE REALIDADE VIRTUAL E AUMENTADA.....	8
2.5.1. <i>Comparaçao das plataformas CORBA, DCOM e JAVA/RMI</i> .....	11
2.5.2 <i>Escolha da Arquitetura</i> .....	12
2.6 O QUE É DISTRIBUIÇÃO?.....	13
2.6.1. <i>CORBA</i> .....	13
2.6.2. <i>ORB – Object Request Broker</i> .....	14
2.7 CONSIDERAÇÕES FINAIS .....	16
<b>CAPÍTULO III .....</b>	<b>17</b>
<b>3. TRABALHOS RELACIONADOS .....</b>	<b>17</b>
3.1 INTRODUÇÃO .....	17
3.1.1 <i>Metodologia para a análise</i> .....	17
3.2 INTERFACES DE REALIDADE AUMENTADA .....	18
3.2.1 <i>Interactive Entertainment Systems Using Tangible Cubes</i> .....	18
3.2.2 <i>Visualização de dados relativos a redes de computadores usando Realidade Virtual e Aumentada</i> ..	19
3.3 REALIDADE VIRTUAL DISTRIBUÍDA.....	21
3.3.1 <i>Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares</i> .....	21
3.4 REALIDADE AUMENTADA DISTRIBUÍDA .....	22
3.4.1 <i>Distributed Applications for Collaborative Augmented Reality</i> .....	22
3.4.2 <i>Ambientes Colaborativos com Realidade Aumentada</i> .....	24
3.4.3 <i>Distributed Augmented Chess System</i> .....	25
3.5 CONSIDERAÇÕES FINAIS .....	26
<b>CAPÍTULO IV .....</b>	<b>29</b>
<b>4. ARQUITETURA DO SISTEMA .....</b>	<b>29</b>
4.1 INTRODUÇÃO .....	29
4.2 TECNOLOGIAS DE APOIO .....	29
4.2.1 <i>CORBA</i> .....	29
4.2.2 <i>Visibroker</i> .....	29
4.2.2.1. <i>Osagent</i> .....	30
4.2.3 <i>ARToolKit</i> .....	30
4.3 ARQUITETURA PARA DISTRIBUIÇÃO DO SISTEMA .....	31
4.4 PIPELINE DO SISTEMA .....	33
4.5 CONSIDERAÇÕES FINAIS .....	35
<b>CAPÍTULO V .....</b>	<b>36</b>
<b>5. DETALHES DE IMPLEMENTAÇÃO .....</b>	<b>36</b>

5.1 INTRODUÇÃO .....	36
5.2 MODELAGEM DOS OBJETOS VIRTUAIS.....	36
5.3 IMPLEMENTAÇÃO DA INTERFACE DE REALIDADE AUMENTADA .....	36
5.3.1 <i>Implementação das Interações via Teclado</i> .....	37
5.3.2 <i>Implementação dos Menus da Interface de Realidade Aumentada</i> .....	37
5.3.3 <i>Implementação da opção de troca de objeto virtual por meio do menu suspenso</i> .....	39
5.3.4 <i>Implementação das Interações por meio de Marcadores</i> .....	40
5.3.7 <i>Interação com os Menus da Interface de Realidade Aumentada</i> .....	41
5.3.8 <i>Implementação dos parâmetros para Distribuição</i> .....	42
5.4 IMPLEMENTAÇÃO DA ARQUITETURA DE COMUNICAÇÃO.....	44
5.4.1 <i>Codificação do Objeto-Cliente</i> .....	46
5.4.2 <i>Codificação do Objeto-Servidor</i> .....	49
5.5 CONSIDERAÇÕES FINAIS .....	50
<b>CAPÍTULO VI .....</b>	<b>52</b>
<b>6. FUNCIONAMENTO DO SISTEMA E ESTUDO DE CASO .....</b>	<b>52</b>
6.1 INTRODUÇÃO .....	52
6.2 ESTUDO DE CASO .....	52
6.2.1. <i>Fisiologia Vegetal</i> .....	52
6.2.2. <i>Transporte de Solutos (Seiva)</i> .....	53
6.3 FUNCIONAMENTO DO SISTEMA .....	54
6.3.1. <i>Funcionamento da Camada de Distribuição</i> .....	54
6.3.2. <i>Funcionamento da Interface de Realidade Aumentada</i> .....	55
6.4 DISTRIBUIÇÃO DA INFORMAÇÃO .....	62
6.5 CONSIDERAÇÕES FINAIS .....	63
<b>CAPÍTULO VII.....</b>	<b>65</b>
<b>7. AVALIAÇÃO, RESULTADOS E LIMITAÇÕES .....</b>	<b>65</b>
7.1 INTRODUÇÃO .....	65
7.2. AVALIAÇÃO DO SISTEMA R.A DISTRIBUÍDA.....	65
7.2.1. <i>Latência da Comunicação e Escalabilidade</i> .....	66
7.2.2. <i>Resultados da avaliação do sistema</i> .....	66
7.3. RECURSOS NÃO IMPLEMENTADOS .....	74
7.4. CONSIDERAÇÕES FINAIS .....	74
<b>CAPÍTULO VIII .....</b>	<b>76</b>
<b>8. CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>76</b>
8.1 INTRODUÇÃO .....	76
8.2 CONCLUSÕES .....	76
8.2.1 <i>Contribuições do Trabalho</i> .....	76
8.3. <i>Trabalhos Futuros</i> .....	78
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>79</b>

## LISTA DE FIGURAS

<i>Figura 2.1 - Diagrama de realidade/virtualidade contínua.....</i>	6
<i>Figura 2.2. Latência dos protótipos .....</i>	12
<i>Figura 2.3. Requisição por meio do ORB .....</i>	15
<i>Figura 3.1. Interação de ambiente com marcador “pá” .....</i>	18
<i>Figura 3.2. Magic History Cube em ação (história bíblica).....</i>	19
<i>Figura 3.3. VR TraceRoute .....</i>	20
<i>Figura 3.4. AR TraceRoute .....</i>	20
<i>Figura 3.5. Ambiente Virtual de Biologia .....</i>	21
<i>Figura 3.6. Ambiente Virtual de Química .....</i>	22
<i>Figura 3.7. Interface de colaboração e projeção em monitores da aplicação.....</i>	23
<i>Figura 3.8. Interface de colaboração e projeção em Realidade Aumentada .....</i>	23
<i>Figura 3.9. Visualização de objetos virtuais em RA compartilhando um ambiente.....</i>	24
<i>Figura 3.10. Jogo em colaboração em uma rede .....</i>	25
<i>Figura 4.1. Arquitetura proposta para a distribuição do sistema R.A. Distribuída.....</i>	31
<i>Figura 4.2. Diagrama da interface R.A adaptada à Arquitetura do ARToolKit .....</i>	34
<i>Figura 5.1. Esquema dos menus interativos e seus objetos virtuais .....</i>	37
<i>Figura 5.2 Relação de objetos virtuais por menu interativo.....</i>	38
<i>Figura 5.3. Parte da implementação dos menus interativos .....</i>	38
<i>Figura 5.4. Menu suspenso na Interface de R.A.....</i>	39
<i>Figura 5.6. Parte do pseudocódigo da função gerenciamouse .....</i>	39
<i>Figura 5.7. Parte da implementação das transformações geométrica por meio de colisão aproximada .....</i>	40
<i>Figura 5.8. Parte da implementação das interações por colisão com os menus interativos.....</i>	41
<i>Figura 5.9. Camada de Comunicação (módulo entrada).....</i>	42
<i>Figura 5.10 Camada de Comunicação (módulo saída) .....</i>	42
<i>Figura 5.11. Parte do código do tratamento de Oclusão.....</i>	43
<i>Figura 5.12. Parte da verificação de objetos vindos do servidor para renderização.....</i>	43
<i>Figura 5.13. Plataforma CORBA 2.0 .....</i>	44
<i>Figura 5.14. IDL da Camada de Distribuição .....</i>	45
<i>Figura 5.15. Visão parcial do arquivo de implementação .....</i>	46
<i>Figura 5.16. Gravação e leitura da camada de comunicação (Cliente) .....</i>	47
<i>Figura 5.17 Inicialização do Objeto-Corba no Cliente .....</i>	48
<i>Figura 5.18. Interface da aplicação distribuída (Cliente) .....</i>	49
<i>Figura 5.19. Verificação de tempo no Cliente .....</i>	49
<i>Figura 5.20. Aplicação Servidora .....</i>	50
<i>Figura 5.21. Inicialização da Aplicação Servidora .....</i>	50
<i>Figura 6.1 Vasos para transporte de Seiva.....</i>	53
<i>Figura 6.2 Vasos de Xilema .....</i>	54

<i>Figura 6.3 Interface da aplicação de distribuição – Cliente.....</i>	55
<i>Figura 6.4 Interface da Aplicação de Interface de Realidade Aumentada .....</i>	56
<i>Figura 6.5 Interação: pegando objeto virtual no menu Interativo.....</i>	57
<i>Figura 6.6 Interação: pegando objeto virtual no menu Interativo.....</i>	58
<i>Figura 6.7a Inicio da distribuição (pc1 com web1) .....</i>	59
<i>Figura 6.7b Distribuição do objeto virtual na árvore com outro visão (pc2 com web2).....</i>	60
<i>Figura 6.7c Distribuição do objeto virtual com outro visão (pc3 com web3).....</i>	60
<i>Figura 6.8 Aumentando a escala dos objetos virtuais com a utilização de marcador.....</i>	61
<i>Figura 6.9 Transferência de um objeto virtual de um marcador para outro .....</i>	61
<i>Figura 6.10 Finalização da transferência do objeto de um marcador para outro.....</i>	61
<i>Figura 7.2. Gráfico: Interface (usabilidade).....</i>	67
<i>Figura 7.3. Gráfico: Facilidade de uso.....</i>	67
<i>Figura 7.4. Gráfico: Nível de Interação na cena .....</i>	68
<i>Figura 7.5. Gráfico: Manipulação dos objetos virtuais na cena .....</i>	68
<i>Figura 7.6. Gráfico: Manipulação dos Menus Interativos.....</i>	69
<i>Figura 7.7. Gráfico: Colaboração entre usuários .....</i>	69
<i>Figura 7.8. Gráfico: Aquisição de conhecimento .....</i>	70
<i>Figura 7.9. Gráfico: Concepção da experiência proposta.....</i>	71
<i>Figura 7.10. Gráfico: Funções Existentes no sistema.....</i>	71
<i>Figura 7.11. Gráfico: Confiança no sistema.....</i>	72
<i>Figura 7.12. Gráfico: Número de passos necessários para realizar uma atividade.....</i>	72
<i>Figura 7.13. Gráfico: Controle sobre as atividades .....</i>	73
<i>Figura 7.14. Gráfico: Visualização clara do que está sendo executado.....</i>	73
<i>Figura 7.15. Gráfico: Visão Geral do Sistema.....</i>	74

## **LISTA DE TABELAS**

<i>Tabela 2.1: Comparação entre plataformas de distribuição.....</i>	<i>14</i>
<i>Tabela 3.1: Características avaliadas em relação ao sistema desenvolvido neste trabalho .....</i>	<i>27</i>
<i>Tabela 3.2: Características avaliadas em relação ao sistema desenvolvido neste trabalho .....</i>	<i>27</i>
<i>Tabela 3.3: Características avaliadas em relação ao sistema desenvolvido neste trabalho .....</i>	<i>28</i>
<i>Tabela 6.1:Realidade Virtual e Aumentada Distribuída – Comparações e contribuições.....</i>	<i>64</i>

## LISTA DE ABREVIATURAS E SIGLAS

2D	– Bidimensional
3D	– Tridimensional
API	– <i>Application Programming Interface</i>
AV	– Ambiente Virtual
AVD	– Ambiente Virtual Distribuído
BOA	– <i>Basic Object Adapter</i>
COM	– <i>Common Object Model</i>
CORBA	– <i>Common Object Request Broker Architecture</i>
DCE	– <i>Distributed Computing Environment</i>
DIVE	– <i>Distributed Interactive Virtual Environment</i>
GUI	– <i>Graphics User Interface</i>
HMD	– <i>Head Mounted Display</i>
IDL	– <i>Interface Definition Language</i>
ISO	– <i>International Standards Organization</i>
OMA	– <i>Object Management Architecture</i>
OMG	– <i>Object Management Group</i>
OpenGL	– <i>Open Graphics Library</i>
ORB	– <i>Object Request Broker</i>
TCP	– <i>Transfer Communication Protocol</i>
VR	– <i>Virtual Reality</i>
VRML	– <i>Virtual Reality Modeling Language</i>
RA	- <i>Realidade Aumentada</i>
RV	- <i>Realidade Virtual</i>
PC	- <i>Personal Computer</i>
HITLAB	- <i>Human Interface Technology Laboratory</i>
GLP	- <i>General Licence Public</i>

# CAPÍTULO I

## 1. INTRODUÇÃO

### 1.1 Considerações Iniciais

A Realidade Virtual (RV) é, antes de tudo, uma “interface avançada do usuário” para acessar aplicações executadas no computador, tendo como características, a visualização de, e movimentação em, ambientes tridimensionais em tempo real e a interação com elementos do mesmo (TORI, KIRNER & SISCOUTTO, 2006). Nesse aspecto pode-se considerar que a Realidade Virtual é considerada uma tecnologia revolucionária, pois possibilita a simulação de mundos reais e imaginários na tela do computador ou em outros dispositivos, criando no usuário a sensação de presença em um “mundo” virtual.

Relacionado com a Realidade Virtual, a Realidade Aumentada (RA) é a inserção de objetos virtuais no mundo real por meio de um dispositivo computacional. Desta forma, a interface do usuário é aquela, que ele usa no ambiente real, adaptada para visualizar e manipular os objetos virtuais colocados no seu espaço (AZUMA, 2001).

Entende-se que a Realidade Aumentada, enriquece o ambiente físico com objetos virtuais, assim essa, foi beneficiada pelo progresso da tecnologia da computação, tornando viável aplicações desta tecnologia, tanto em plataformas sofisticadas quanto em plataformas populares. Então, diferentemente da Realidade Virtual, que transporta o usuário para o ambiente virtual, a Realidade Aumentada mantém o usuário no seu ambiente físico e transporta o ambiente virtual para o espaço do usuário, permitindo a interação com o mundo virtual, de maneira mais natural e sem necessidade de treinamento ou adaptação (TORI, KIRNER & SISCOUTTO, 2006). Nesse contexto, novas interfaces multimodais vêm sendo desenvolvidas para facilitar a manipulação de objetos virtuais no espaço do usuário, usando as mãos ou dispositivos mais simples de interação.

Assim, as pesquisas em Realidade Virtual e Aumentada vêm crescendo consideravelmente por meio de vários grupos de pesquisa ligados à indústria, ao entretenimento e, principalmente à educação. Além disso, existem muitos estudos, soluções e implementações para possibilitar que mais de uma pessoa faça parte de um Ambiente Virtual (JING et al., 2005).

Neste sentido o objeto de estudo desta dissertação é a distribuição de ambientes de Realidade Aumentada, deixando claro que as principais pesquisas concentram-se na melhoria

do processo de comunicação entre as cópias de um ambiente. Porém, dependendo da aplicação, nem sempre a preocupação será com a comunicação entre cópias de um ambiente, mas também com a comunicação entre ambientes virtuais distintos, além da colaboração nos ambientes virtuais. Sendo assim, a motivação para esta dissertação é a necessidade de pesquisas e desenvolvimento de técnicas para distribuição de ambientes virtuais para Realidade Aumentada com suporte a uma interface que possua flexibilidade considerável para interações de objetos virtuais distribuídos, onde a distribuição deve ocorrer em decorrência de interações realizadas em interfaces de Realidade Aumentada.

## **1.2 Objetivos**

Essa dissertação tem por objetivo apresentar uma abordagem computacional/algorítmica e, ainda, uma arquitetura que seja suficiente para suportar o funcionamento de um Ambiente Virtual de Realidade Aumentada Distribuída. Assim, para realizar esta distribuição faz-se necessário a implementação de uma interface de Realidade Aumentada que possua um nível razoável de interação.

Desta forma, para atingir os objetivos, as seguintes metas foram definidas:

1. Escolha de uma ferramenta para a programação da interface de Realidade Aumentada.
2. Escolha de uma plataforma para a implementação da distribuição dos objetos virtuais.
3. Realizar a implementação de um protótipo baseado em uma interface de Realidade Aumentada interativa, e uma camada para a distribuição dos ambientes virtuais de Realidade Aumentada.
4. Escolha de um estudo de caso para implementação (*Fisiologia Vegetal*).
5. Validar o protótipo junto a usuários, de forma a obter realimentação que possibilite melhorias e adequações necessárias ao aprimoramento da abordagem escolhida.

## **1.3 Contribuições desse trabalho**

Essa dissertação pretende contribuir com a descrição de uma abordagem (arquitetura) que possibilite a criação de um *software* unindo duas áreas da computação (Sistemas Distribuídos e Realidade Aumentada) na construção de Ambientes Virtuais Aumentados Distribuídos, usando uma plataforma de distribuição baseada em objetos e o padrão de um determinado *toolkit* de programação para Realidade Aumentada. Esta aplicação facilitará a implementação de ambientes virtuais de Realidade Aumentada Distribuída, uma vez que, proporciona a implementação de uma “camada” para distribuição de Ambientes Virtuais Aumentados, independentes da área de aplicação.

Este trabalho apresenta ainda um algoritmo desenvolvido para possibilitar a interação do usuário com os objetos virtuais. Essas interações são realizadas por meio de transformações geométricas que desencadeiam colisões, oclusões e, ainda troca de objetos na cena. Assim, o software proporciona um nível satisfatório de interações, as quais devem ser distribuídas em uma rede de computadores.

## 1.4 Organização da Dissertação

O trabalho está dividido em oito capítulos, descritos resumidamente a seguir:

**Capítulo I – Introdução:** São apresentadas as considerações iniciais da dissertação, assim como os objetivos e a contribuição e, ainda a organização dos capítulos deste trabalho.

**Capítulo II – Fundamentos Tecnológicos:** Aborda o referencial teórico com os principais conceitos de Realidade Virtual e Aumentada e, ainda os Ambientes de Realidade Virtual Distribuída, destacando suas características básicas e estrutura de funcionamento.

**Capítulo III – Trabalhos Relacionados:** Apresenta o estado da arte na área da pesquisa. Neste capítulo dividiu-se as avaliações dos trabalhos relacionados três características fundamentais: *Interfaces de Realidade Aumentada*, *Realidade Virtual Distribuída* e *Realidade Aumentada Distribuída*.

**Capítulo IV – Arquitetura do Sistema:** Descreve-se a arquitetura do sistema, as técnicas utilizadas para a implementação da interface e da camada de distribuição.

**Capítulo V – Detalhes de Implementação:** Apresenta a implementação do sistema, desta forma é efetuada uma descrição sintética da interface desenvolvida.

**Capítulo VI – Funcionamento do Sistema e Estudo de Caso:** Descreve o funcionamento do sistema proposto e, ainda o estudo de caso proposto para o desenvolvimento do mesmo.

**Capítulo VII – Avaliação, Resultados e Limitações:** Relata os resultados, bem como suas limitações, obtidos por meio de testes aplicado a usuários sobre questões de usabilidade e importância da aplicação para área educacional.

**Capítulo VIII – Conclusões e Trabalhos Futuros:** Apresenta as conclusões e sugestões de trabalhos futuros.

## CAPÍTULO II

### 2. FUNDAMENTOS TECNOLÓGICOS

#### 2.1 Introdução

Este capítulo aborda os conceitos fundamentais de Realidade Virtual, Realidade Aumentada, Sistemas Distribuídos e Ambientes Virtuais Distribuídos necessários à compreensão desta dissertação.

#### 2.2 Realidade Virtual

Realidade Virtual é uma interface avançada para aplicações computacionais, que permite ao usuário navegar e interagir, em tempo real, com um ambiente tridimensional gerado por computador, usando dispositivos multisensoriais (KIRNER & SISCOUTTO, 2007).

Entende-se que a interação em ambientes virtuais consiste na capacidade do usuário atuar em ambientes virtuais, promovendo alterações e reações às suas ações (KIRNER & SISCOUTTO, 2007).

Ressalta-se que a Realidade Virtual é o uso do computador e de interfaces humano-computador para criar o efeito em ambientes tridimensionais com objetos interativos, apoiado em três fundamentos: *interação, imersão e navegação* (KIRNER & TORI, 2004).

Neste contexto é necessário um forte senso de presença neste espaço tridimensional (imersão), permitindo experimentar (navegação) e interagir (interação) de forma a propiciar sensações de prazer e de conhecimento (KIRNER & TORI, 2004).

Nessa perspectiva, entende-se que a Realidade Virtual envolve tecnologias revolucionárias para o desenvolvimento de aplicações, em função de novas possibilidades que oferecem interfaceamento com o uso de dispositivos multisensoriais, navegação em espaços tridimensionais, imersão no contexto da aplicação e interação em tempo real (CARDOSO et al., 2007).

Assim, verifica-se que trabalhar com um ambiente virtual permite aos usuários entender melhor os relacionamentos entre os dados e encorajar colaboração e tomada de decisões (BOTELHO, 2002). Ainda, pode-se destacar o uso da Realidade Virtual na educação e em jogos educativos, onde busca-se explorar a integração e a interação dentro do processo de aprendizado, despertando o senso de análise dos alunos (FREITAS et al., 2007).

Entendendo que o senso de presença é primordial para o bom funcionamento de um sistema baseado em Realidade Virtual, a mesma pode ser classificada, em imersiva e não imersiva (KIRNER & TORI, 2004).

Assim, a **Realidade Virtual é imersiva**, quando o usuário em teoria é transportado predominantemente para o domínio da aplicação, por meio de dispositivos multisensoriais, que capturam seus movimentos e comportamento e reagem a eles (KIRNER & TORI, 2004).

Por outro lado, a **Realidade Virtual é não imersiva**, quando o usuário é, em teoria transportado parcialmente ao mundo virtual, por meio de uma janela, mas continua a sentir-se predominantemente no mundo real (KIRNER & TORI, 2004).

## 2.3 Ambientes Virtuais

De acordo com Sementille (2000), um Ambiente Virtual (AV) é um ambiente com o qual há a imersão do usuário em um ambiente tridimensional simulado. Um Ambiente Virtual também pode ser entendido como um sistema de software que cria a ilusão de um mundo que não existe na realidade. Isto requer a combinação de entrada (interação do usuário), computação (simulação de processos) e saída (estímulos multisensoriais).

Os objetos no Ambiente Virtual podem corresponder a objetos que existem no mundo real. Quando isso acontece, espera-se que esses objetos reajam como os objetos reais. Portanto, pode haver a necessidade de se modelar, no mínimo, uma parte da Física (movimento e detecção de colisão entre objetos, por exemplo).

Alguns objetos no Ambiente Virtual devem ser capazes de responder às ações dos usuários e às ações de outros objetos: devem existir meios de especificar esses comportamentos e associá-los aos objetos no modelo. Alguns objetos podem ter comportamentos autônomos, como flutuar, por exemplo. Isto envolve interações entre os objetos (SEMENTILLE, 2000).

Em um Ambiente Virtual, o usuário deve poder navegar em três dimensões, ou seja, com seis graus de liberdade. Cada grau se aplica a uma direção ou rotação do movimento. Na verdade, conferem ao usuário a possibilidade de se movimentar em seis direções simultâneas: translação em torno dos três eixos cartesianos (x, y e z) e rotação em torno de cada um (SEMENTILLE, 2000).

## 2.4 Realidade Aumentada

Pode-se definir Realidade Aumentada (RA) como a amplificação da percepção sensorial por meio de recursos computacionais. Assim, associando dados computacionais ao

mundo real, a Realidade Aumentada permite uma interface mais natural com dados e imagens geradas por computador (CARDOSO et al., 2007). Desta forma, entende-se que um sistema de Realidade Aumentada deve prover ao usuário condições de interagir com estes dados de forma mais natural possível.

Milgram et al. (1994) define Realidade Aumentada como a sobreposição de objetos virtuais tridimensionais, gerados por computador, com um ambiente real, por meio de algum dispositivo tecnológico. Entretanto, esta conceituação é muito geral, e fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

De acordo com Kirner e Tori (2004), Realidade Misturada, mistura o real com o virtual, abrange duas possibilidades: a Realidade Aumentada, cujo ambiente predominante é o mundo real, e a Virtualidade Aumentada, cujo ambiente predominante é o mundo virtual. Pode-se dizer, então, que a Realidade Aumentada é uma particularização da Realidade Misturada. A Figura 2.1, apresenta o diagrama de realidade/virtualidade contínua, mostrando as possibilidades gradativas de sobreposição do real com o virtual e vice-versa.

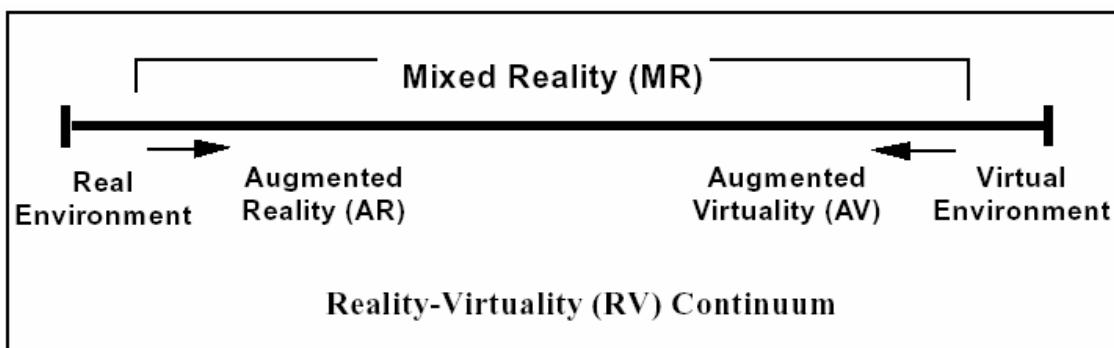


Figura 2.1 - Diagrama de realidade/virtualidade contínua (MILGRAM et al, 1994).

A Realidade Aumentada hoje é aplicada como ferramenta de visualização nas mais variadas áreas, como visualização médica, simulação, jogos, pode ser aplicada também à pesquisa e ainda à educação.

De acordo com o exposto, entende-se que as soluções de Realidade Aumentada envolvem a geração de elementos virtuais que são inseridos no ambiente real, de tal forma que o usuário crê que os mesmos são partes do meio no qual está inserido.

Assim, Cardoso et al. (2007) afirma que são características básicas de sistemas de Realidade Misturada:

- ✓ Processamento em tempo real.
- ✓ Combinação de elementos virtuais com o ambiente real.
- ✓ Uso de elementos virtuais concebidos em 3D.

Observa-se que a concepção de soluções de Realidade Aumentada necessita de componentes que permitam avaliar a posição de quem interage, o ponto de vista e gerar os elementos virtuais para, finalmente, combiná-los com o mundo real por meio de um sistema de projeção. Para tanto, os elementos reais e virtuais necessitam ser alinhados corretamente, um em relação ao outro (AZUMA, 1997).

#### **2.4.1 Aplicações Educacionais em Ambientes de Realidade Aumentada**

Os sistemas virtuais possibilitam experiências com a sensação de presença, por meio da integração dinâmica de diferentes modalidades perceptivas, que envolvem imagens, sons, tato, etc. Assim, torna-se possível a capacidade de manipular, relacionada às reações sensório-motora em tempo real (LÉVY, 1999). Ainda, tem-se que, por outro lado, os ambientes educativos devem oferecer condições favoráveis à criação, comportando-se como um espaço agradável e permitindo aplicações práticas e a relação do conhecimento com experiências, necessidades e realidade do aprendiz (usuário) (BILLINGHURST, 2005).

De maneira geral, a construção do conhecimento dá-se por meio da reflexão, da crítica, da identificação e da busca de resoluções dos problemas, propiciando situações que determinem o desafio-papel importante na formação de atitudes (VALENTE, 2001). Neste contexto os ambientes virtuais podem contribuir, estimulando a curiosidade e auxiliando no desenvolvimento da autonomia. Entende-se que a aprendizagem ocorre, quando o indivíduo está engajado e utiliza de forma consciente, estratégias de resolução de problemas para a construção significativa.

Entende-se a possibilidade de interação entre objetos reais e virtuais, que ocorre por meio da Realidade Aumentada (RA), pode oferecer ao usuário maiores informações sensitivas, facilitando a associação e a reflexão sobre a situação. Os sistemas de Realidade Aumentada permitem que o usuário decida sobre os ambientes, compondo cenas com imagens de objetos tridimensionais geradas por computador misturadas com imagens reais, aumentando as informações do cenário e oferecendo condições para a imersão no ambiente criado (BILLINGHURST, 2005). Desta forma, a principal característica destes ambientes é que as informações do mundo real são utilizadas para criar um cenário incrementado com elementos gerados por computador (DAINESE, 2003).

#### **2.4.2 Intervenções Educacionais com Realidade Aumentada**

Três características são responsáveis por tornar as situações de intervenção educacionais interessantes: *curiosidade, fantasia e desafio*. Por meio dos ambientes de

Realidade Aumentada, é possível proporcionar ao aprendiz (usuário) situações lúdicas, tornando as atividades mais motivadoras (KALAWSKY, 1993).

Em ambientes de Realidade Aumentada, o mundo real é “aumentado” com informações que não estão presentes na cena capturada, e o usuário passa ser um elemento participativo no cenário em que imagens reais são misturadas com virtuais para criar uma percepção aumentada (AZUMA, 2001).

A interface deve ser entendida como um espaço de comunicação, um sistema semiótico, onde signos são usados para interação, possibilitando o acesso ao ambiente (GARBIN et al., 2004). Assim, para garantir uma boa usabilidade, os fatores humanos devem ser respeitados. Isso remete à questão da diversidade dos usuários, suas características cognitivas, de personalidade, cultura, idade, comportamento, habilidades e necessidades especiais (BARANAUSKAS & ROCHA, 2003). Neste contexto, entende-se que os estudos sobre a memória humana (principalmente a de curta duração), vêm oferecendo subsídios para soluções inteligentes sobre a interface, cuja idéia central é liberar o usuário da memorização de comandos para tornar o ambiente mais agradável e natural possível, dentro das características das ferramentas de desenvolvimento, por meio de interfaces gráficas (PINHO, 1996). Entende-se que um ambiente educativo deve ser atrativo e, interessante, oferecendo, por meio de situações lúdicas e espontâneas, atividades que proporcionem o desenvolvimento cognitivo (LÉVY, 1999). Nesse contexto, de acordo com Azuma (1997), a Realidade Aumentada enriquece o ambiente real com objetos virtuais, com base em tecnologias que permitem misturar o cenário real com virtuais. Com esse fator motivacional, a Realidade Aumentada possibilita implementar ambientes que permitam a aplicação de estratégias diferenciadas para o desenvolvimento cognitivo. O sistema de Realidade Aumentada desta dissertação consistem de três etapas: a) desenvolvimento tecnológico, b) possíveis aplicações educacionais e) teste de usabilidade do software desenvolvido.

## 2.5. Sistemas Distribuídos de Realidade Virtual e Aumentada

As aplicações de Realidade Virtual e Aumentada podem ser vistas sob um aspecto bastante amplo, variando de uma única pessoa, usando um único computador, até muitos usuários, usando um sistema distribuído (RINALDI et al., 2006). Dentro deste contexto, os Ambientes Virtuais Distribuídos (AVD's) vêm crescendo e apresentando um elevado potencial de aplicação. Eles são caracterizados como Ambiente Virtual (AV) interativo em que usuários dispersos geograficamente têm como objetivos a cooperação e o compartilhamento dos recursos computacionais em tempo real usando um suporte de rede de

computadores para melhorar o desempenho coletivo por meio da troca de informações (RINALDI et al., 2006).

De acordo com Rinaldi et al. (2006), em AVD's os usuários podem compartilhar um mesmo espaço tridimensional virtual de trabalho (*workspace*), onde poderão se auxiliar na execução de uma determinada tarefa, baseando-se nos princípios de trabalho cooperativo baseado em computador (*CSCW - Computer Supported Cooperative Work*).

Ainda, de acordo com Ribeiro (2006), a Realidade Virtual pode empregar várias técnicas para reproduzir o mundo real e imaginário e possibilitar a manipulação e visualização de informações no computador como se fosse o mundo real. Entende-se que a complexidade desses ambientes virtuais aumenta na medida em que essas informações tornam-se comuns a uma série de usuários, ou seja, esses ambientes são distribuídos (RIBEIRO, 2006). Assim, pela sensação de compartilhamento de espaço, todos os participantes de um AVD têm a ilusão de estarem localizados no mesmo lugar, tais como na mesma sala, prédio ou região.

Baseando-se nesta constatação, Rinaldi et al. (2006) destaca que este espaço compartilhado representa um local comum, podendo ser real ou fictício. O local compartilhado deve apresentar as mesmas características a todos os participantes. Desta forma, entende-se que a comunicação dos Ambientes Virtuais pode ocorrer por meio de gestos, textos, eventos e áudio. Assim, de acordo com Rinaldi et al. (2006), um AVD consiste de quatro componentes básicos: *displays* gráficos, dispositivos de comunicação e controle, sistema de processamento e rede de comunicação. Estes componentes devem trabalhar juntos para fornecer a sensação de imersão em diferentes localidades.

De acordo com Rinaldi et al. (2006), de forma geral, os AVD's podem ser classificados em *centralizados* e *distribuídos*. No modelo centralizado, todos os usuários compartilham o ambiente virtual, enquanto no modelo distribuído, o ambiente virtual pode estar replicado ou particionado.

Assim, Rinaldi et al. (2006) define que “os ambientes replicados geralmente estão relacionados com AV's de pequeno porte”. E, ainda nesse caso, réplicas do AV são distribuídas para cada participante. Na presença de *n* usuários, quando um usuário fizer qualquer alteração no ambiente virtual a mesma deverá ser transmitida, para todas as outras (*n-1*) versões desse ambiente, ou seja, todos os usuários têm uma cópia completa do ambiente (SINGHAL & ZYDA, 1999).

Os modelos de comunicação em um AVD podem ser baseados no modelo Cliente/Servidor, com computação distribuída orientada a objetos. Neste sentido, Rinaldi et al. (2006) destaca que o paradigma Cliente/Servidor em um AVD implica na comunicação direta

somente do cliente com o servidor ou vice-versa, não havendo a troca de mensagens entre os clientes. Assim, entende-se que existem diversas plataformas que proporcionam essa distribuição como CORBA, RPC, Java/RMI (RAJ, 2004).

Entende-se que um Sistema Distribuído deve possuir uma estrutura básica de comunicação: *unicast*, *broadcast*, *multicast* (MACEDONIA & ZYDA, 2005). Assim, na estrutura *unicast*, a comunicação dos dados é conhecida como “um-para-um”, sendo realizada somente entre dois *hosts* (RINALDI et al., 2006). Já na estrutura *broadcast* (difusão) é o mecanismo de comunicação dos dados conhecido como “um-para-vários”, em que um *host* envia um dado para todos os outros *hosts* da rede, que podem ignorá-la, se a mensagem não for de seu interesse. Ambientes Virtuais replicados se enquadram neste mecanismo (RINALDI et al., 2006). E por fim, o *multicast*, por sua vez, é o mecanismo de comunicação dos dados conhecido como “um-para-vários” ou “vários-para-vários” em uma única operação, onde os dados são enviados para um grupo de máquinas simultaneamente. (RINALDI et al., 2006).

Resumidamente, um AVD (Ambiente Virtual Distribuído) pode ser definido de forma simplificada como um sistema que permite vários usuários interagirem tanto com o ambiente, quanto entre “eles” em tempo real, mesmo que estes estejam em diferentes localidades geográficas (RIBEIRO, 2006).

Então, entende-se que, como qualquer outro Ambiente Virtual, a criação de Ambientes Virtuais distribuídos está bastante ligado com realismo visual e interação. Rinaldi et al. (2006) destaca que assuntos como modelagem gráfica tridimensional e interação homem-máquina constituem uma parte fundamental. Em especial na elaboração de um AVD destaca-se o suporte de comunicação em rede. Por meio de uma rede é fornecida a base para que unidades computacionais heterogêneas separadas fisicamente sejam unificadas para implementar um único ambiente virtual.

Outro ponto importante é o comportamento de um objeto que, depende principalmente do nível de interação que o usuário pode ter. Objetos que o usuário pode somente observar têm um nível de complexidade menor que aquele com o qual o usuário pode interagir, modificar e ajustar. Por fim, Oliveira (2000) define que o padrão de comportamento, entretanto, depende da finalidade do mundo virtual. Assim, de acordo com Rinaldi et al. (2006) um outro fator de importância em um AVD são os protocolos de comunicação.

Desta forma, pode-se destacar os modelos de suporte para comunicação, onde Alburquerque (2001) destaca que são vários os modelos de suporte de comunicação que podem ser utilizados. Assim, Rinaldi et al. (2006) destaca que geralmente o desenvolvedor

procura um ambiente que lhe abstraia detalhes de comunicação. São vários os pacotes de software que oferecem tais serviços. Pode-se dividi-los em quatro categorias: *sockets*, *toolkits*, *middlewares e frameworks* (RICCIONI, 2000).

Riccioli (2000) relata que “os middlewares são soluções independentes de plataformas e de domínio de aplicação. Como exemplo pode-se citar o modelo CORBA<sup>1</sup>, JAVA/RMI<sup>2</sup> além de outros”.

Ribeiro (2006) destaca que a área de aplicação dos AVD’s é muito abrangente: treinamento (pilotos, militar), pesquisa, ensino à distância, comércio, cultura e engenharia. Desta forma, entende-se que o computador apresenta um grande potencial como ferramenta de apoio ao ensino, que aliada às técnicas de Realidade Virtual e colaboração, pode enriquecer e valorizar a informação transmitida, simulando realidades, muitas vezes, fora do alcance dos usuários (RIBEIRO, 2006).

### **2.5.1. Comparação das plataformas CORBA, DCOM e JAVA/RMI**

De acordo com Costa (2000), o CORBA em relação ao DCOM é uma plataforma completa e bem definida, assim como, mais organizada em termos de documentação. O CORBA por ser mais abrangente é sem dúvida mais complexa. Produtos baseados nas especificações CORBA, utilizando a IDL<sup>3</sup> CORBA e serviços CORBA, são oferecidos por várias empresas e em muitas plataformas, inclusive Windows, mas a desvantagem é que não existe um esquema de verificação de conformidade das implementações destes produtos com as suas especificações. Também, o fato de existirem várias empresas fornecendo famílias de produtos baseados nas especificações CORBA, faz com que não se somem esforços nos desenvolvimentos desses produtos o que facilitaria a “interoperabilidade” (COSTA, 2000).

Ainda de acordo com Costa (2000) em relação à possibilidade de distribuição de objetos por meio do Java/RMI tem-se a comentar que, o RMI é uma forma de distribuição exclusiva da linguagem Java. O Java também não define uma IDL que possa ser utilizada em conjunto com outras linguagens de programação, pois as interfaces dos objetos Java são descritas com “statements” Java. Para que o Java/RMI possa ser utilizado em ambientes distribuídos em larga escala é necessário que ele suporte serviços tais como os serviços CORBA. Assim, como um passo para que o Java venha a ser realmente um ambiente de distribuição completo, a Sun Microsystems desenvolveu a especificação “Enterprise JavaBeans” (EJB) que define o serviço de transação baseado em Java. Nesta especificação o

---

<sup>1</sup> *Common Object Request Broker Architecture*

<sup>2</sup> *Java/Remote Method Invocation*

<sup>3</sup> *Interface Definition Language*

RMI é o protocolo nativo e passa a ser denominado “Java Remote Method Protocol” (JRMP). A partir deste e outros serviços a serem desenvolvidos o ambiente de distribuição baseado em Java poderá vir a ser uma importante opção de distribuição (COSTA, 2000).

Para o desenvolvimento da camada de distribuição desta dissertação utilizou-se o CORBA na sua versão 2.0 e, a implementação Visibroker 4.1 da Borland. Entende-se que as pesquisas que provêem suporte para programação distribuída em tempo real são recentes, e que ainda se faz necessário criar adaptabilidade de padrões convencionais para este fim. No entanto, CORBA 2.2 traz a possibilidade de implementação de aplicações em tempo real.

### 2.5.2 Escolha da Arquitetura

De acordo com a comparação no item anterior e outros estudos comparativos, não existem grandes diferenças funcionais entre as plataformas estudadas. Em um estudo comparativo, Siqueira (2005) ratifica a pequena diferença entre plataformas de distribuição, como mostra o gráfico da Figura 2.2.

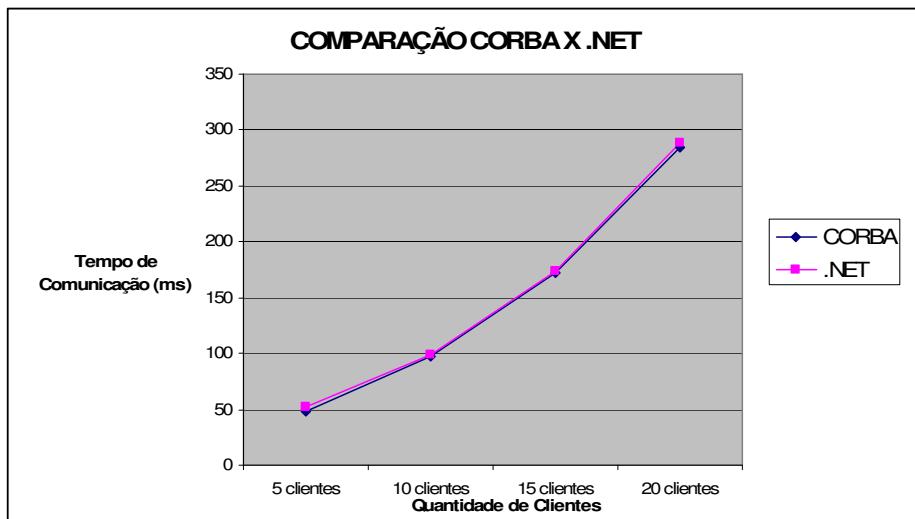


Figura 2.2. Latência dos protótipos (SIQUEIRA, 2005)

Assim, Siqueira (2005) comparou o desempenho de dois protótipos, sendo cada um implementado em uma plataforma de distribuição, mostrando a proximidade dos resultados obtidos. Siqueira (2005) sugere, porém que, com o aumento de clientes (escalabilidade), a arquitetura CORBA (OMG, 2008) se porte melhor. Entre CORBA, JAVA/RMI e DCOM/.NET, além da análise de outro estudo, onde Ribeiro (2006) relata que CORBA foi escolhida pelos motivos a seguir:

- a) É independente de plataforma.
- b) Pode ser implementada na maioria das linguagens existentes.

c) Possui uma implementação menos complexa e com quantidade de linhas de código também inferior às demais.

d) Possui várias implementações, com destaque ao *Visibroker*, que foi usado no desenvolvimento do protótipo desta dissertação e, por estar presente em vários ambientes de programação.

Apesar de ser uma comparação simples, a Tabela 2.1 mostra as principais diferenças entre as plataformas estudadas, ressaltando a escolha por CORBA.

**Tabela 2.1. Comparação entre plataformas de distribuição (RIBEIRO, 2006).**

CARACTERÍSTICAS	PLATAFORMAS		
	CORBA	DCOM	JAVA/RMI
Independência de plataforma	<b>SIM</b>	<b>NÃO</b>	<b>SIM</b>
Implementação em diversas linguagens	<b>SIM</b>	<b>SIM</b>	<b>NÃO</b>
Implementações diferentes	<b>SIM</b>	<b>NÃO</b>	<b>NÃO</b>

Dante do exposto, foi escolhido a plataforma CORBA 2.0 e a implementação Visibroker 4.1.

## 2.6 O que é Distribuição?

Segundo Sementille (1999), um sistema distribuído é uma coleção de computadores independentes que aparecem para os usuários do sistema como um único computador.

Outra definição seria um sistema em que componentes de *hardware* e *software* localizados em computadores de uma rede se comunicam por meio de mensagens (ALBURQUERQUE, 2001). Portanto, um Sistema Distribuído é aquele que “roda” em um conjunto de máquinas sem memória compartilhada, aparecendo como um único computador para seus usuários.

### 2.6.1. CORBA

CORBA é a especificação de uma arquitetura para objetos distribuídos e heterogêneos. O padrão CORBA é uma solução aberta de objetos distribuídos desenvolvido pela OMG (*Object Management Group*) para se tornar um padrão no mercado. Recebe destaque por ser independente de linguagem e fabricante, possibilitando que objetos de sistemas distribuídos troquem mensagens entre si de forma transparente, não importando onde eles estejam, em que plataforma ou sistema operacional estejam rodando, em que linguagem de programação foram implementados e até mesmo qual protocolo de comunicação utilizam

(RIBEIRO, 2006). A arquitetura definida para CORBA possui um alto nível de abstração, permitindo que a implementação do cliente e do servidor possa ser feita em qualquer linguagem e que os objetos se comuniquem de forma totalmente transparente por meio de um “barramento de *software*”. Isso só é possível porque os objetos têm suas interfaces descritas em uma linguagem padrão, chamada de *Interface Definition Language* (IDL). A função da IDL é descrever as interfaces das implementações de objetos, que são acessadas por seus clientes.

Foi definida pela OMG uma arquitetura denominada OMA (*Object Management Architecture*) para realizar a integração entre aplicações. Enquanto CORBA permite a interoperabilidade entre objetos, a OMA agrupa um conjunto de objetos CORBA em serviços e facilidades, que oferecem suporte para o desenvolvimento de aplicações que usam objetos CORBA (RIBEIRO, 2006).

Tudo na arquitetura CORBA depende de um *Object Request Broker* (ORB). O ORB atua como um *Object Bus* central sobre cada objeto CORBA, interagindo transparentemente com outros objetos localizados no mesmo computador ou remotamente. Cada objeto servidor CORBA tem uma interface e expõe um grupo de métodos. Para solicitar um serviço, um cliente adquire uma referência de objeto para o objeto servidor CORBA. O cliente pode fazer pedidos de métodos na referência de objeto como se o objeto servidor residisse no espaço de endereço do cliente. O ORB é responsável por achar a implementação de objetos CORBA, preparando-a para receber e enviar pedidos e carregar a resposta de volta ao cliente. Uma vez que CORBA é somente uma especificação, pode ser usada em diversas plataformas de sistemas operacionais de *mainframes* a UNIX, de máquinas Windows a aparelhos *handheld*, desde que haja uma implementação ORB para aquela plataforma (OMG, 2008).

### **2.6.2. ORB – Object Request Broker**

Um ORB é um mecanismo básico pelo qual objetos transparentemente fazem requisições para – e recebem respostas de – cada outro objeto na mesma máquina ou por meio de uma rede. O cliente não precisa estar ciente do mecanismo usado para comunicar ou ativar um objeto, ou como o objeto é implementado, nem onde o objeto está localizado. Assim, o ORB forma a base para a construção de aplicações com objetos distribuídos e para a interoperabilidade entre aplicações em ambientes homogêneos e heterogêneos. (RIBEIRO, 2006).

O ORB é responsável pela localização de um objeto ao qual se destina uma requisição, assim como o encaminhamento dos parâmetros dessa requisição em um formato

que este objeto aceite. Se houver parâmetros de saída da requisição para o cliente, o retorno de tais parâmetros também é função do ORB.

Um objeto CORBA interage de várias maneiras: com o ORB; por meio da interface ORB ou por um *Object Adapter* (Adaptador de Objeto)<sup>4</sup> – ou um *Basic Object Adapter* (BOA); por um *Portable Object Adapter* (POA).

A Figura 2.3 mostra uma requisição de um cliente, enviada através do ORB, a uma implementação de objeto. O cliente é qualquer objeto que solicita um serviço e a implementação do objeto contém o código e os dados que caracterizam o comportamento de um objeto. O ORB é responsável por todos os mecanismos necessários para achar a implementação de objeto de um pedido, por preparar a implementação de objeto para receber um pedido e pela comunicação.

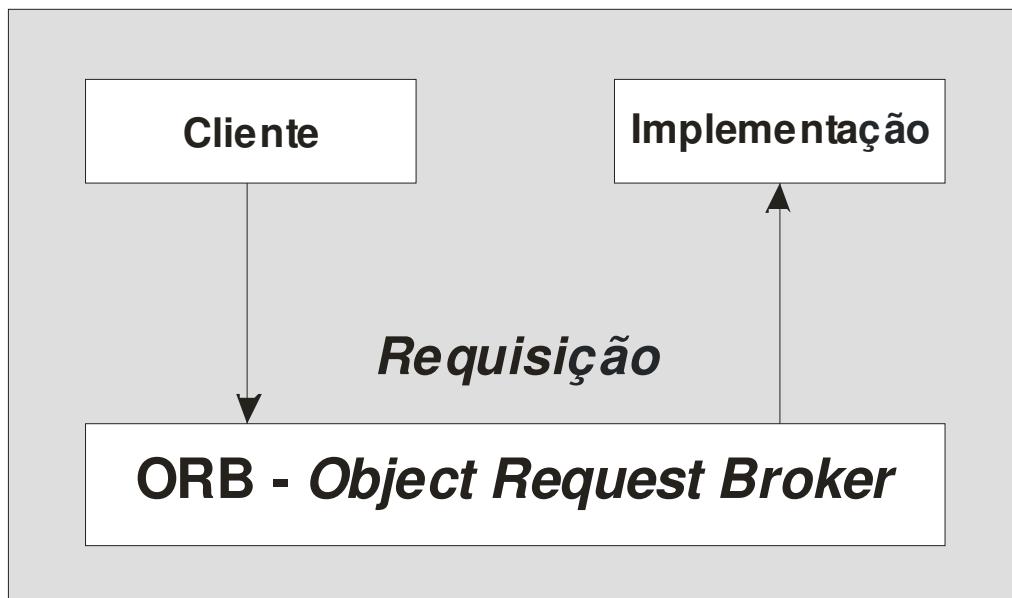


Figura 2.3. Requisição por meio do ORB.

A interface, com a qual o cliente tem contato, é totalmente transparente e independente de qualquer fator relacionado à heterogeneidade do ambiente distribuído no qual se encontra, não importando onde o objeto está localizado, em qual linguagem de programação foi implementado ou a ordenação de bytes da máquina na qual se executa o objeto. Assim, qualquer software que ofereça as interfaces e os serviços especificados pode ser considerado um ORB.

Entretanto, segundo Riccioni (2000), ORBs diferentes podem apresentar características de implementação diferentes, resultando em serviços prestados a objetos e clientes com qualidades e propriedades diferentes.

<sup>4</sup> Um adaptador de objetos possibilita a comunicação dinâmica de objetos, mesmo quando o objeto não foi implementado no mesmo momento da construção da aplicação.

## 2.7 Considerações Finais

Os conceitos vistos neste capítulo são essenciais para o bom entendimento da Realidade Virtual e sua integração com as tecnologias de distribuição. O objetivo deste capítulo não é ensinar Ambientes Virtuais Distribuídos, Realidade Virtual ou Realidade Aumentada, mais sim, dar entendimento para que o pesquisador/leitor entenda os conceitos de Realidade Virtual e Aumentada, e ainda de Ambientes Virtuais Distribuídos utilizados no desenvolvimento do protótipo apresentado por meio desta dissertação. Desta forma, no próximo capítulo apresenta-se o estado da arte desta pesquisa por meio dos trabalhos relacionados.

# CAPÍTULO III

## 3. TRABALHOS RELACIONADOS

### 3.1 Introdução

Com o intuito de buscar novas tecnologias que auxiliem na criação do sistema proposto, foi realizado um estudo do estado da arte em relação à distribuição de ambientes de Realidade Virtual e Aumentada e, ainda, sistemas com interface de Realidade Aumentada que possuam características de interação e manipulação de objetos em tempo real. Assim, este capítulo tem como objetivo oferecer uma visão dos principais trabalhos encontrados na literatura, expondo suas características mais importantes.

#### 3.1.1 Metodologia para a análise

Dentre os trabalhos analisados, verificou-se os que utilizam o ARToolKit<sup>5</sup> (*pacote de ferramentas para desenvolvimento de ambientes de Realidade Aumentada*) ou similar, buscando verificar a existência de alguma interface que proporcione interações com os objetos virtuais em tempo real e, que faça a distribuição em uma rede de computadores destas interações realizadas e, ainda, que tenha uma “*Camada*” que possa ser re-utilizada, com muita facilidade, em outras aplicações de Realidade Aumentada, independentemente de linguagem de programação ou de arquitetura de projeto e, por fim, buscou-se verificar as características que pudessem ser utilizadas como apoio pedagógico, ou seja, software com características educacionais. Para a construção deste capítulo verificou-se trabalhos em três vertentes: *Interfaces de Realidade Aumentada, Realidade Virtual Distribuída e Realidade Aumentada Distribuída*. Assim, neste aspecto buscou-se como metodologia analisar os seguintes itens: a)

Quanto à distribuição de Ambientes Virtuais em Realidade Aumentada:

- Arquitetura de distribuição.
- Modelo de distribuição.
- Replicação.
- Distribuição.
- Camada de Comunicação.

---

<sup>5</sup> Descrito por Kato e Billinghurst (2005), o ARToolKit utiliza-se de técnicas de visão computacional para calcular o ponto de vista da câmera em relação a um marcador existente no mundo real.

b) Quanto às interações em Interfaces de Realidade Aumentada:

- Existência de colisão.
- Marcadores com interação.
- Realização de transformações geométricas em tempo real de execução.
- Uso do teclado e mouse.
- Troca de objetos na cena em tempo real de execução.
- Uso de menus.

E por fim, verifica-se o potencial do software para a utilização em sala de aula.

## 3.2 Interfaces de Realidade Aumentada

### 3.2.1 Interactive Entertainment Systems Using Tangible Cubes

Desenvolvido pelo *Laboratório de Tecnologia em Interface Humana do Departamento de Engenharia da Computação da Universidade Nacional de Cingapura*, os autores (ZHOU et al., 2004) relatam um sistema de entretenimento baseado em cubos interativos e tangíveis. O software possui fortes características educacionais, pois trabalha como se fosse um contador de histórias, sua primeira versão, basicamente é uma extensão do MagicBook (BILLINGHURST; KATO & POUPYREV, 2001), porém é acrescido de interações por meio de uma “pá” (em formato de um marcador) que permite a troca de objetos virtuais de um marcador para outro em uma cena e, ainda possui efeitos de sons. A Figura 3.1 ilustra o funcionamento do aplicativo e suas interações por meio do marcador “pá”.

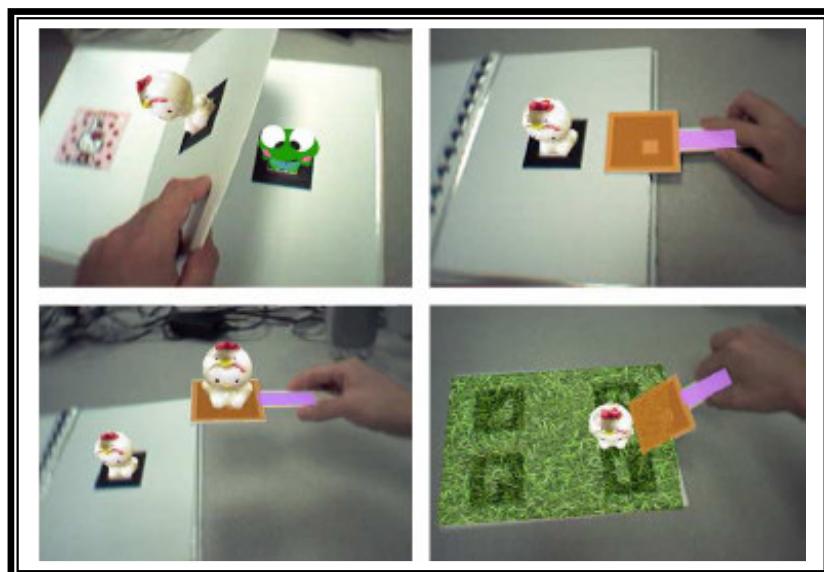


Figura 3.1. Interação de ambiente com marcador “pá” (ZHOU et al., 2004)

A segunda versão do aplicativo é bem parecida com a primeira, porém é acrescido de algumas peculiaridades: o aplicativo é em formato de um cubo (cada lado do cubo possui um marcador relacionado a um cenário de realidade aumentada), o aplicativo possui, além dos sons, o acréscimo de vídeos. Assim, de acordo com manipulação do cubo os ambientes virtuais vão trocando os cenários. A Figura 3.2 ilustra a utilização do cubo mágico.

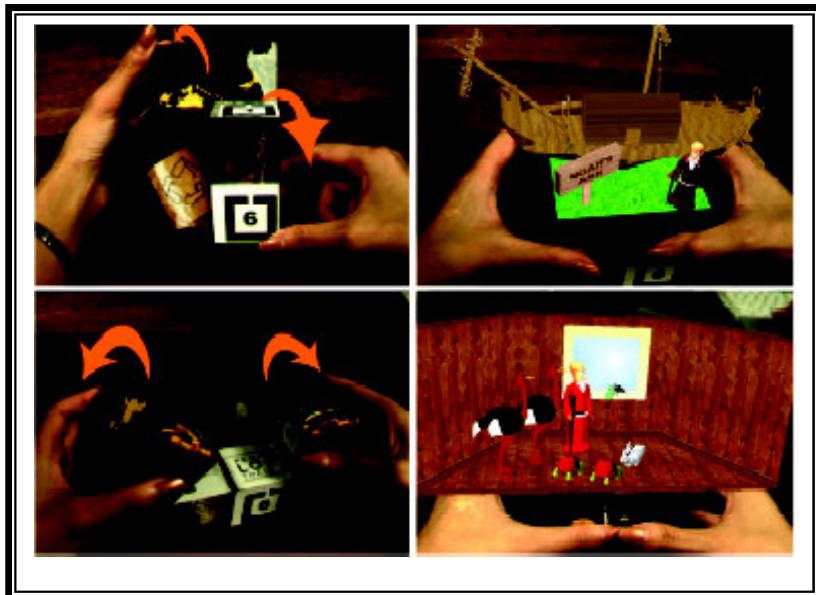


Figura 3.2. Magic History Cube em ação (história bíblica) (ZHOU et al., 2004)

Desta forma, a aplicação possui vários cenários modelados em formato de um jogo 3D onde, vídeo clipes, sons, imagens, modelos 2D e 3D são aplicados para proporcionar ao usuário uma experiência multisensorial em um ambiente de entretenimento interativo.

### 3.2.2 Visualização de dados relativos a redes de computadores usando Realidade Virtual e Aumentada

Neste trabalho desenvolvido pelo *Grupo de Realidade Virtual da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia (UFU)*, os autores (ZORZAL et al., 2007) relatam uma aplicação para visualização de informações. O artigo relata a implementação de duas aplicações: VR TraceRoute, que utiliza como base de seu funcionamento o comando “tracert”, interpretando os dados informados e inserindo-os em uma interface de visualização mais intuitiva. Assim, o VR TraceRoute apresenta os dados em um ambiente virtual, e o usuário pode interagir com os mesmos, por meio de dispositivos tecnológicos, tais como teclado e mouse. Após a inserção do endereço de destino a ser traçado, e após calculado o trajeto, o usuário pode fazer um percurso pelo caminho por onde o pacote de dados trafegou. Para isto foi adicionado um sensor no computador situado no ambiente virtual, o qual representa o endereço de origem do usuário. Assim, para que o

percurso seja realizado, basta clicar com o botão esquerdo do mouse sobre ele. A Figura 3.3 ilustra o ambiente do VR TraceRoute.

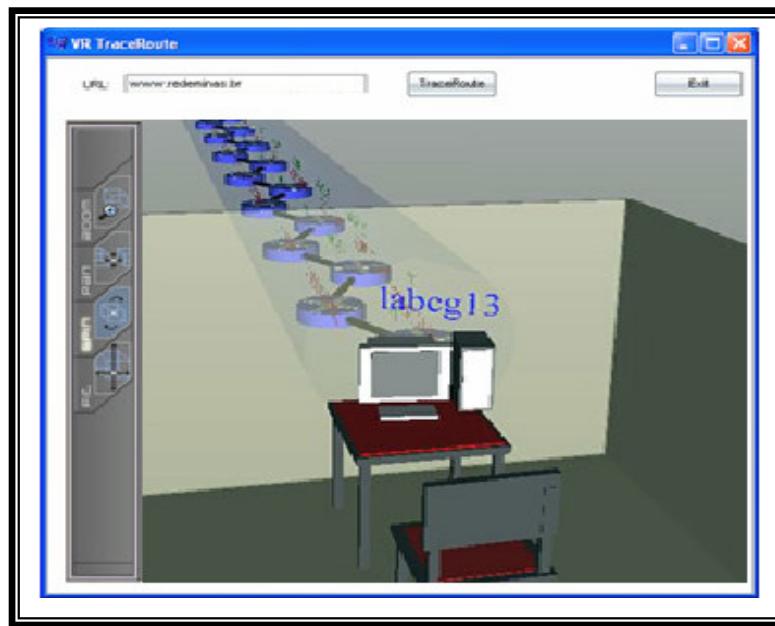


Figura 3.3. VR TraceRoute (ZORZAL et al., 2007)

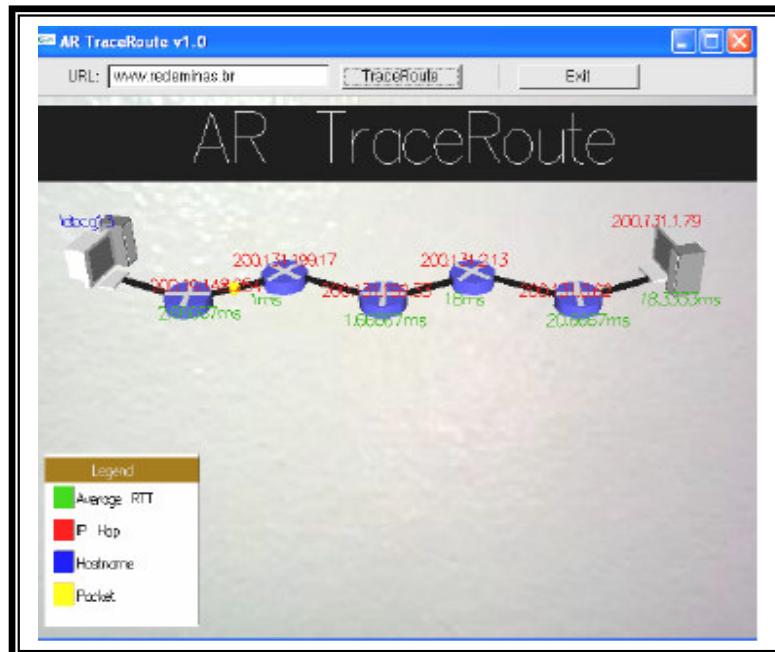


Figura 3.4. AR TraceRoute (ZORZAL et al., 2007)

A segunda parte do artigo, apresentado no *IX Symposium on Virtual and Augmented Reality*, traz a implementação de um sistema de visualização de informações parecido com o sistema demonstrado anteriormente, que na verdade é a extensão para um ambiente de Realidade Aumentada do VR TraceRoute. O sistema AR TraceRoute também utiliza como base de seu funcionamento o comando “tracert” do sistema operacional Windows para interpretar os dados adquiridos e, inseri-los em uma interface de Realidade Aumentada.

A forma com que esses dados são processados é similar àquela utilizada pelo sistema VR TraceRoute; o que muda entre essas duas aplicações, no entanto, é o modo de interação dos usuários com os objetos virtuais e o ambiente em que esses objetos são apresentados.

Para que o sistema suportasse a interface de Realidade Aumentada, foi necessário encontrar um software que permitisse fácil customização às necessidades do sistema. Dentre várias possibilidades, foi escolhido o ARToolKit, que é um conjunto de ferramentas e bibliotecas para Realidade Aumentada em código aberto, permitindo alterações nos exemplos e desenvolvimento de novas funções. Além disso, ele oferece suporte a arquivos gerados em VRML, possibilitando a fácil integração com o módulo que gera os ambientes virtuais. A Figura 3.4 ilustra o ambiente da ferramenta AR TraceRoute.

### 3.3 Realidade Virtual Distribuída

#### 3.3.1 Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares

Neste trabalho desenvolvido pelo *Grupo de Realidade Virtual da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia (UFU)*, o autor (RIBEIRO, 2006) relata em sua tese, uma arquitetura para distribuição de ambientes virtuais multidisciplinares de ensino.

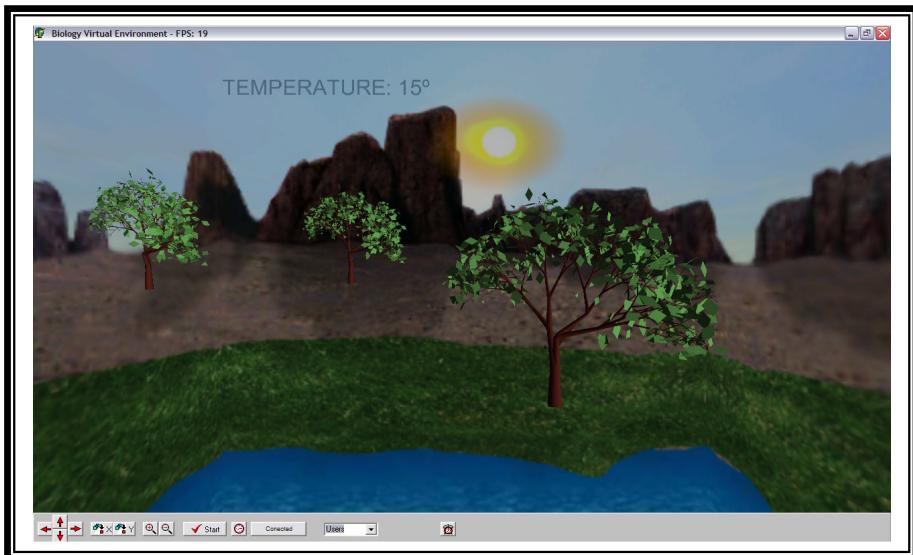


Figura 3.5. Ambiente Virtual de Biologia (RIBEIRO, 2006)

Para tanto, diferentes arquiteturas de distribuição foram avaliadas com o objetivo de identificar aquela que com mais eficiência permita que interações ocorridas em um ambiente alterem o comportamento de outros, mesmo que estes sejam relacionados a outras áreas do conhecimento. Protótipos construídos sobre a plataforma escolhida para a distribuição, seguindo uma mesma metodologia (onde aspectos do modelo de dados foram alterados) e

ainda, tendo a latência, escalabilidade e extensibilidade como parâmetros de comparação demonstraram qual a melhor abordagem para construção de ambientes virtuais multidisciplinares. Cada protótipo foi construído com base em algoritmos de distribuição que permitiram ao sistema funcionar corretamente em situações passíveis de erros. Ambientes Virtuais de Biologia (paisagem com plantas, água, luz e terra) e Química (membrana de uma folha) foram utilizados tendo o fenômeno da fotossíntese como estudo de caso e relação entre os dois ambientes. O sistema foi avaliado por professores e alunos e os resultados alcançados permitiram concluir que o mesmo é eficaz e aplicável. Observe as Figuras 3.5 e 3.6, onde se ilustra os ambientes virtuais de biologia e química respectivamente.

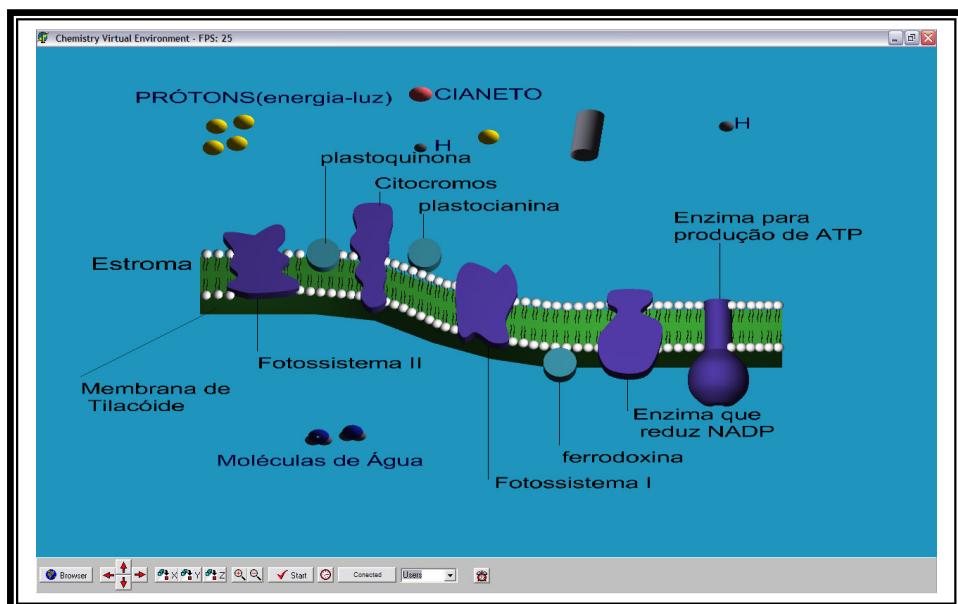
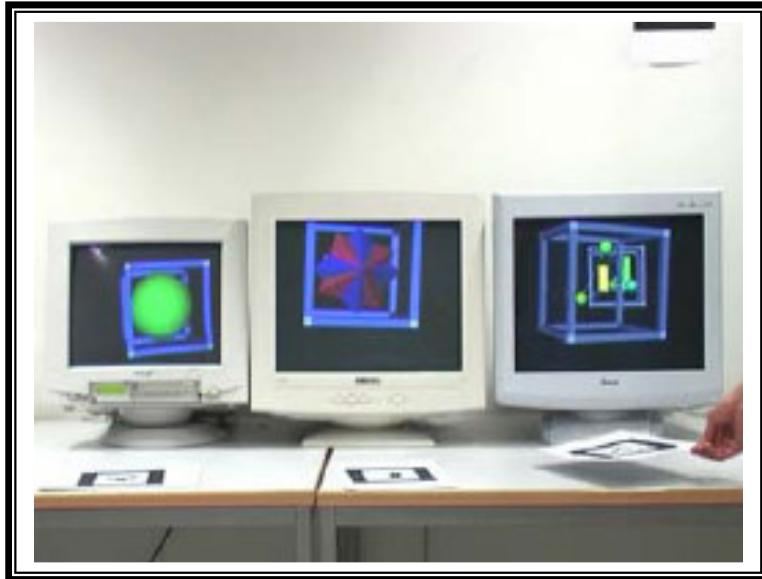


Figura 3.6. Ambiente Virtual de Química (RIBEIRO, 2006)

### 3.4 Realidade Aumentada Distribuída

#### 3.4.1 Distributed Applications for Collaborative Augmented Reality

Desenvolvido pelo *Grupo de Sistema de Mídia Interativa da Universidade de Tecnologia de Viena* (HESINA & SCHMALSTIEG, 2005), este software de Realidade Aumentada apresenta uma arquitetura para distribuição e colaboração de um espaço de trabalho tri-dimensional. Assim, o sistema permite que vários usuários possam compartilhar uma experiência qualquer em 3D, que por sua vez conta com várias aplicações, montando uma espécie de quebra-cabeças que é agrupado por vários monitores e outros sistemas de projeção, tais como capacetes e projeção em Realidade Aumentada, observe a Figura 3.7.



**Figura 3.7. Interface de colaboração e projeção em monitores da aplicação (HESINA & SCHMALSTIEG, 2005).**

O software tridimensional foi construído por meio do *toolkit* OpenInventor<sup>6</sup>. Assim, essa biblioteca de alto nível gráfico 3D trabalha com conceitos de orientação a objetos.



**Figura 3.8. Interface de colaboração e projeção em Realidade Aumentada (HESINA & SCHMALSTIEG, 2005).**

A distribuição, é feita por meio de facilidades de distribuição contida no *toolkit* OpenInventor, que por sua vez trabalha com o compartilhamento de cenas gráficas usando a semântica de memória compartilhada. Assim, destaca-se que a distribuição funciona por meio de um mecanismo que mantém múltiplas réplicas de uma cena sincronizada. É válido ressaltar que este mecanismo é do OpenInventor, assim a programação é baseada em uma notificação

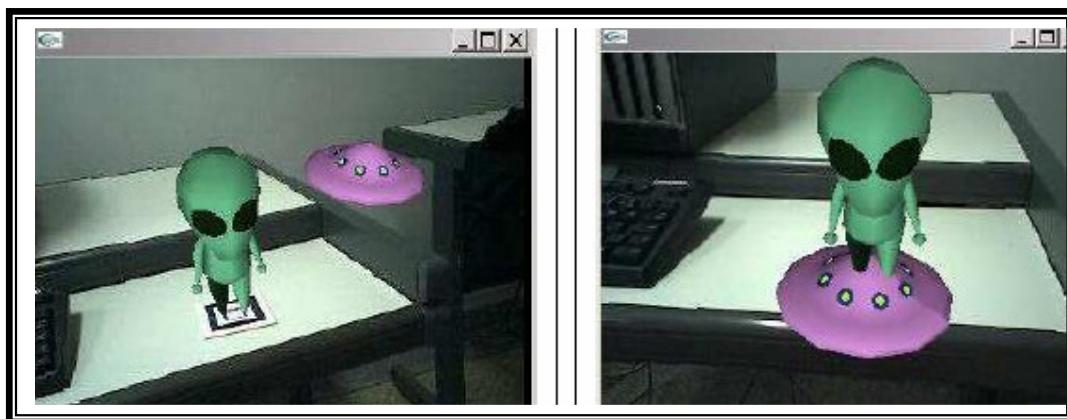
---

<sup>6</sup> <http://oss.sgi.com/projects/inventor>

automática para desencadear “alterações” no observador sempre que um pedido de mudança for realizado na cena, em seguida, todas as alterações são replicadas e propagadas para todas as réplicas da cena gráfica. Desta forma, ressalta-se que este processo, segundo os autores, é multicast. A Figura 3.8 ilustra a colaboração da aplicação em ambientes de Realidade Aumentada.

### 3.4.2 Ambientes Colaborativos com Realidade Aumentada

Neste trabalho desenvolvido na Universidade Metodista de Piracicaba, os autores (KIRNER & GERALDI, 2005) descrevem o desenvolvimento de um trabalho bastante interessante. Baseado no uso do software ARToolKit, configurado para funcionar em rede por meio do uso de soquetes. Para isto, os autores relatam que o ARToolKit foi modificado para importar e exportar posições, permitindo a colocação de objetos virtuais em posições recebidas e o envio das posições das placas marcadoras existentes no ambiente local. Desta forma, os autores destacam que inicialmente, os objetos virtuais que foram compartilhados, são cadastrados em todos os computadores de uma rede, enquanto cada local terá cadastrado suas placas marcadoras associadas aos objetos virtuais. Assim, quando um usuário insere sua placa marcadora no campo de visão da “webcam”, o objeto virtual associado aparece sobre a placa marcadora, de forma que ao mover-se a placa, o objeto move-se junto. Como a posição e a identificação da placa são exportadas para outros computadores. Assim, todos os computadores remotos recebem as informações e adicionam o mesmo objeto virtual na mesma posição, permitindo que os usuários enxerguem e manipulem seus objetos no mesmo ambiente, observe a Figura 3.9, os objetos compartilhando um ambiente.



**Figura 3.9. Visualização de objetos virtuais em RA compartilhando um ambiente**  
**(KIRNER & GERALDI, 2005)**

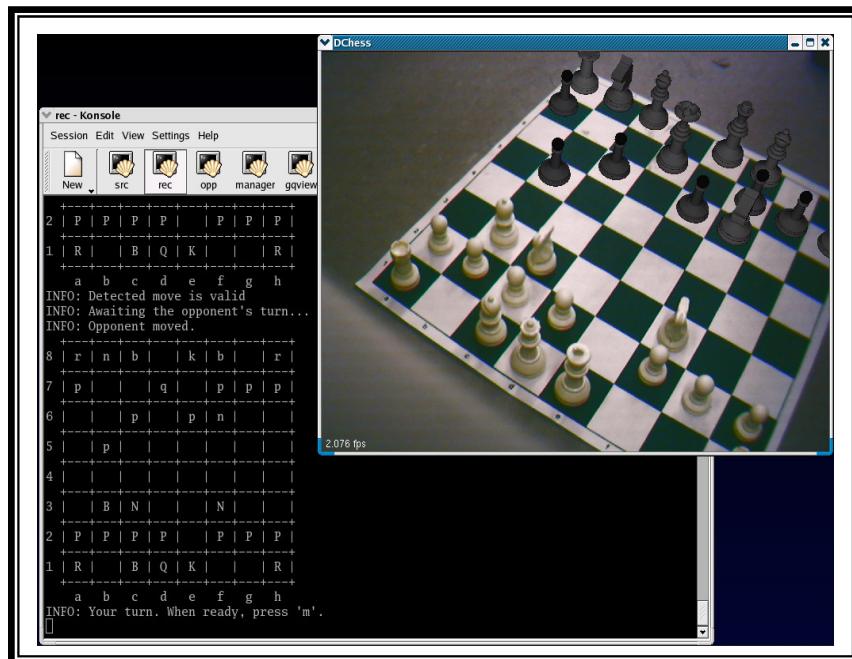
Com esta pesquisa e implementação, os autores concluíram que a utilização da Realidade Aumentada beneficia de maneira muito significativa a percepção dos usuários e sua interação, principalmente em ambientes colaborativos.

### 3.4.3 Distributed Augmented Chess System

Neste trabalho desenvolvido na Pontifícia Universidade Católica do Rio de Janeiro (PUC - Rio), os autores (RAPOSO et al., 2003) desenvolveram um sistema em Realidade Aumentada cujo objetivo foi desenvolver um jogo de xadrez em que: a) o jogador possa utilizar uma câmera de captura digital para filmar um tabuleiro real, sendo reconhecidas as jogadas feitas, b) as peças do adversário, seja ele um computador ou um humano, são desenhadas em cima da imagem capturada, c) o adversário pode ser remoto, com ou sem este sistema de captura, seja em uma rede local ou na internet. A arquitetura do sistema viabiliza servidores de jogos de xadrez, suportando espectadores de jogos, campeonatos etc.

Para a parte de colaboração entre os jogadores, os autores desenvolveram um protótipo de gerenciador de jogo e foi inserido o suporte à colaboração no programa de reconhecimento de jogadas e desenho das peças virtuais.

Desta forma, foi utilizada a tecnologia CORBA para a definição das interfaces de comunicação entre os dois módulos, tendo a ferramenta LuaCorba para a simplificação do uso de CORBA no projeto. A Figura 3.10 ilustra o projeto em fase final de implementação, com as peças reais e as peças virtuais.



**Figura 3.10. Jogo em colaboração em uma rede (RAPOSO et al., 2003)**

O algoritmo de reconhecimento se mostrou robusto, sendo capaz de reconhecer o tabuleiro em situações não triviais, conforme Figura 3.10. Certamente a sua grande limitação é a sua complexidade computacional. Alguns algoritmos podem ajudar a otimizar a busca, viabilizando o seu uso em tempo real. Desta forma, entende-se que o nível de interação do

software é baixo e, que o foco maior do trabalho baseou-se nas implementações dos algoritmos de detecção de linhas, jogadas e o processamento para gerar as jogadas feitas por um adversário qualquer.

A interação do software se dá por meio de mudanças das peças do tabuleiro, não havendo interações diretas nos objetos virtuais e, ainda o trabalho não possui foco educacional, estando voltado apenas para a aplicação da tecnologia.

### **3.5 Considerações Finais**

Analizando os trabalhos relacionados, pode-se perceber o uso do ARToolKit em muitas aplicações de Realidade Aumentada. Destaca-se também que há poucos softwares com interfaces de Realidade Aumentada que trabalhem em um ambiente distribuído. Desta forma, as principais características dos softwares descritos nos trabalhos avaliados são sintetizadas nas Tabelas 3.1, 3.2 e 3.3 a seguir.

Assim, as tabelas de comparação estão baseadas nos parâmetros definidos para avaliação desta aplicação em relação aos trabalhos avaliados, sendo divididas em três, onde a Tabela 3.1 ilustra apenas a comparação da interface de Realidade Aumentada, tendo como foco principal a interação dos objetos virtuais (essa interação pode ser feita de várias maneiras). Já a Tabela 3.2 ilustra apenas a comparação dos ambientes virtuais distribuídos, onde são avaliadas as questões de plataforma de distribuição e a possibilidade do sistema em se adaptar a ambientes de Realidade Aumentada e, por fim é apresentada a Tabela 3.3 que por sua vez ilustra os ambientes de Realidade Aumentada Distribuídos, neste item, destaca-se a comparação dos itens da Tabela 3.1 e também da Tabela 3.2.

Por meio deste aspecto técnico, entende-se que assim, se pode ter uma visão global do estado da arte dos sistemas de Realidade Aumentada Distribuídos.

Tabela 3.1. Interfaces de Realidade Aumentada - Características avaliadas.

Trabalhos Avaliados	Interfaces de Realidade Aumentada							Aplicabilidade na Educação	
	Interações em tempo de execução por meio de:								
	Colisão	Marcadores	Troca de Objetos	Hardware		Menus de Intereração			
Interactive Entertainment Systems Using Tangible Cubes	★	★	★	✗	✗	✗	✗	★	
Visualização de Dados Relativos a Redes de Computadores usando Realidade Virtual e Aumentada	✗	★	✗	★	★	✗	✗	✗	

Tabela 3.2. Realidade Virtual Distribuída - Características avaliadas.

Trabalhos Avaliados	Realidade Virtual Distribuída					Aplicabilidade na Educação	Utilização na R. A. sem esforço computacional		
	Distribuição realizada por meio de:			Modelo Cliente/Servidor	Aplicabilidade na Educação				
	Arquitetura CORBA	Camada de Comunicação	Modelo Cliente/Servidor						
Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares	★	✗	★	★	★	★	✗		

**Tabela 3.3. Realidade Aumentada Distribuída - Características avaliadas.**

Realidade Aumentada Distribuída											
Trabalhos Avaliados	Interações em tempo de execução por meio de:						Distribuição realizada por meio de:			Aplicabilidade na Educação	
	Colisão	Marcadores	Troca de Objetos	Teclado	Mouse	Menus de Interação	Arquitetura CORBA	Camada de Comunicação	Modelo Cliente/Servidor		
	Hardware										
Distributed Applications for Collaborative Augmented Reality	★	✗	★	★	★	★	✗	✗	✗	★	
Ambientes Colaborativos com Realidade Aumentada	✗	★	✗	✗	✗	✗	✗	✗	★	✗	
Distributed Augmented Chess System	✗	★	✗	★	★	✗	★	✗	★	✗	

Legenda:

★ Item Contemplado

✗ Item não Contemplado

Baseado no estudo comparativo dos sistemas acima, percebe-se a necessidade de investigar técnicas computacionais que possam melhorar o uso e a performance de sistemas de Realidade Aumentada Distribuídos, nos seguintes aspectos:

- Criação de Interfaces Interativas em R.A, onde pode-se navegar no mundo virtual por meio de teclado ou de marcadores.
- Camada de distribuição, permitindo a integração à interface de Realidade Aumentada e fazendo a distribuição das interações realizadas na interface. Essa camada é descrita em forma de uma arquitetura de distribuição de objetos em Realidade Aumentada.

# CAPÍTULO IV

## 4. ARQUITETURA DO SISTEMA

### 4.1 Introdução

A partir dos estudos realizados nos Capítulos II e III, foi possível conceber uma arquitetura de distribuição para ambientes de Realidade Aumentada que ofereça ao usuário condições de acompanhar e interagir com objetos virtuais em uma rede de computadores.

### 4.2 Tecnologias de Apoio

Novas tecnologias têm sido criadas para dar suporte ao desenvolvimento de aplicações em Realidade Virtual e Aumentada (SEMENTILLE, 1999). Tradicionalmente, os displays gráficos somente eram disponíveis em estações de trabalho. Porém, atualmente os preços do *hardware* têm caído, a ponto de permitir que essas capacidades gráficas estejam presentes nos computadores pessoais (PC's).

#### 4.2.1 CORBA

O padrão CORBA é uma plataforma que permite às aplicações distribuídas (local ou mesmo na internet) comunicarm-se e trocarem informações em uma rede de computadores. Esta tecnologia subsidia a distribuição e é um dos principais componentes da arquitetura proposta para essa dissertação.

#### 4.2.2 Visibroker

De acordo com Ribeiro (2006), pode-se afirmar que existem muitas implementações CORBA disponíveis. Para implementação deste sistema, foi utilizada a implementação CORBA da *Borland*<sup>TM7</sup>. O produto CORBA é chamado visibroker, que é um ORB (*Object Request Broker*) desenvolvido pela INPRISE (TEIXEIRA, 2002).

Para a escolha deste tipo de implementação recorre-se à Ribeiro (2006), Costa (2000) e Siqueira (2005) que relatam e compararam as arquiteturas de distribuição, demonstrando o potencial da implementação do CORBA.

---

<sup>7</sup> <http://info.borland.com.br>

#### **4.2.2.1. Osagent**

Objetos CORBA precisam de um meio para localizar um ao outro. O OMG oferece uma solução para isso com o *Naming Service* (serviço de nomeação), que é descrito na especificação CORBA. O *Naming Service* é um programa executado em algum lugar na rede. Os objetos no servidor se registram com o *Naming Service*, de modo que as aplicações cliente tenham um meio de localizar objetos específicos. O *Naming Service* exige um código adicional, tanto no cliente quanto no servidor. O local do processo precisa ser conhecido previamente, antes da solicitação de uma conexão com um objeto servidor.

Esse é um modo bem complicado de localização de servidores. O Visibroker possui um utilitário, que torna a localização do objeto muito mais fácil que usar o *Naming Service*. Esse programa é o *OSAgent* (TEIXEIRA, 2002). Ele não faz parte da especificação CORBA. *OSAgent* é um utilitário próprio, que só está disponível nesse ORB. Desde que o ORB Visibroker seja usado dentro da implementação CORBA, o *OSAgent* é o método preferido para a localização e conexão a objetos (TEIXEIRA, 2002).

#### **4.2.3 ARToolKit**

Kirner (2008) destaca que o ARToolKit é um *toolkit* que viabiliza o desenvolvimento de interfaces de Realidade Aumentada. Disponível gratuitamente no site do laboratório *HITLAB*<sup>8</sup> da Universidade de Washington, o ARToolKit emprega métodos de visão computacional para detectar marcadores na imagem capturada por uma câmera. O rastreamento óptico do marcador permite o ajuste de posição e orientação para realizar a renderização do objeto virtual, de modo que esse objeto pareça estar “atrelado” ao marcador.

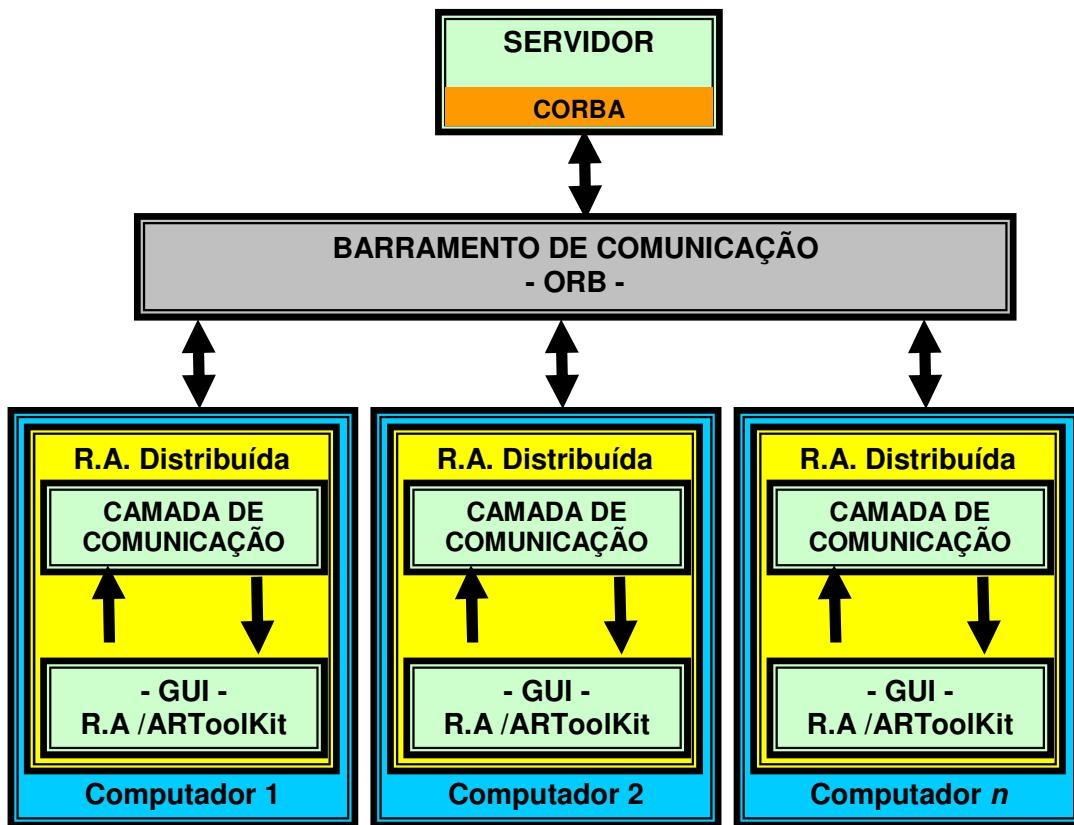
O ARToolKit tem suporte para a biblioteca Gráfica OpenGL na renderização dos objetos virtuais e na criação da janela gráfica onde são visualizados os objetos virtuais da cena em Realidade Aumentada e, ainda, trabalha com as premissas do OpenGL quando se trata de criar interações, animações, transformações geométricas, iluminação e criação menus.

---

<sup>8</sup> Laboratório de Tecnologia em Interfaces Humanas.

### 4.3 Arquitetura para Distribuição do Sistema

O propósito desta seção é a elaboração de uma arquitetura que permita a existência de  $n$  computadores capazes de hospedar Ambientes de Realidade Aumentada, fazendo assim, a distribuição dos objetos virtuais visualizados e manipulados na cena. Ainda, destaca-se a implementação de uma interface que permita a interação de objetos virtuais na cena em tempo real. Propõe-se a seguinte arquitetura para distribuição do sistema de Realidade Aumentada, conforme a Figura 4.1, baseado em um modelo CORBA Cliente/Servidor.



**Figura 4.1. Arquitetura proposta para a distribuição do sistema R.A. Distribuída.**

O sistema proposto nesta dissertação (*R.A. Distribuída*) é composto pelos seguintes módulos:

- **Interface de Realidade Aumentada (IRA):** A interface propicia a manipulação dos parâmetros interativos que serão distribuídos e, também a visualização dos objetos distribuídos que podem ter sido manipulados por outros usuários.
- **Servidor:** Servidor da aplicação recebe dos clientes e distribui na rede de computadores as informações acerca dos objetos que devem ser tratados e visualizados

nos marcadores em todas as interfaces de Realidade Aumentada. O servidor é independente da interface de Realidade Aumentada, sendo inicializado automaticamente no computador do primeiro cliente que for ativado.

- **Clientes:** Para cada interface R.A têm-se um Cliente de Comunicação. Esse Cliente faz parte da camada de Realidade Aumentada Distribuída.

- **ARToolKit:** *Toolkit* de código aberto para criação de ambientes de Realidade Aumentada, neste caso o produto de sua compilação é a interface de R.A.

- **Camada de Comunicação:** Junção das implementações feitas no ARToolKit e nos Clientes de Comunicação, onde é feita a passagem dos parâmetros para distribuição da interface de Realidade Aumentada para os Clientes de Comunicação e vice-versa.

Resumidamente, verifica-se que:

- ✓ A Figura 4.1 ilustra a existência de *n* computadores, sendo que cada computador pode hospedar apenas um único ambiente virtual de Realidade Aumentada e sempre haverá uma aplicação servidora que proverá serviços para os clientes. Cada objeto “R.A Distribuída” possui uma interface de R.A., uma Camada de Comunicação e um Cliente de Comunicação. A interface R.A. possui um ambiente baseado na *fisiologia vegetal*, já o Cliente de Comunicação trabalha diretamente com a distribuição.
- ✓ Cada interface de Realidade Aumentada pode ser executado separadamente e possui interações dentro do próprio ambiente. A distribuição ocorre quando uma interação feita por meio da oclusão for disparada.
- ✓ Os objetos virtuais foram modelados em VRML e carregados no ARToolKit, onde são transformados em uma matriz de pontos e vértices OpenGL.
- ✓ Por meio da integração das implementações realizadas no ARToolKit e na plataforma de distribuição CORBA, criou-se uma *camada de comunicação*.
- ✓ O usuário interage com um ambiente virtual em Realidade Aumentada, de tal forma que as informações capturadas pela interface de Realidade Aumentada são repassadas para a Camada de Comunicação. Desta forma, a aplicação-cliente fica lendo a Camada de Comunicação, verificando o que está sendo posicionado e, assim, enviando para a aplicação-servidora que, por sua vez disponibiliza para todas as outras aplicações-clientes.

- ✓ A aplicação-cliente localiza na rede de computadores uma aplicação-servidora com a ajuda do Visibroker (implementação CORBA), que responderá à requisição da aplicação-cliente.
- ✓ As informações de cada aplicação estarão armazenadas sempre em um objeto-servidor. Porém, cada cliente armazena informações sobre sua atual situação.

Dentro do contexto da implementação CORBA, é importante deixar claro que neste caso, não foi implementado nenhum algoritmo com aspecto de tolerância à falhas, ou seja, caso o servidor venha a parar de funcionar, a aplicação de Realidade Aumentada continua a funcionar, porém os parâmetros de distribuição são interrompidos por uma falha de falta de servidor ativo para os clientes.

#### 4.4 Pipeline<sup>9</sup> do Sistema

De acordo com a Figura 4.2, ilustra-se o funcionamento de um Pipeline de funcionamento do algoritmo apresentado nesta dissertação e adaptado à arquitetura do ARToolKit, onde pode-se relatar:

✓ **Camada de Comunicação:** Como já mencionado no item “arquitetura para distribuição” neste Capítulo, esta *camada* recebe parâmetros<sup>10</sup> do Cliente de Comunicação, que por sua vez recebe os parâmetros do Servidor de Comunicação. Então estes parâmetros são lidos pela interface de Realidade Aumentada (ARToolKit) e, de acordo com o parâmetro é então feita a visualização do objeto virtual na cena para todos os clientes e interfaces R.A que tenham o marcador de referência do objeto.

✓ **Controle de Interação:** O controle realizado nas interações da interface de Realidade Aumentada é realizado por meio de várias técnicas para criar interações que permitam obter um ambiente com um nível satisfatório de interatividade. Assim, pode-se destacar as interações na interface do sistema: a) Transformações Geométricas: o sistema realiza em tempo de execução, todas as transformações geométricas. b) Colisão de Objetos Virtuais: o sistema detecta a colisão<sup>11</sup> de objetos virtuais, esses objetos estão dispostos na interface em forma de menus suspensos, onde o usuário ao colidir o marcador em qualquer um dos objetos disponíveis, faz a sua captura, ou seja, o objeto

---

<sup>9</sup> Pipeline: conjunto de processos encadeados por meio das suas saídas padrões, de forma que a saída de um processo é utilizada como entrada do processo seguinte.

<sup>10</sup> Cada objeto virtual na cena gráfica do ARToolKit possui uma numeração, que vai de 1 à 6.

<sup>11</sup> A colisão trabalhada aqui se trata de uma colisão por aproximação, onde verifica-se a posição do objeto virtual visualizado na cena gráfica, e ao aproximar-se de um valor determinado efetua-se uma função qualquer, desta forma a cena gráfica do ARToolKit foi mapeada.

virtual que está sendo visualizado na tela é instanciado para o marcador que realizou a colisão. c) Oclusão de Marcadores: ao realizar o item “b” deste parágrafo, o objeto virtual com seu respectivo marcador ao ser posicionado à frente de um outro marcador, onde o sistema de Realidade Aumentada verifica se o marcador que esta sendo sobreposto está ocluído ou não, caso a oclusão tenha ocorrido, o marcador que foi ocluído recebe o objeto virtual do marcador de ocluiu.

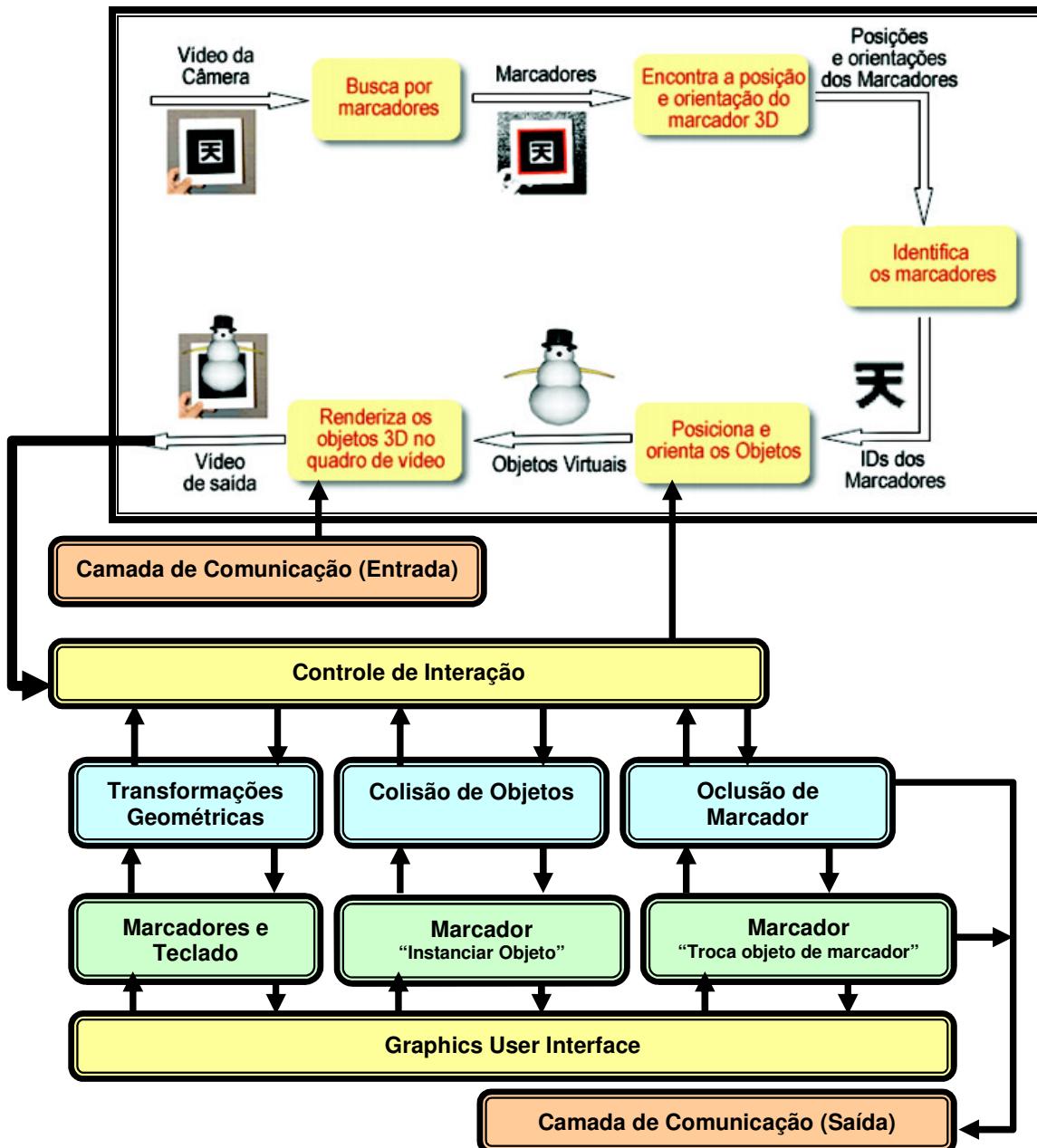


Figura 4.2. Diagrama da interface R.A adaptada à Arquitetura do ARToolKit

✓ **Graphics User Interface (GUI):** Este bloco permite ao usuário visualizar graficamente, de modo interativo e em tempo real, a entrada de dados e a saída de

informações. A GUI exibe então o cenário, apresentando os ambientes virtuais em Realidade Aumentada em uma janela gráfica por meio de dispositivo de captura de vídeo e saída no monitor. Essa interface gráfica com o usuário permite realizar interações em tempo real. Essas interações ocorrem por meio das transformações geométricas, troca de objetos virtuais na cena, colisão de objetos virtuais e oclusão de marcadores.

#### **4.5 Considerações Finais**

Entende-se que a junção do Pipeline de interações da Interface de Realidade Aumentada com a Arquitetura de Distribuição feita por meio de uma *Camada de Distribuição* proporciona um ambiente virtual distribuído, onde as camadas de interface R.A e a Aplicação de Distribuição são conectadas por uma Camada de Comunicação.

No próximo capítulo, serão mostrados os detalhes da implementação da arquitetura e do pipeline do sistema proposto.

# CAPÍTULO V

## 5. DETALHES DE IMPLEMENTAÇÃO

### 5.1 Introdução

Este Capítulo apresenta os detalhes da implementação do sistema **R.A. Distribuída**, abordando todas as fases necessárias à execução da interface de Realidade Aumentada e a Distribuição dos objetos virtuais nela contidos.

### 5.2 Modelagem dos Objetos Virtuais

Os objetos virtuais visualizados no sistema de Realidade Aumentada são baseados nos conceitos da *fisiologia vegetal*<sup>12</sup>. Desta forma, com a concepção de objetos virtuais baseados em uma área do conhecimento específica, neste caso a biologia, pode ilustrar o potencial que este protótipo possui para a aplicação em áreas educacionais.

Neste sentido, o presente protótipo possui sete objetos virtuais, porém é válido ressaltar que podem ser acrescidos uma quantidade maior de objetos virtuais. Esses objetos foram modelados na ferramenta 3D Studio Max. Esta ferramenta possui um alto poder de produtividade, onde o usuário não fica preso a implementações com linha de código. Ao término da modelagem, os objetos virtuais foram exportados para o formato “.wrl” do VRML. A exportação dos objetos é necessária, pois este formato é suportado pelo ARToolKit em sua versão 2.52.

### 5.3 Implementação da Interface de Realidade Aumentada

A interface de Realidade Aumentada foi implementada, como já visto anteriormente, utilizando-se a base do ARToolKit versão 2.52 (KATO & BILLINGHURST, 2005). Desta forma, o código do ARToolKit alterado foi o *simpleVRML.cpp* que fica contido na pasta *examples*. Este código foi escolhido devido ao fato de trabalhar com objetos em formato VRML.

---

<sup>12</sup> A fisiologia vegetal estuda os fenômenos vitais que concerne às plantas. Estes fenômenos podem referir-se ao metabolismo vegetal; ao desenvolvimento vegetal; ao movimento vegetal ou a reprodução vegetal.

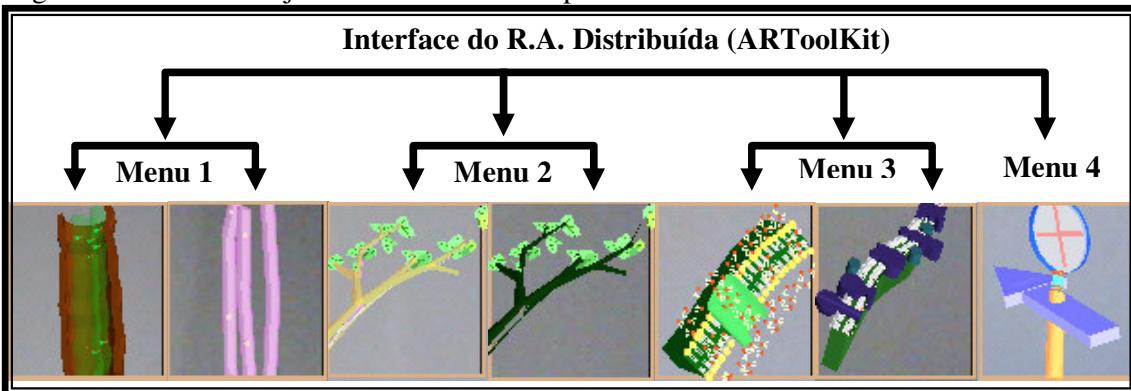
### 5.3.1 Implementação das Interações via Teclado

Para a implementação das interações via teclado, utilizou-se a função *keyEvent* implementada no ARToolKit. Por padrão esta função vem apenas com o evento da tecla “escape” para fechar a aplicação.

As interações na interface de Realidade Aumentada do software *R.A Distribuída* baseiam-se nas transformações geométricas: escala, translação e rotação. Essas transformações ocorrem em tempo real de execução. Assim, destacam-se as interações implementadas: redução e aumento da escala dos objetos, translação de objetos para todos os eixos, rotação dos objetos para todos os eixos e uma função de retorno ao ponto inicial de *default* do aplicativo.

### 5.3.2 Implementação dos Menus da Interface de Realidade Aumentada

Para a implementação das interações realizadas por meio de marcadores tornou-se necessário a criação de menus na interface de Realidade Aumentada. Esses menus de interações ficam visíveis na cena gráfica da aplicação sem a utilização de marcadores. A Figura 5.1 ilustra os objetos virtuais e seus respectivos menus.



**Figura 5.1. Esquema dos menus interativos e seus objetos virtuais**

Os menus interativos de um a três podem conter *n* objetos, porém para a implementação desta dissertação, optou-se por dois objetos por menu. Já o menu interativo de número quatro funciona como um painel de controle para a realização das transformações geométricas por meio de marcadores. A Figura 5.2 ilustra a relação. Desta forma, dentro da função *mainLoop* na implementação do ARToolKit “pede-se” para visualizar os objetos na cena gráfica do vídeo. Para proporcionar a opção de dois objetos virtuais por menu, trabalha-se com estruturas de condição. É importante ressaltar que todos os objetos virtuais relacionados devem estar previamente cadastrados. Desta forma, ao incorporar os objetos virtuais, o ARToolKit por meio das bibliotecas *libvrml97gl.lib* e *libARvrml.lib* cria na função

*arVrml97Draw* uma lista de objetos virtuais do tipo VRML e, assim por meio desta lista de objetos virtuais pode-se criar várias possibilidades de interatividade. A Figura 5.3 ilustra parte do pseudocódigo da implementação dos menus interativos.

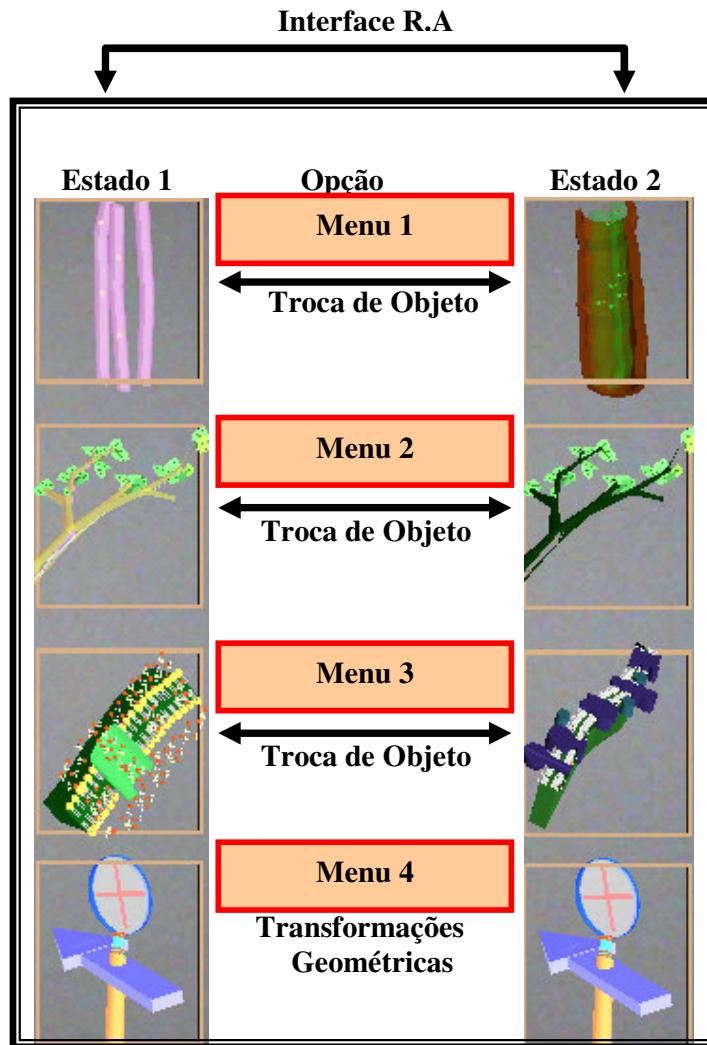


Figura 5.2 Relação de objetos virtuais por menu interativo

Para a disposição dos menus interativos na cena gráfica do ARToolKit foi necessário mapear toda a tela gráfica gerada pela biblioteca *libARgsub.lib*, para isso foi realizado vários testes durante o desenvolvimento da aplicação *R.A. Distribuída..*

```
// Primeiro Menu
Se (variável == 0) {
    objeto_vrml(visualiza objeto 1);
}
Senão se (variável == 1) {
    objeto_vrml(visualiza objeto 2);
...
}
```

Figura 5.3. Parte da implementação dos menus interativos

### 5.3.3 Implementação da opção de troca de objeto virtual por meio do menu suspenso

Para a implementação da opção de troca de objeto virtual por menu, como demonstrado no item “*Implementação dos Menus da Interface de Realidade Aumentada*” neste capítulo, necessitou-se utilizar as propriedades de janelas da biblioteca *glut32.lib* (GLUT, 2007) para criação de um menu suspenso. (menu do Windows). Assim, a Figura 5.4 ilustra o menu suspenso.



Figura 5.4. Menu suspenso na Interface de R.A

O menu suspenso possibilita a troca de objetos virtuais na cena gráfica, desta forma utiliza-se estruturas de condição para verificar qual objeto virtual será visualizado na cena. A Figura 5.5 ilustra parte do pseudocódigo do gerenciamento do menu suspenso.

<pre>//Gerenciamento do Menu 01 função Menu01(opção ) {     Escolha (opção) {         caso 0:             variável = 1;             pare;         caso 1:             variável = 0;             pare;     } } ...</pre>	<pre>// Gerenciamento do Menu 02 função Menu02(opção ) {     Escolha (opção) {         caso 0:             variável = 0;             pare;         caso 1:             variável = 1;             pare;     } } ...</pre>
---	--

Figura 5.5. Parte do pseudocódigo do menu suspenso na interface de R.A

E, por fim, cria-se uma função para gerenciar o mouse. Esta função verifica se o botão direito do mouse foi pressionado ou não. Caso o botão direito do mouse tenha sido pressionado, a função *gerenciamouse* chama a função *criamenu*, conforme pode ser observado na Figura 5.6.

<pre>função gerenciamouse( Qual botão foi pressionado?) {     se (botão == Botão_Direito)         se (estado == Foi_Pressionado)             CriaMenu(); }</pre>
--

Figura 5.6. Parte do pseudocódigo da função gerenciamouse

### 5.3.4 Implementação das Interações por meio de Marcadores

A implementação das interações por meio de marcadores segue as mesmas características do item “*Implementação das Interações via Teclado*”, porém é feita por meio da colisão de objetos virtuais com o *menu de interação quatro* (Menu 04) descrito no item “*Implementação dos Menus da Interface de Realidade Aumentada*”. Assim, neste sentido, tem-se um marcador para cada transformada geométrica.

A colisão trabalhada aqui se trata de uma colisão por aproximação, onde verifica-se a posição do objeto virtual visualizado na cena gráfica, e ao aproximar-se de um valor determinado efetua-se uma função qualquer. Como já descrito no item anterior, a cena gráfica o ARToolKit foi mapeada, assim, quando tem-se um objeto virtual que, neste caso fica invisível, atrelado a um marcador que estiver entre dois valores no eixo “X” e dois valores no eixo “Y”, tem-se efetuada uma colisão aproximada.

Assim, entende-se que, ao verificar tal condição, pode-se implementar qualquer tipo de ação e, neste caso, foi implementado o mesmo código do item “*Implementação das Interações via Teclado*”. Neste sentido ilustra-se apenas parte do pseudocódigo da colisão aproximada, conforme demonstrado na Figura 5.7.

```
//Colisão aproximada para realizar transformações geométricas com Marcadores
se (marcaA > 130 E marcaA < 220 E marcaA < 190 E marcaA > 130 )
{
    funçãoaumentaescala();
}
senão
se (marcaR > 130 E marcaR < 220 E marcaR < 190 E marcaR > 130 )
{
    funçãoreduzescala();
}
senão
se (marcaD > 130 E marcaD < 220 E marcaD < 190 E marcaD > 130 )
{
    funçãotransladadireita();
}
    funçãotransladabaixo();
}
senão
se (marcaZM > 130 E marcaZM < 220 E marcaZM < 190 E marcaZM > 130 )
{
    funcaotransladazmais();
}
senão ...
```

Figura 5.7. Parte da implementação das transformações geométrica por meio de colisão aproximada

### 5.3.7 Interação com os Menus da Interface de Realidade Aumentada

Para interação com os menus da interface de Realidade Aumentada, como já descrito no item anterior “*Implementação das Interações por meio de Marcadores*” neste capítulo, utiliza-se colisão aproximada de objetos virtuais. Assim, trabalha-se com apenas um marcador para interagir com todos os três menus. Este marcador faz a interação apenas com os três primeiros menus de cima para baixo na interface de Realidade Aumentada.

O marcador em formato de uma “pá”, ao fazer a colisão com algum objeto virtual que esteja entre dois valores no eixo “X” e dois valores no eixo “Y”, faz a captura do objeto virtual, ou seja, cria uma instância do objeto que está sendo visualizado na cena gráfica no marcador “pá” que fez a colisão.

Ainda, nesta parte da implementação são gravados alguns parâmetros para a distribuição dos objetos virtuais, desta forma, ao se verificar a colisão de algum objeto, os parâmetros do mesmo são gravados em uma variável. Deste modo, criou-se uma variável global do tipo texto (*objetoatual*) que armazena sempre o último objeto que sofreu algum tipo de colisão. Mas adiante, o parâmetro desta variável será distribuído para todas as outras interfaces de Realidade Aumentada disponíveis na rede. É importante ressaltar que existem apenas três menus de interação, porém são seis objetos virtuais para serem visualizados, ou seja, são seis objetos virtuais que podem sofrer colisão do marcador “pá”. Neste aspecto, utilizou-se de estruturas de condição para verificar qual objeto virtual está sendo visualizado no momento da colisão do marcador “pá”. Este assunto já foi comentado no item “*Implementação dos Menus da Interface de Realidade Aumentada*” deste capítulo. A Figura 5.8 ilustra o pseudocódigo da implementação.

```
//Verifica o objeto que deverá ser instanciado no marcador “pá”
se (variável == 0) {
    se (marca0 >130 E marca0 < 220 E marca0 < -50 E marca0 > -110 )
    {
        marca0 = objetovirtualdomenu1;
        objetoatual = '1';
    }
    senão
    se (variável == 1) {
        se (marca0 >130 E marca0 < 220 E marca0 < -50 E marca0 > -110 )
        {
            marca0 = objetovirtualdomenu1;
            objetoatual = '2';
        }
    senão ...
}
```

Figura 5.8. Parte da implementação das interações por colisão com os menus interativos

### 5.3.8 Implementação dos parâmetros para Distribuição

Para implementação da distribuição foi necessário criar uma camada para gravação e leitura dos parâmetros dos objetos virtuais que devem ser distribuídos na rede. Essa camada foi chamada de *Camada de Comunicação* e é composta de dois módulos. Um dos módulos é chamado de “*saída*”, este módulo possui duas funcionalidades: para a interface de Realidade Aumentada é um módulo de leitura, já para a interface de comunicação, é um módulo de gravação. O outro módulo criado é chamado de “*entrada*”, este módulo também possui duas funcionalidades: para a interface de Realidade Aumentada é um módulo de gravação, já para a interface de comunicação, é um módulo de leitura. As Figuras 5.9 e 5.10 ilustram o pseudocódigo dos módulos da camada de comunicação.

```
//Grava módulo para Distribuição
função atualizaarquivo(caracter)
{
    TipoModulo *modulo = nulo;

    se ((modulo=estiveraberto("entrada","escreve")))
    {
        retorno;
    }
    escreva(valor de entrada no modulo);
    fecha(modulo);
}
```

Figura 5.9. Camada de Comunicação (módulo entrada)

```
// Lê módulo para Distribuição
caracter de inicialização ='0';
TipoModulo *modulo;

if(modulo=estiveraberto("saida","leia"))
{
    variável=ler(modulo);
    fechar(modulo);
}
variavelatual = variavel;

// Grava modulo para Distribuição
atualizamódulo(variavelatual);
```

Figura 5.10 Camada de Comunicação (módulo saída)

Assim, tanto o código de implementação do módulo “*saída*”, quando o código do módulo “*entrada*” devem estar dentro da função “*argmainLoop*” implementado no ARToolKit. Para o objeto virtual escolhido ser gravado na *camada de comunicação*, o evento de oclusão de marcadores deve ocorrer. Esse evento ocorre quando um marcador já previamente cadastrado no sistema de Realidade Aumentada é ocluso. Para este protótipo foi cadastrado no sistema *R.A Distribuída* o marcador *hiro*. Quando o marcador *hiro* for ocluso pelo marcador “*pá*” uma instância do objeto que está anexado ao marcador “*pá*” será criada no marcador *hiro*. E, ainda, será disparado um evento para gravação na *camada de comunicação* do parâmetro referente ao objeto instanciado, assim o mesmo será visualizado por todos os clientes R.A da uma rede de computadores. A Figura 5.11 ilustra parte do pseudocódigo que proporciona o evento de oclusão.

```
// Tratamento de Oclusão
se (ObjetoMarcadorHiro não for visível) {
    ObjetoMarcadorHiro=ObjetoMarcadorPa;
    atualizaarquivo(variavelatual);
}
...
...
```

Figura 5.11. Parte do código do tratamento de Oclusão

É importante destacar que o código do tratamento de oclusão deve estar dentro da função *argmainLoop* implementada no ARToolKit. A seguir é demonstrado o pseudocódigo para visualização de objetos virtuais recebidos do servidor, observe a Figura 5.12.

```
// Gerenciamento de parâmetros enviados pelo servidor
se (objetolidoat != objetolidoan) {
    se (objetolidoat != objetoatual) {
        objetolidoan = objetolidoat;
    se (objetolidoat == '1')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[1];
    senão se (objetolidoat == '2')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[2];
    senão se (objetolidoat == '3')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[3];
    senão se (objetolidoat == '4')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[4];
    senão se (objetolidoat == '5')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[5];
    senão se (objetolidoat == '6')
        ObjetoMarcadorHiro=ObjetoMarcadorPa[6];
    senão se (objetolidoat == '0')
        ObjetoMarcadorHiro=ObjetoMarcadorPa;
}
}
```

Figura 5.12. Parte da verificação de objetos vindos do servidor para renderização

Desta forma, na interface de Realidade Aumentada, para receber do servidor os parâmetros de objetos virtuais para a visualização, foi implementada uma rotina de verificação, onde o sistema **R.A Distribuída**, ao identificar um objeto virtual que não esteja visualizado na tela, automaticamente faz a visualização na cena.

## 5.4 Implementação da Arquitetura de Comunicação

Na implementação da distribuição e comunicação entre os Ambientes Virtuais de Realidade Aumentada foi usado um ambiente de programação com suporte ao Visibroker 4.1 (TEIXEIRA, 2002). A Figura 5.13 mostra o diagrama de blocos da plataforma CORBA 2.0. A parte comum para cliente e servidor é o ORB (LIMA, 2002). O ORB trata de toda comunicação entre os objetos. Além de cuidar de todo o tráfego de mensagens, o ORB também corrige variações de plataforma.

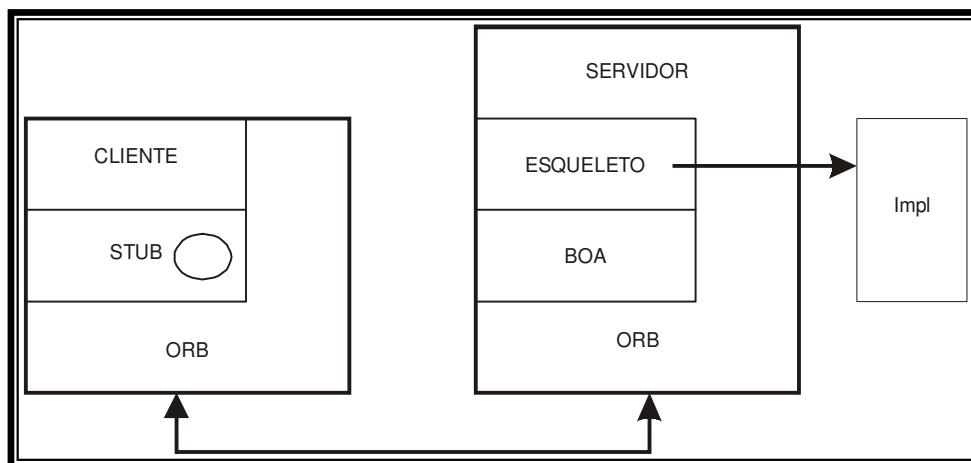


Figura 5.13. Plataforma CORBA 2.0 (TEIXEIRA, 2002).

A Figura 5.13, o lado do cliente consiste em duas camadas. O bloco cliente é a aplicação que foi escrita pelo desenvolvedor. A parte mais interessante é o *stub*. O *stub* é um arquivo gerado automaticamente pelo compilador de IDL. Sua finalidade descrever as interfaces do servidor e gerar código que possa interagir com o ORB da plataforma CORBA 2.0. O arquivo de *stub* contém uma ou mais classes que “espelham” o servidor CORBA. As classes contêm as mesmas interfaces publicadas e tipos de dados que são expostos pelo servidor. O cliente usa classes *stub* a fim de se comunicar com o servidor. As classes *stub* atuam como um *proxy* para os objetos servidores. O símbolo arredondado no bloco de *stub* representa uma conexão com o servidor. A conexão é estabelecida por meio de uma chamada de ligação emitida pelo cliente. O *stub* invoca a chamada, por meio de sua referência ao objeto servidor, usando o ORB. Quando o servidor responde, a classe *stub* recebe a mensagem do ORB e entrega a resposta de volta ao cliente. O cliente também pode chamar algumas funções no ORB.

O lado servidor contém uma interface ORB chamada de adaptador básico de objetos (*Basic Object Adaptor – BOA*) (LIMA, 2002). O BOA é responsável por “*rotar*” as mensagens do ORB para a interface de esqueleto. O esqueleto é uma classe gerada pelo compilador IDL, assim como o *stub*. O esqueleto contém uma ou mais classes que publicam as interfaces CORBA no lado do servidor. Existe outro arquivo, que contém as classes que representam os detalhes funcionais do servidor. Este é conhecido como arquivo IMPL (implementação).

Quando uma mensagem chega ao lado do servidor, o ORB passa um *buffer* de mensagem ao BOA, que, por sua vez, passa o *buffer* à classe do esqueleto. O esqueleto desembrulha os dados do *buffer* e determina qual método deve ser chamado no arquivo IMPL. O esqueleto apanha os resultados de retorno e os conduz de volta ao cliente. (TEIXEIRA, 2002).

Como existe apenas um tipo de ambiente virtual e, neste se faz necessário distribuir por meio de replicação ou particionamento, assim, a metodologia aplicada resume-se em implementar as duas partes (cliente e servidor), sendo que a aplicação-cliente fica junto com a aplicação do ambiente virtual de Realidade Aumentada, e o servidor ficaria responsável por controlar os clientes e as informações dos mesmos. A primeira aplicação iniciada na rede sempre faz a requisição de um servidor. Outras aplicações iniciadas usam o mesmo servidor.

Para este protótipo, a implementação está dividida em duas fases: a) criação de uma interface para o objeto, pois todo objeto CORBA é descrito por sua interface; b) a segunda fase consiste em escrever o código respectivo a cada método declarado na interface. A aplicação servidora necessita publicar declarações de tipo, interfaces e métodos específicos num projeto puramente orientado a objeto. Esta descrição de interface foi escrita na linguagem de Definição de Interface (IDL – *Interface Definition Language*) e traduzida para a linguagem específica, por meio de um compilador (termo mais apropriado seria gerador de código). A seguir a Figura 5.14 ilustra o pseudocódigo da interface do ambiente de distribuição.

```
// IDL da Camada de Distribuição
modulo Distribui {
    interface RAD {
        variável interfere(tipo inteiro posição);
        variável situação();
        função usuário(tipo caractere nome);
    };
};
```

Figura 5.14. IDL da Camada de Distribuição

A interface necessária ao projeto foi processada gerando os seguintes arquivos:

**Distribui\_I:** arquivo que contêm todas as interfaces e definições de tipo.

**Distribui\_C:** arquivo que contêm quaisquer tipos definidos pelo usuário, exceções e classes *stub* do cliente.

**Distribui\_S:** esse arquivo possui o esqueleto das definições de classe no servidor.

**Distribui\_Impl:** esse arquivo possui uma definição de classe geral para uma aplicação no servidor. Neste arquivo foram inseridas as implementações para realizar as ações necessárias aos clientes e que são responsabilidade do servidor, observe o pseudocódigo da Figura 5.15.

```
// Distribui_Impl
...
tipo
    usuarios = gravação
        identificadorusuario : tipo inteiro;
        local : tipo caractere;
        conexão : (verdadeiro/falso);
    fim;
...
tipo
    TRAD = classe;
    TRAD = classe(Interface de Objetos, Distribui_i.RAD)
parte protegida;
    função posicao(): tipo inteiro;
público
    construtor Cria();
    função interfere ( constante tipo inteira posicao);
    função situacao do tipo inteiro;
    função usuário ( constante tipo caractere nome);
fim;
```

Figura 5.15. Visão parcial do arquivo *de implementação*

A segunda fase da implementação foi escrever os códigos de pedido dos clientes e as respectivas respostas dos servidores. No servidor, o código recebe os pedidos e os armazena, verificando se os mesmos não causavam nenhuma exceção. As implementações nas aplicações clientes resumem-se em interceptar a interação do usuário no ambiente virtual de Realidade Aumentada e transformá-la em valor (estado), enviando ao servidor.

#### 5.4.1 Codificação do Objeto-Cliente

A idéia principal do protótipo de distribuição deste trabalho é usar um servidor para armazenar as informações de seus clientes. Porém, a abordagem (protótipo) desenvolvida

possui uma característica específica. A aplicação de distribuição não está anexada na aplicação de interface de Realidade Aumentada, possuindo uma **camada de comunicação** entre as aplicações. Os parâmetros para distribuição são gravados na camada de comunicação pela interface de Realidade Aumentada e, assim, a aplicação-cliente fica lendo essas informações que são enviadas para a aplicação-servidora que, por sua vez faz a distribuição para todas as outras aplicações-clientes ativadas na rede de computadores.

Em cada computador de uma rede possui uma aplicação-cliente para a distribuição, uma interface de Realidade Aumentada e uma camada de comunicação para gravar e ler os objetos virtuais que deverão ser visualizados na cena gráfica. A Figura 5.16 ilustra os pseudocódigos da aplicação-cliente, em relação às rotinas de gravação e de leitura na camada de comunicação.

```
//Atualiza Módulo de Saída
função ClienteRA.AtualizaSaida( );
Variável Texto : Tipo Caractere;
início
    Texto := CrieLista.Create;
    Texto.Abre('Saída');
    Texto.AdicionaParametro( );
    Texto.Limpar;
    Texto.Grava('Saída');
    MostreMensagem ('Lendo e Atualizando Módulo!');
fim;
//Atualiza Módulo de Entrada
função ClienteRA.AtualizaEntrada( );
Variável Texto : Tipo Caractere;
início
    Texto := CrieLista.Create;
    Texto.Abre('Entrada');
    Texto.AdicionaParametro( );
    Texto.Limpar;
    Texto.Grava('Entrada');
    MostreMensagem('Lendo e Atualizando Módulo!');
fim;
```

Figura 5.16. Gravação e leitura da camada de comunicação (Cliente)

É importante elucidar que a estrutura de comunicação da arquitetura de comunicação deste projeto, foi construída automaticamente por um *toolkit* para implementações CORBA 2.0. Essa estrutura criada, é baseada em uma IDL e, ainda, a comunicação é proporcionada pelo ORB Visibroker 4.1 da Borland. Desta forma, implementam-se apenas as passagens de fluxo, os tratamentos de erros e regras gerais necessárias para funcionamento desta aplicação. Assim, a Figura 5.17 ilustra o parte do pseudocódigo da função de inicialização do objeto-

cliente por meio do objeto CORBA, essa função é criada automaticamente pelo *toolkit* da ferramenta, porém os códigos necessários para o funcionamento devem ser implementados.

```
//Inicialização do Cliente-CORBA
procedimento ClienteRA.InitCorba;
variáveis: enc, fechar: tipo (verdadeiro/falso) ; tent : tipo inteiro;
Início
    tent := 0;
    fechar := falso;
    // Caso não encontre a aplicação, executar o servidor
    servidor := verdadeiro;
    repita
        enc := verdadeiro;
        tente realizar
            Incrementa(tent);
            RA := RADHelper.ProcureServidor;
            caso haja um erro faça
                enc := falso;
                Execute arquivo executável ('RAServer.Exe',0);
                se (tent > 20) então
                    se MensagemApresentada ('Servidor não encontrado, deseja continuar?', 
                        TipoConfirmação, [Sim, Não], 0) = Sim então
                            inicio
                                enc := verdeiro;
                                servidor := falso;
                            fim
                            senão
                                inicio
                                    enc := verdadeiro;
                                    fechar := verdadeiro;
                                fim;
                                até que (enc = verdadeiro);

                se fechar = verdadeiro então
                    TermiteTodaAplicação();
                fim;
```

Figura 5.17 Inicialização do Objeto-Corba no Cliente

De acordo com a Figura 5.17, quando uma aplicação cliente for inicializada, o Cliente-CORBA deverá ser inicializado automaticamente, quando isso é feito, a primeira coisa que é realizada é a procura por um Servidor-CORBA. Caso não haja nenhum servidor ativo na rede de computadores, automaticamente será retornada uma mensagem de erro padrão. Neste caso, foi implementado um parâmetro de tratamento de erros, ou seja, caso o servidor não seja encontrado, automaticamente o cliente irá chamar o servidor da aplicação e, irá se auto-inicializar após o servidor estar ativo.

As implementações apresentadas neste tópico são exclusivamente da aplicação Distribuída (Cliente) do projeto proposto para esta dissertação, assim, ilustra-se por meio da Figura 5.18 a interface de comunicação cliente.



Figura 5.18. Interface da aplicação distribuída (Cliente)

Esta aplicação chama em seu código os arquivos: *Corba*, *Distribui\_I*, *Distribui\_C*, *Distribui\_S* e *Distribui\_Impl*, mostrados neste tópico, isso é padrão da plataforma CORBA.

Outro fator importante é a verificação que a aplicação cliente fica fazendo na rede para gravar e ler os parâmetros de distribuição dos objetos virtuais e, ainda para verificar se a aplicação está conectada ou não. Desta forma, ilustra-se por meio da Figura 5.19 parte do pseudocódigo da função que verifica se a aplicação está conectada.

```
//Verifica o que está ocorrendo
procedimento ClienteRA.Tempo();
inicio
  se servidor=verdadeiro então
    MostreMensagem('Conectado');
  senão
    MostreMensagem ('Desconectado');
    AtualizaEntrada(0);
    Se (PosEntrada <> PosSaída) então
      Atualiza();
    PosSaída := RA.situacao;
    AtualizaSaida(PosSainda);
    MostreMensagem (PosSaída);
fim;
```

Figura 5.19. Verificação de tempo no Cliente

É importante deixar claro que os pseudocódigos apresentados neste tópico podem necessitar de algumas variáveis globais ou mesmo funções de sistema ou da própria ferramenta que não foram contempladas. Entende-se que os pseudocódigos apresentados nesta dissertação são para ilustrar e demonstrar como foi realizada a implementação do protótipo.

#### 5.4.2 Codificação do Objeto-Servidor

A implementação da aplicação servidora é bem simples, pois não há interface, conforme pode ser observado na Figura 5.20.

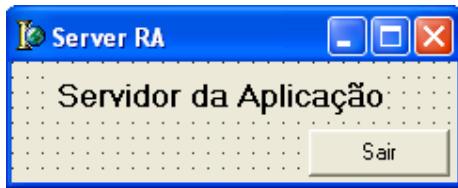


Figura 5.20. Aplicação Servidora

A aplicação servidora resume-se em receber os parâmetros dos clientes e distribuir na rede de computadores para todos os outros clientes que estão conectados a ela. A aplicação servidora também necessita dos arquivos da estrutura CORBA, isto é padrão para implementações baseadas em CORBA em qualquer linguagem de programação, assim os arquivos são os mesmos inseridos na implementação da aplicação cliente, ou seja: Corba, Distribui\_i, Distribui\_s, Distribui\_C, Distribui\_Impl. Desta forma, a Figura 5.21 ilustra parte do pseudocódigo da função de inicialização do Objeto-CORBA na aplicação servidora.

```
//Inicialização do Servidor-CORBA
procedimento ServerRA.InitCorba;
inicio
    InicializaçãoCORBA();
    tente realizar
        InicializaçãoCORBA();
        RA := RADSkeleton.CrieServidor('Servidor',TRAD.Crie);
        BOA.Leitura de Objeto(Objeto RA);
    caso haja um erro faça
        MostreMensagem ('Erro na Execução do Servidor!');
    fim;
    fim;
```

Figura 5.21. Inicialização da Aplicação Servidora

Basicamente a implementação do servidor, no caso desta dissertação, se resume a implementação realizada na Figura 5.21. Nota-se que para a implementação de sistemas distribuídos baseados em CORBA é importante entender os conceitos da plataforma de distribuição CORBA e, ainda entender a ferramenta em que se está implementado a aplicação, pois cada ferramenta possui uma peculiaridade, geralmente as plataformas de programação orientadas a objetos possuem *toolkit's* para criar os arquivos de implementação e de comunicação do CORBA. Porém existem inúmeros detalhes que devem ser observados, como por exemplo, entender todos os arquivos gerados pelo CORBA e acrescentar em seu código as regras de funcionamento da aplicação em desenvolvimento.

## 5.5 Considerações Finais

O Capítulo apresentou um ambiente de Realidade Aumentada que se comunica devido à aplicação do padrão para objetos distribuídos CORBA 2.0 por meio da

#### implementação Visibroker 4.1.

O sistema **R.A. Distribuída** implementado mostra a abordagem de comunicação baseado em Cliente/Servidor, no qual, cópias do mesmo ambiente de Realidade Aumentada se comunicam em uma rede de computadores, sendo que a localização do computador, onde está um servidor ou cliente, é transparente. A implementação mostrou a possibilidade de uma comunicação bidirecional entre os ambientes com replicação para as cópias distribuídas em uma rede de computadores.

No próximo capítulo, apresenta-se o funcionamento do sistema e o estudo de caso, onde deverá ser ilustrado o funcionamento da aplicação juntamente com os procedimentos que demonstram como ocorre parte da fisiologia vegetal. Para esta dissertação será ilustrado por meio de uma árvore de médio porte.

# CAPÍTULO VI

## 6. FUNCIONAMENTO DO SISTEMA E ESTUDO DE CASO

### 6.1 Introdução

Este capítulo descreve a forma de funcionamento do sistema proposto baseado em um estudo de caso (*fisiologia vegetal*). Assim, descreve-se o que é necessário para que o sistema *R.A Distribuída* funcione corretamente. Entende-se que um estudo de caso é importante para deixar clara a idéia da utilização do *software* na área educacional, neste sentido, apresentam-se alguns conceitos sobre a Fisiologia Vegetal e, em seguida, o funcionamento do sistema implementado.

### 6.2 Estudo de Caso

O presente estudo de caso abrange os aspectos da *fisiologia vegetal*, porém entende-se que este protótipo de *software* está focado também nas questões de tecnologia, ou seja, uma das contribuições está na construção de uma aplicação distribuída de Realidade Aumentada com interface interativa com forte apelo pedagógico. Entende-se que o estudo de caso contribui para ilustrar a possibilidade da utilização do software no processo de ensino/aprendizagem e, ainda para a demonstração de funcionamento geral do sistema em um aspecto educativo.

#### 6.2.1. Fisiologia Vegetal

Fisiologia Vegetal é a ciência que se propõe a explicar todos os processos da vida por meio de princípios da física e da química (AWAD et al., 1993). Isso quer dizer que o objetivo básico da fisiologia vegetal é tentar explicar qualquer processo que se realiza na planta em bases físicas e químicas. A fisiologia vegetal é realmente o estudo da planta em trabalho (TAIZ, 2006). Um dos objetivos básicos é a determinação da natureza química precisa de todos os materiais encontrados na planta e o caminho que ela segue para obtê-los ou faze-los. Isso é a bioquímica<sup>13</sup> da planta. O outro objetivo é o conhecimento de toda espécie de trabalho feito pela planta (mecânico, químico, osmótico, elétrico) e a natureza da energia envolvida. Isso é a biofísica<sup>14</sup> da planta. Mesmo a bioquímica da planta sempre envolve mudanças

---

<sup>13</sup>Para maiores informações acesse o sitio: <http://www.portalbiologia.com.br/biologia/principal>

<sup>14</sup>Para maiores informações acesse o sitio: <http://www.portalbiologia.com.br/biologia/principal>

energéticas. Assim, o denominador comum de todos os aspectos da fisiologia da planta é o trabalho e a energia (PORTAL BIOLOGIA, 2007).

### 6.2.2. Transporte de Solutos (Seiva)

O sistema de condução de materiais pelos corpos dos seres vivos deve garantir a distribuição de nutrientes e retirada de substâncias tóxicas das células dos tecidos de todo o organismo (TAIZ, 2006).

Nos vegetais a condução de solutos (**Seiva**), isto é, soluções salinas e soluções açucaradas, é realizada por meio dos sistemas de vasos, que se distribuem ao longo do corpo das plantas. A Figura 6.1 ilustra os vasos que realizam a distribuição dos solutos.

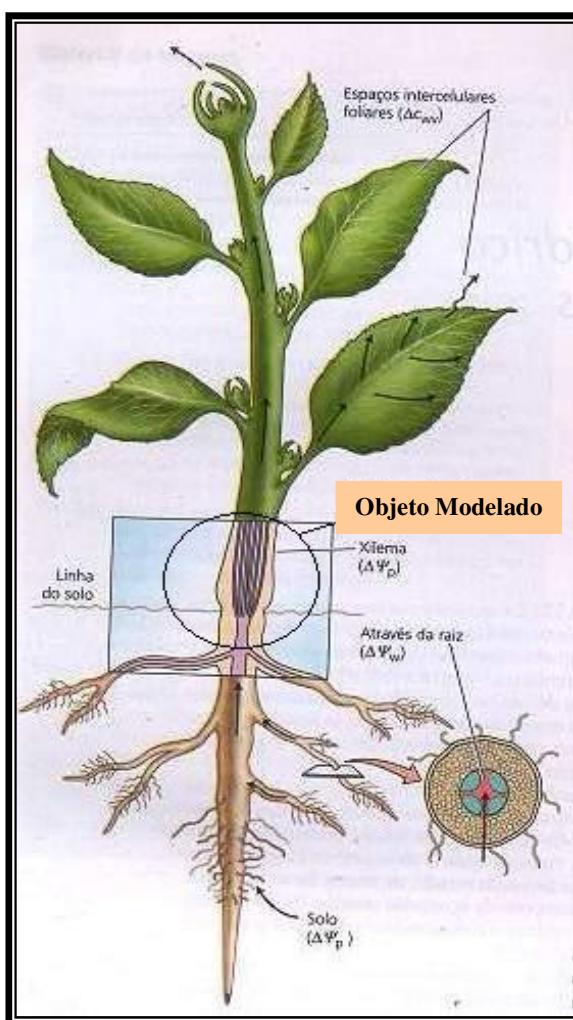


Figura 6.1 Vasos para transporte de Seiva (TAIZ, 2006)

A distribuição de seiva bruta ou inorgânica (água e sais minerais) é realizada pelos vasos de xilema ou lenho. A distribuição de seiva elaborada ou orgânica (água e açúcares) é realizada pelos vasos de floema ou líber. O presente estudo de caso, aborda no *software* de

Realidade Aumentada, o transporte de solutos pelos vasos de xilema ou lenho. A Figura 6.2 ilustra de forma ampliada o funcionamento do xilema.

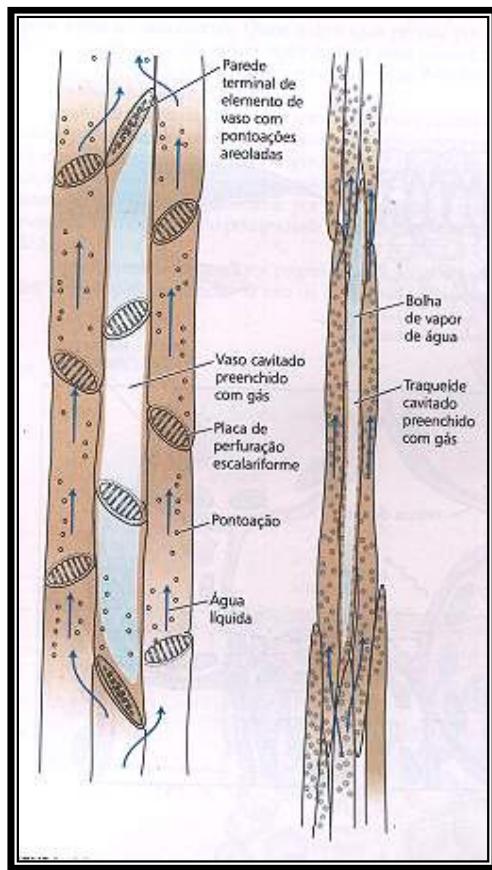


Figura 6.2 Vasos de Xilema (TAIZ, 2006)

### 6.3 Funcionamento do Sistema

#### 6.3.1. Funcionamento da Camada de Distribuição

Para o sistema **R.A. Distribuída** funcionar corretamente, é necessário ter instalado nos computadores que compõem a rede onde as interfaces de Realidade Aumentada, a Camada de Comunicação e de distribuição irão trabalhar, o ORB da Visibroker. Assim, se faz necessário inicializar o aplicativo **osagent** da Visibroker em todos os computadores, para isso, pode-se utilizar o *prompt* do MS-DOS, inserindo o comando: **osagent -c**.

Na inicialização do cliente, caso não haja servidor ativo, o primeiro cliente a ser inicializado deverá chamar a aplicação servidora, assim a aplicação servidora pode estar em qualquer computador de uma rede compartilhada e, ainda, ela não está vinculada a nenhuma interface de Realidade Aumentada. Não há nenhuma interfaceamento com a aplicação servidora, pois a mesma funciona apenas distribuindo as informações recebidas dos clientes.

Já a aplicação cliente, recebe as informações enviadas pelo servidor e grava em uma **Camada de Comunicação** para que a interface de Realidade Aumentada leia essas informações e faça a visualização do objeto virtual na cena gráfica. Por outro lado, a aplicação cliente também fica lendo as informações gravadas na **Camada de Comunicação** pela interface de Realidade Aumentada.

O **R.A. Distribuída** é composto por três camadas: uma interface de Realidade Aumentada, uma camada de comunicação e uma camada de distribuição. Assim a camada de interface realiza a visualização dos objetos virtuais na cena gráfica por meio do ARToolKit; a camada de comunicação faz a comunicação entre a interface de Realidade Aumentada e a camada de distribuição e, por fim; a camada de distribuição funciona como uma arquitetura baseada em Cliente/Servidor que faz a distribuição dos objetos virtuais informados pela interface de Realidade Aumentada.

Desta forma, a Figura 6.3 ilustra a aplicação de distribuição cliente.

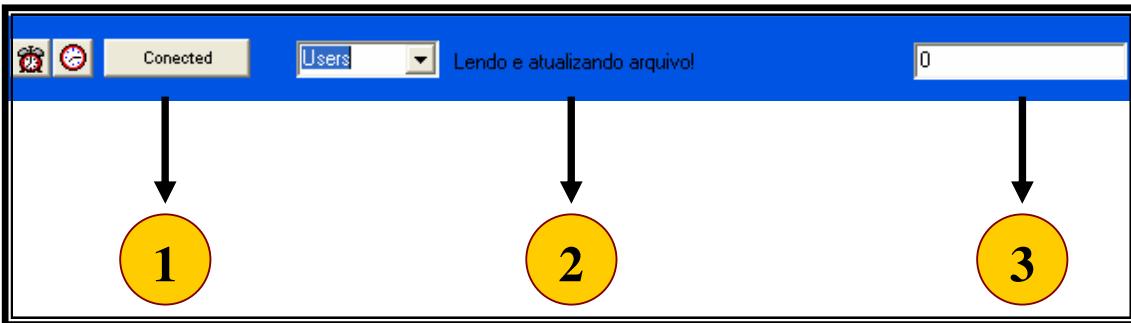


Figura 6.3 Interface da aplicação de distribuição – Cliente

Onde:

**1** – Mostra para o usuário se a aplicação está conectada, ou não, retorna dois valores: conectado ou desconectado.

**2** – Mostra informações sobre a leitura e gravação na camada de comunicação, caso haja algum erro na leitura, o tipo de erro é informado.

**3** – Mostra qual objeto está sendo lido e enviado para a aplicação de interface de Realidade Aumentada visualizar na cena gráfica.

A camada de comunicação resume-se a dois módulos, de entrada e de saída de informações para as aplicações distribuídas e para a interface de Realidade Aumentada.

### 6.3.2. Funcionamento da Interface de Realidade Aumentada

A aplicação de interface de Realidade Aumentada possui três menus visualizados na cena gráfica (sem a utilização de marcadores), cada um desses menus pode, nesta aplicação, possuir dois objetos virtuais. A Figura 6.4 ilustra a interface de Realidade Aumentada.

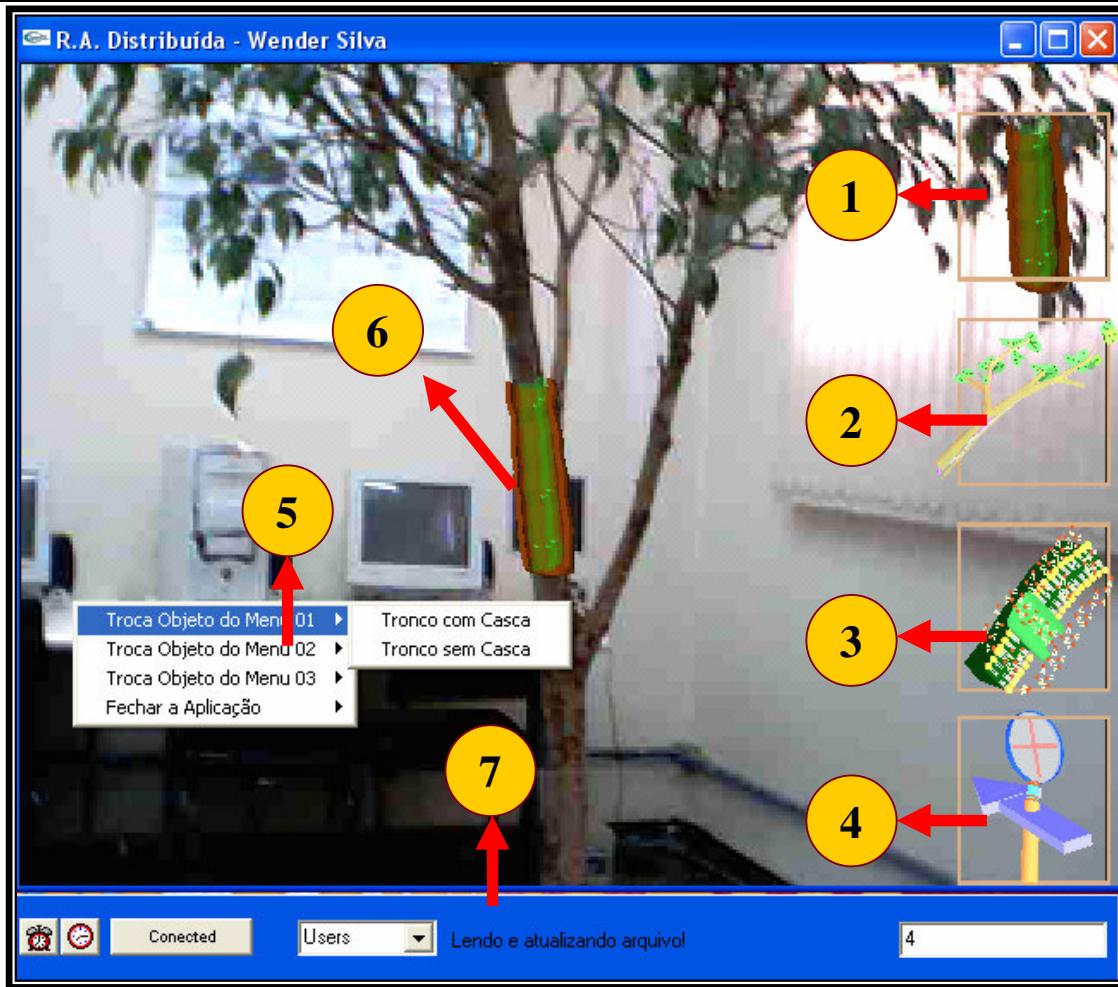


Figura 6.4 Interface da Aplicação de Interface de Realidade Aumentada

Onde:

**1** – Primeiro Menu: possui dois objetos virtuais, um tronco cortado com vasos de transporte de solutos no seu interior e, um objeto ilustrando apenas os vasos de transporte.

**2** – Segundo Menu: possui dois objetos virtuais, um galho transparente com os vasos de transporte de solutos no seu interior e, um galho sem os vasos e também sem transparência.

**3** – Terceiro Menu: possui um estroma com detalhamento do processo da fotossíntese e da evaporação de água no processo do transporte de solutos e, um estroma sem detalhamento.

**4** – Quarto Menu: possui interatividade por meio das transformações geométricas, trabalhando com colisão de marcadores. Para tanto, têm-se um marcador para cada tipo de transformada geométrica.

**5** – Menu Suspenso: possui as rotinas para trocar o objeto virtual de qualquer um dos três menus visualizados na cena gráfica em tempo real.

**6** – Objeto virtual tronco cortado sobreposto sobre a árvore real: esse tronco cortado ilustra o transporte de seiva (solutos) no interior de uma árvore. Esse transporte de solutos é baseado no estudo de caso apresentado nos itens *fisiologia vegetal e Transporte de solutos* descritos neste capítulo.

**7** – Camada de Distribuição da Aplicação: já comentado no item *funcionamento da camada de distribuição*.

Com a utilização de um marcador, pode-se instanciar qualquer um dos seis objetos virtuais contidos nos três menus e pode-se adicionar esses objetos a um outro marcador qualquer. Assim, ao se instanciar um objeto virtual de um marcador para o outro, é disparado um comando que faz a gravação do objeto em uma *camada de comunicação*. Por outro lado, a camada de distribuição faz a leitura da camada de comunicação e envia para o servidor as informações lidas. O servidor por sua vez, faz a distribuição para todos os clientes ativos na rede de computadores, esses clientes fazem a leitura ao receber as informações do servidor e gravam-nas na camada de comunicação. Desta forma, a interface de Realidade Aumentada também faz a leitura da camada de comunicação e faz a visualização do objeto virtual na cena gráfica. Assim, a Figura 6.5 ilustra a ação de instanciar objeto virtual no menu interativo.

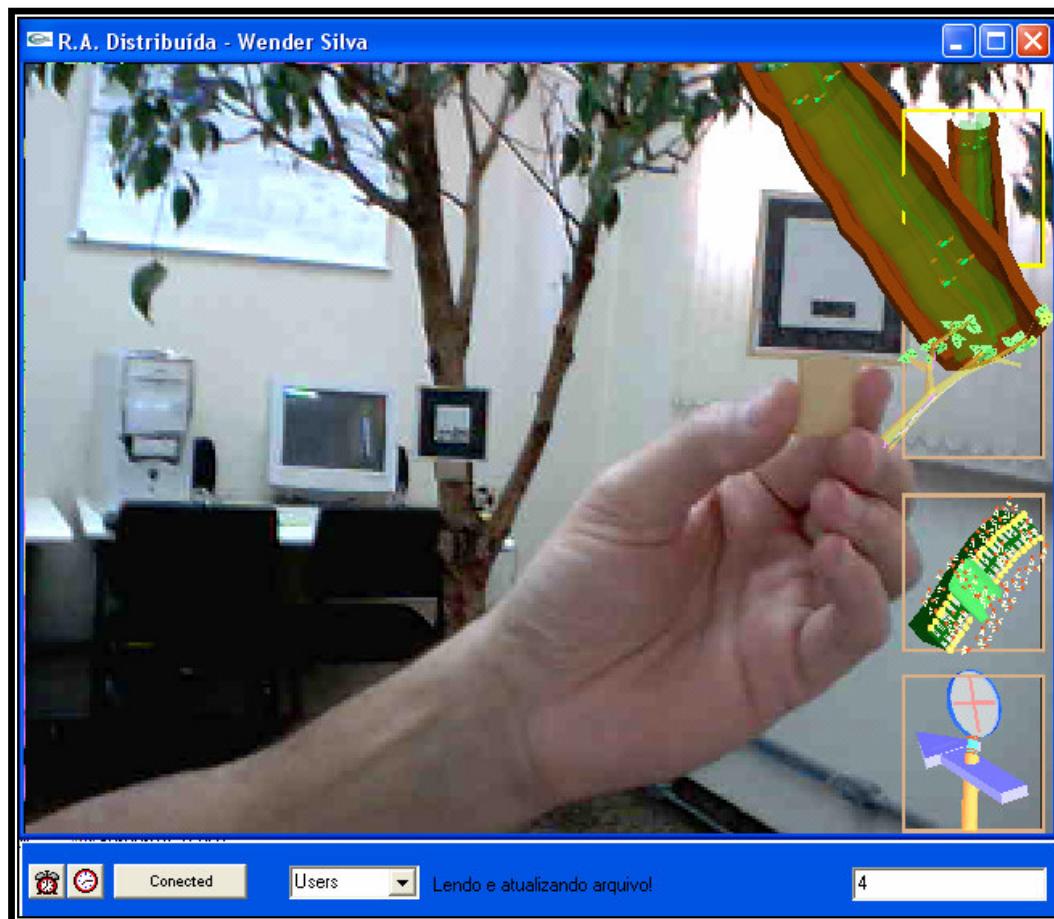


Figura 6.5 Interação: pegando objeto virtual no menu Interativo

Entende-se que um cliente de aplicação do **R.A Distribuída** é formado por três camadas: interface de Realidade Aumentada, Camada de Comunicação e Camada de Distribuição. Assim, nota-se que as aplicações **R.A Distribuída** são essencialmente clientes, e que necessitam de um cliente CORBA, e da Camada de Comunicação para realizar a distribuição, por outro lado, o servidor apenas distribui as informações recebidas pelos clientes CORBA. A **camada de comunicação** permite implementar a distribuição de outras interfaces de Realidade Aumentada baseadas no mesmo padrão deste trabalho, sem muito esforço de programação.

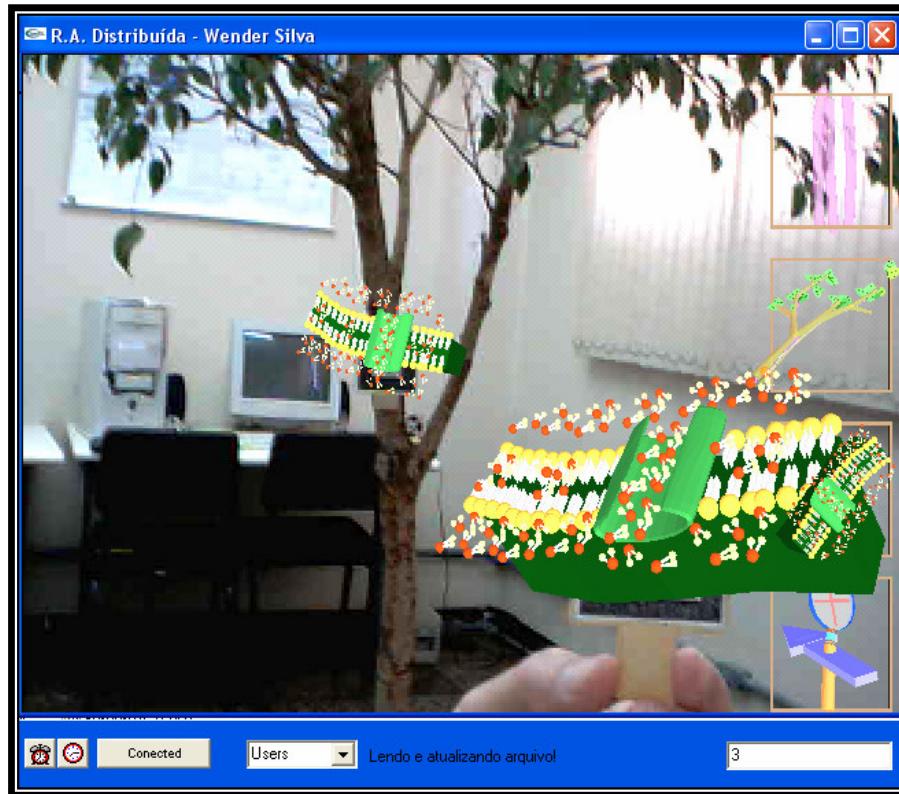


Figura 6.6 Interação: pegando objeto virtual no menu Interativo

Como ilustrado na Figura 6.5, ao se instanciar um objeto virtual em um dos três menus interativos, o parâmetro de referência do objeto virtual é gravado na camada de comunicação e, automaticamente a camada de distribuição faz a leitura dos parâmetros gravados e, informa para o usuário qual objeto virtual está sendo lido. Observe na Figura 6.5, o parâmetro do objeto virtual selecionado é o quatro. A Figura 6.6 ilustra a troca de parâmetros de distribuição por meio da troca de objetos virtuais no marcador “pá”. Assim, observa-se que o novo objeto virtual selecionado é o de número três.

Para a distribuição, os clientes devem estar com o marcador padrão de distribuição disposto na imagem capturada pelo sistema de Realidade Aumentada. Desta forma, ao identificar o marcador padrão, já previamente cadastrado e, também identificar que existe um

objeto virtual para ser visualizado, automaticamente o objeto virtual é mostrado pela interface de Realidade Aumentada sobre o marcador. As Figuras 6.7a e 6.7b ilustra a relação. Desta forma, a Figura 6.7a ilustra a visualização da árvore real com o objeto virtual “tronco cortado” por meio de uma webcam (*web1*), em um computador conectado à uma rede (*pc1*), estes equipamentos possuem um ponto de vista específico da árvore real.



Figura 6.7a Inicio da distribuição (*pc1* com *web1*)

Por outro lado, a Figura 6.7b faz a visualização por meio de uma outra webcam (*web2*) e, também por outro computador (*pc2*) conectado em uma rede, estes equipamentos possuem um ponto de vista diferente dos especificados pelo primeiro computador (*pc1*) e webcam (*web1*). Assim, a *web1* conectada no *pc1* faz a captura da cena e a interface de Realidade Aumentada desta aplicação **R.A. Distribuída** faz a visualização do objeto virtual tronco cortado no marcador que está posicionado na árvore real.

Desta forma, a *web2* conectada no *pc2* que possui uma visão diferente da árvore real e, que está funcionando com uma cópia da aplicação **R.A. Distribuída**, recebe o parâmetro de distribuição e visualiza o objeto recebido na cena gráfica por meio do marcador padrão que está posicionado na árvore. Sabe-se que este objeto virtual foi selecionado pelo usuário do *pc1*. Assim, todos os Clientes **R.A. Distribuída** que estiverem ativos na rede, e que estiverem capturando em sua interface de Realidade Aumentada o marcador padrão cadastrado, deveram realizar a visualização do objeto virtual selecionado pelo *pc1*. A Figura 6.7c ilustra uma terceira aplicação funcionando. Esta terceira aplicação não está apontando para uma árvore,

porém está visualizando o objeto virtual por meio do marcador padrão definido para esta aplicação.



Figura 6.7b Distribuição do objeto virtual na árvore com outro visão (pc2 com web2)



Figura 6.7c Distribuição do objeto virtual com outro visão (pc3 com web3)

Por fim as Figuras 6.9 e 6.10 ilustram a transferência de um objeto virtual de um marcador, chamado de “pá” para um outro marcador já previamente cadastrado.

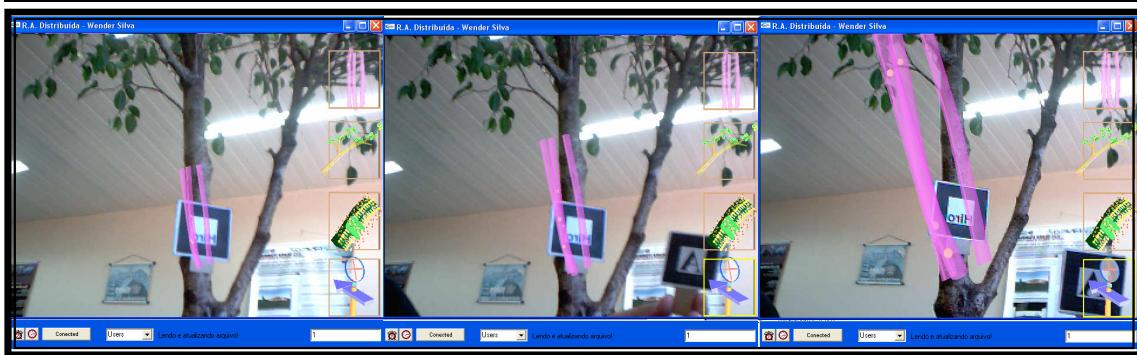


Figura 6.8 Aumentando a escala dos objetos virtuais com a utilização de marcador

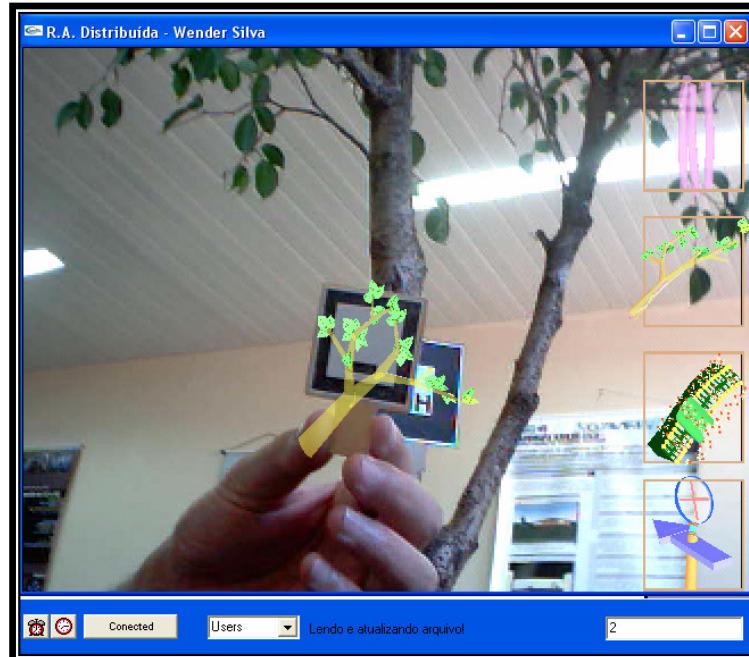


Figura 6.9 Transferência de um objeto virtual de um marcador para outro

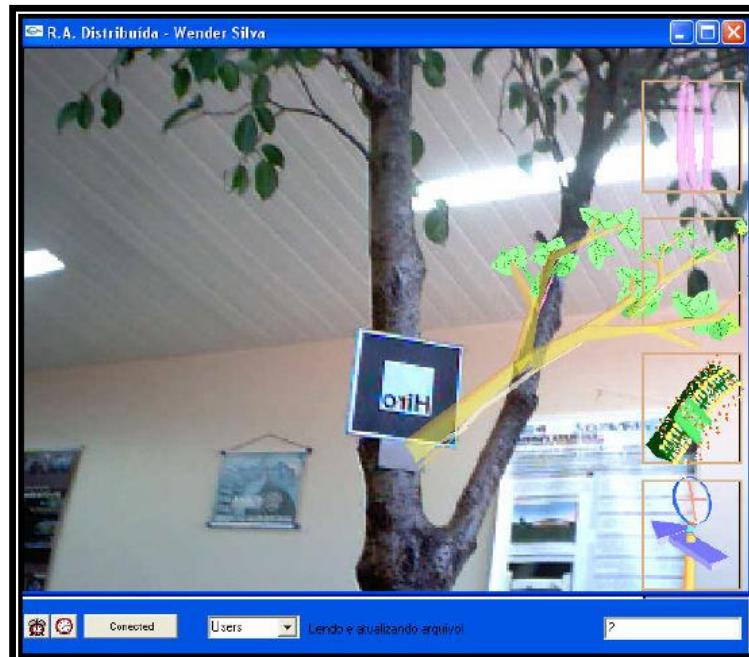


Figura 6.10 Finalização da transferência do objeto de um marcador para outro

## 6.4 Distribuição da Informação

Segundo Ribeiro (2006) qualquer que seja a atividade ou fenômeno simulado nos ambientes virtuais, sejam eles de Realidade Virtual sejam eles de Realidade Aumentada, os mesmos acontecem apenas seqüencialmente. Ou seja, não é possível, por exemplo, iniciar dois processos ao mesmo tempo ao mesmo tempo em ambientes virtuais diferentes.

Assim, portanto, para cada tipo de interação em cada ambiente é associado um parâmetro que possui valor que varia de “*zero*” a “*seis*” de acordo com cada objeto virtual a ser visualizado e distribuído. Sempre que um ambiente é iniciado na rede, o mesmo procura por um servidor. Caso não o encontre, o ambiente começa a interagir com o usuário a partir do parâmetro zero, ou seja, não há nenhum objeto virtual para ser distribuído na rede de computadores. A responsabilidade do armazenamento destes parâmetros é do primeiro trio (interface de Realidade Aumentada, Camada de Comunicação e Camada de Distribuição) de aplicações que for iniciado na rede.

A arquitetura proposta para este trabalho suporta a comunicação de *n* ambientes que se comunicam ou interagem com *m* cópias do ambiente de *R.A Distribuída*. Diante disso, após a ativação do primeiro trio de aplicações, sempre que houver outra ativação, a aplicação irá iniciar com parâmetros de visualização de acordo com o valor armazenado no servidor. Sempre que houver alguma interação, o valor do parâmetro do ambiente será modificado e enviado ao servidor, que distribui esta informação para todas as aplicações *R.A. Distribuída* na rede.

Além do parâmetro inicial, valor zero, os ambientes possuem os possíveis parâmetros:

**Parâmetro 1:** objeto virtual vasos de condução de solutos.

**Parâmetro 2:** objeto virtual galho transparente com visualização interna dos vasos de transporte de solutos.

**Parâmetro 3:** objeto virtual estroma detalhado com a representação do fluxo de água pela aguaporina.

**Parâmetro 4:** objeto virtual tronco cortado com os vasos de transporte de solutos internamente.

**Parâmetro 5:** objeto virtual galho sem transparência para visualização e complemento da árvore real.

**Parâmetro 6:** objeto virtual estroma simples sem detalhes de representação do fluxo de água, apenas para visualização do formato de um estroma.

## 6.5 Considerações Finais

Este capítulo apresentou o sistema **R.A. Distribuída**, detalhando o funcionamento dos ambientes virtuais no contexto do estudo de caso da fisiologia vegetal, e explicitou como e qual informação é distribuída.

Cabe ressaltar que os detalhes apresentados neste capítulo foram expostos por meio de figuras, que mostram apenas uma instância de cada ambiente, porém a arquitetura suporta cópia de várias ambientes em uma rede.

Baseado no estudo comparativo dos sistemas descritos nos trabalhos relacionados, percebe-se a contribuição deste trabalho, descrito na tabela 6.1, onde faz-se uma comparação entre os trabalhos e este protótipo. O software **R.A. Distribuída** difere-se dos demais trabalhos verificados nos seguintes aspectos:

- Criação de Interfaces Interativas em R.A, onde pode-se navegar no mundo virtual por meio de teclado ou de marcadores.
- Camada de distribuição, permitindo a integração à interface de Realidade Aumentada e fazendo a distribuição das interações realizadas na interface. Essa camada é descrita em forma de uma arquitetura de distribuição de objetos em Realidade Aumentada.

No próximo capítulo, apresenta-se a análise dos resultados obtidos.

Tabela 6.1. Realidade Virtual e Aumentada Distribuída – Comparação e Contribuições desta dissertação

Realidade Virtual e Aumentada Distribuída										
Trabalhos Avaliados	Interações em R.A tempo de execução por meio de:						Distribuição realizada por meio de:			Aplicabilidade na Educação
	Colisão	Marcadores	Troca de Objetos	Hardware		Menus de Intereração	Arquitetura CORBA	Camada de Comunicação	Modelo Cliente/Servidor	
 Item Contemplado  Item não Contemplado				Teclado	Mouse					
Distributed Applications for Collaborative Augmented Reality										
Ambientes Colaborativos com Realidade Aumentada										
Distributed Augmented Chess System										
Interactive Entertainment Systems Using Tangible Cubes										
Visualização de Dados Relativos a Redes de Computadores usando Realidade Virtual e Aumentada										
Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares										
R.A. DISTRIBUÍDA										

# CAPÍTULO VII

## 7. AVALIAÇÃO E RESULTADOS

### 7.1 Introdução

Neste capítulo é apresentada uma análise do protótipo descrito nos Capítulos V e VI. Esta análise teve como objetivo demonstrar a viabilidade, a eficácia e eficiência do protótipo. Assim, para tal são abordadas questões sobre avaliação do sistema, por meio de testes básicos sobre usabilidade. A análise realizada abrange, sobretudo, os seguintes aspectos:

- a) A obtenção de uma melhor condição de trabalho em uma interface de Realidade Aumentada.
- b) A obtenção de um ambiente interativo, por meio dos aspectos de distribuição da aplicação.
- c) Verificar os aspectos educacionais do ambiente proposto.

Desta forma, para realizar esta análise foi necessário à construção de uma metodologia específica baseada nos preceitos da ISONORM para fazer tal verificação.

### 7.2. Avaliação do Sistema R.A Distribuída

Para a avaliação e validação da aplicação **R.A. Distribuída** foi elaborado, com base em preceitos da ISONORM 9241-10 Usability (PRUGMPER, 1999) (conjunto de normas que permite avaliar a capacidade de um sistema interativo oferecer ao usuário a possibilidade de realizar tarefas de maneira eficaz e agradável) um questionário em forma de um **checklist** onde os usuários puderam informar suas opiniões. Este questionário teve por objetivo verificar questões sobre usabilidade do sistema desenvolvido e, ainda quando da concepção do questionário utilizou-se conceitos pedagógicos e, de software educacional (CAMPOS, 1998). Assim, o questionário foi aplicado a uma amostra de 80 indivíduos em potencial. Foram escolhidos aleatoriamente alunos e professores do curso de informática de uma Instituição de Ensino Superior da cidade de Itumbiara - Goiás. Ainda, foram abordados pedagogos do curso de Pedagogia e, também professores do curso de Biologia. Assim, a amostra compõe-se de: 10 professores pedagogos, 10 professores biólogos, 10 professores do curso de informática e 50 alunos dos cursos de informática. Todos foram submetidos ao software e responderam ao mesmo questionário. Desta forma, o questionário encontra-se no anexo desta dissertação.

### 7.2.1. Latência da Comunicação e Escalabilidade

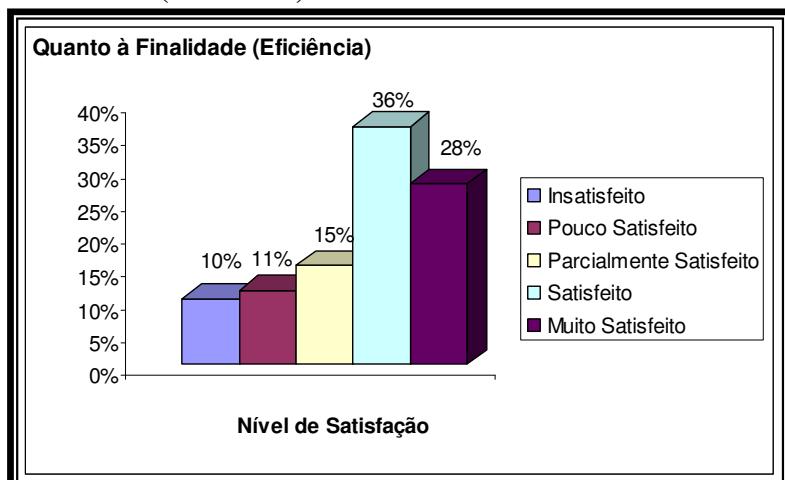
Do ponto de vista do ambiente virtual distribuído, é a latência que controla a natureza interativa e dinâmica do sistema. Se o ambiente distribuído existe para emular o mundo real, deve operar em termos da percepção humana (MACEDONIA, 2005). Segundo Ribeiro (2006), sistemas que envolvem operadores humanos devem entregar pacotes com a mínima latência e gerar gráficos texturizados 3D a uma freqüência de 30 a 60 Hz, para garantir a ilusão de realidade. Este desafio torna-se maior em sistemas que utilizam redes a longa distância. A latência da rede pode ser reduzida, até certo ponto, por meio da utilização de enlaces (*links*) dedicados. Todavia, uma maior largura de banda não é necessariamente a melhor solução. A escalabilidade é interpretada como sendo a capacidade do sistema em aceitar novos clientes, numa mesma máquina ou em máquinas diferentes.

Latência e escalabilidade são fatores decisivos na avaliação de sistemas distribuídos, porém, este trabalho preocupou-se em apresentar um protótipo de RA distribuída que fosse suficiente para suportar o estudo de caso apresentado (*fisiologia vegetal*) como instrumento de melhoria do aprendizado, ou seja, um protótipo voltado à educação. Recomenda-se fortemente o uso destes instrumentos de análise em trabalhos futuros que tenham foco na avaliação de eficiência da arquitetura.

### 7.2.2. Resultados da avaliação do sistema

O questionário é dividido em quatorze questões que abrangem aspectos tecnológicos, pedagógicos, de usabilidade e de conteúdo específico proposto no estudo de caso desta dissertação. Desta forma, descreve-se os resultados relativos ao questionário em forma de gráficos.

#### I. Quanto à Finalidade (Eficiência):



**Figura 7.1. Gráfico: Finalidade e eficiência.**

## II. Quanto à Interface (Usabilidade): entendimentos dos comandos

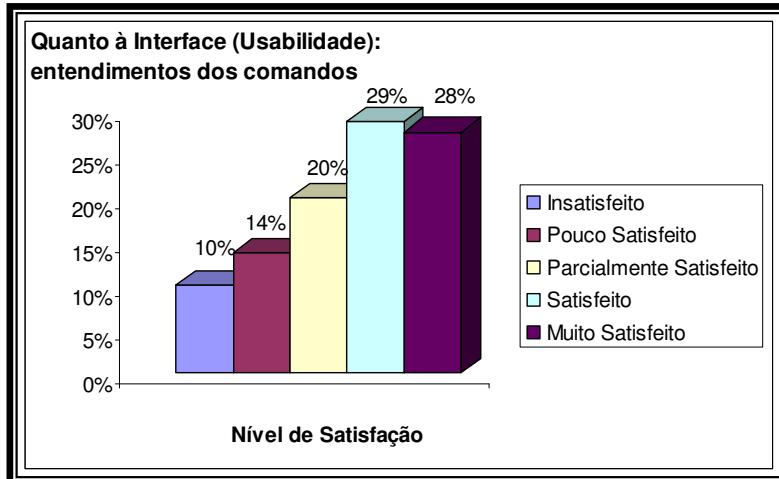


Figura 7.2. Gráfico: Interface (usabilidade)

No gráfico da Figura 7.2 observa-se que alguns usuários consideraram-se **insatisfeitos** e não justificaram as suas respostas. A maioria das pessoas elogiaram o sistema, considerando que sua interface é de fácil entendimento em relação aos comandos. Já uma pequena porcentagem dos avaliadores, citou a necessidade de um prévio conhecimento de Informática para a execução do mesmo.

## III. Quanto à facilidade de uso (Funcionalidade)

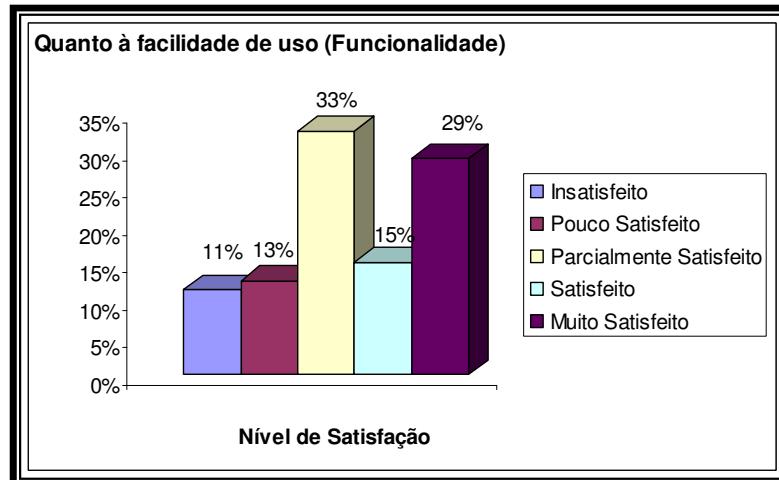


Figura 7.3. Gráfico: Facilidade de uso

Neste item, uma porcentagem de entrevistados relatou que o *software* necessita de um prévio conhecimento em Informática, considerando que muitos usuários poderão ter dificuldades em executar os comandos, exigindo antes da execução a apresentação de algumas informações adicionais para operacionalização do sistema. Assim o item **satisfeto** ficou abaixo do esperado, isto se justifica pela divisão de opinião dos entrevistados, onde em sua maioria dividiram-se em **parcialmente satisfetos e muito satisfetos**.

#### IV. Quanto ao nível de interação com os objetos disponíveis na cena

Neste item alguns dos entrevistados consideravam-se **pouco satisfeitos ou insatisfeitos** com a interatividade do protótipo apresentado, isto justifica-se pela dificuldade de entendimento da interface de Realidade Aumentada face às interfaces proporcionadas pelos sistemas *desktops* convencionais. Os outros entrevistados relataram à falta de interatividade (transformações geométricas) por meio de menus interativos e marcadores, entendendo que por se tratar de uma aplicação de Realidade Aumentada deveria estar presente esta possibilidade. Isto justifica a baixa porcentagem no item **muito satisfeito**.

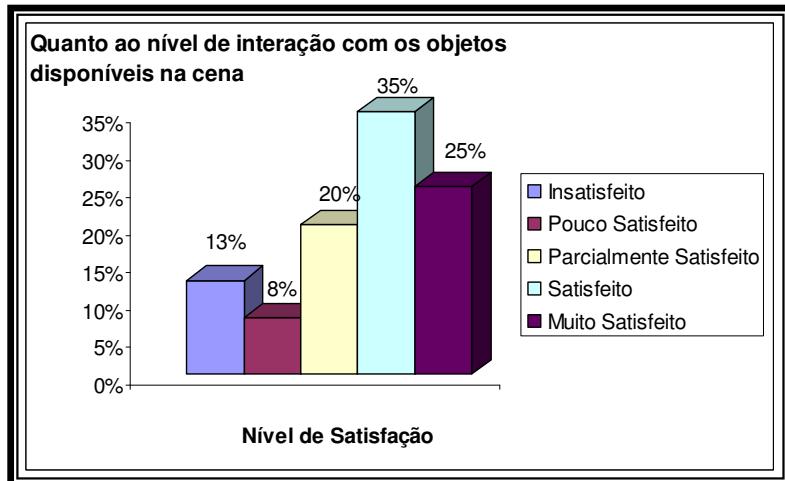


Figura 7.4. Gráfico: Nível de Interação na cena

#### V. Quanto à manipulação dos objetos virtuais na cena (facilidade de uso)

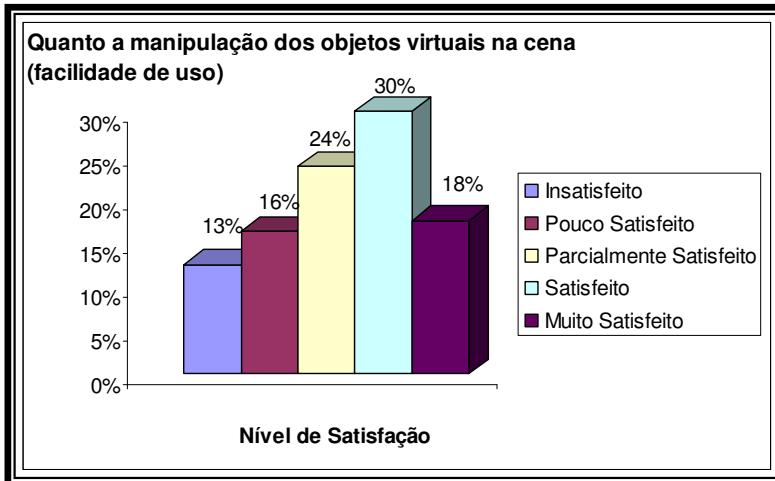


Figura 7.5. Gráfico: Manipulação dos objetos virtuais na cena

O gráfico da Figura 7.5 ilustra que uma porcentagem dos entrevistados que não se sentiram a vontade em relação à facilidade de uso e manipulação dos objetos virtuais. Estes comentaram que a manipulação dos objetos virtuais na cena é de fácil entendimento, porém, para as pessoas que não possuem acesso à informática ou que não tiverem um treinamento

prévio, torna-se uma tarefa difícil, selecionar os objetos virtuais nos menus interativos. Assim, entende-se que existe um abismo entre entender a interface e manusear o aplicativo

## VI. Quanto à manipulação dos menus interativos (facilidade de uso)

No Gráfico da Figura 7.6, a maioria das pessoas sentiram-se **satisfeitas** quanto a manipulação dos menus, porém houve alguns comentários relatando a dificuldade de manusear o marcador para instanciar os objetos virtuais. Uma pequena porcentagem de entrevistados sentiram-se **muito satisfeitos** com a manipulação dos menus interativos, isto remete à questão da falta de treinamento, ou ainda, da dificuldade de adaptação para uma nova interface.

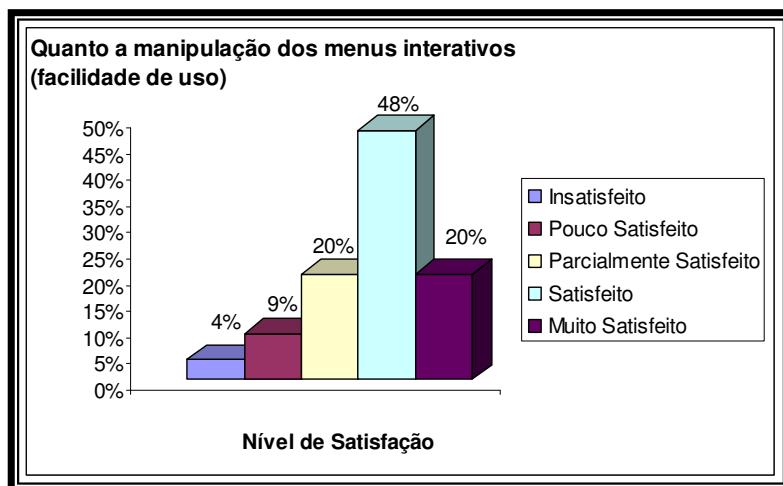


Figura 7.6. Gráfico: Manipulação dos Menus Interativos

## VII. Quanto à colaboração entre os usuários (é perceptível)

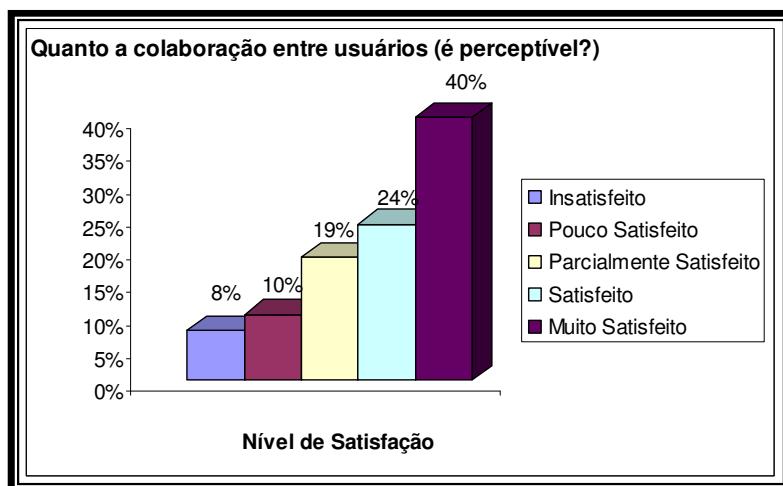


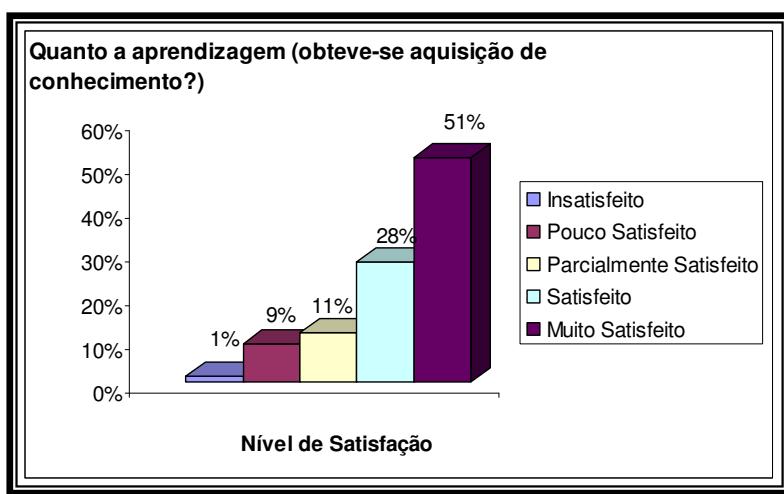
Figura 7.7. Gráfico: Colaboração entre usuários

No gráfico da Figura 7.7, a maioria dos usuários (alunos do curso de informática) sentiram-se **satisfeitos** com a distribuição realizada pelo software e, realmente entenderam o propósito da aplicação, deixando inclusive algumas sugestões. Uma das sugestões é bastante

interessante, e diz respeito a fazer a referência dos objetos virtuais que devem ser distribuídos por meio de um marcador para que o usuário não ficassem perdido na cena e, para tentar gerenciar melhor onde o objeto virtual deve ficar na cena gráfica dos clientes.

### **VIII. Quanto à aprendizagem (obteve-se aquisição de conhecimento?)**

O gráfico da Figura 7.8 ilustra que a maioria dos usuários ficaram **muito satisfeitos** com a questão da aprendizagem proporcionada pela aplicação software. Alguns comentários relatam que o estudo de caso está contemplado. Porém os profissionais da biologia entenderam ser necessário um melhor detalhamento dos objetos virtuais e, ainda, uma interface que possua mais informações sobre os objetos virtuais visualizados em cena.



**Figura 7.8. Gráfico: Aquisição de conhecimento**

Já os profissionais da pedagogia comentaram que o software realmente possui características para proporcionar a aquisição de conhecimento por parte dos usuários, porém, algumas discussões ainda são necessárias para que o mesmo possa realmente ser caracterizado como um software educacional.

### **IX. Quanto aos objetos virtuais na cena (permite a concepção da experiência proposta)**

No gráfico da Figura 7.9, a maioria dos entrevistados consideraram-se **muito satisfeitos** em relação à concepção da experiência proposta. Não houve comentários por parte dos alunos de graduação. Já os professores do curso de pedagogia relataram que é boa a relação entre o que é proposto, e que o software realiza, porém, ainda há a necessidade de melhorias nas questões de implementação para ser aplicado à educação.

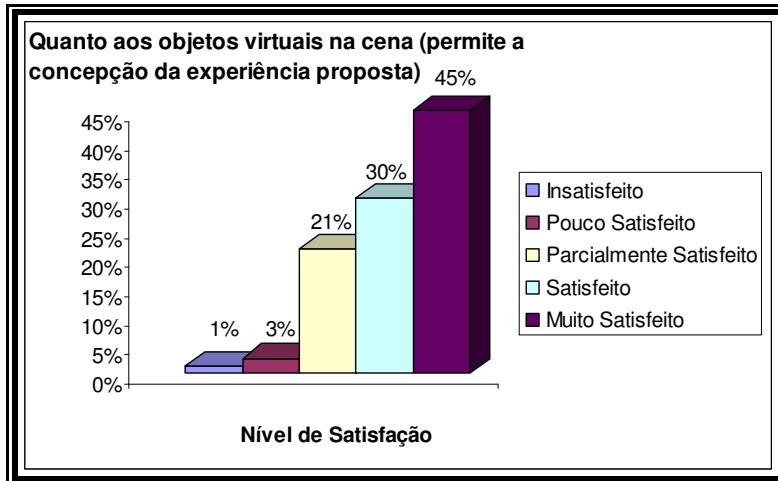


Figura 7.9. Gráfico: Concepção da experiência proposta

#### X. Quanto ao número de funções existentes no sistema

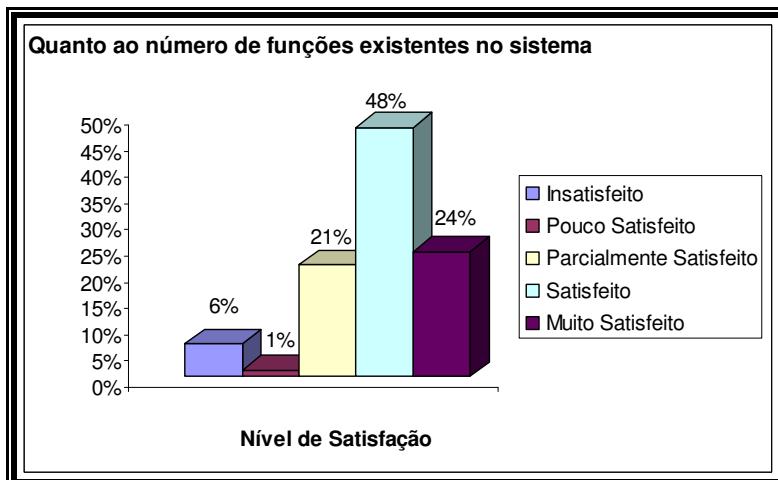


Figura 7.10. Gráfico: Funções Existentes no sistema

O gráfico da Figura 7.10 relata que a maioria das pessoas sentiram-se **satisfaitas** com a quantidade de funções existentes no sistema, porém relataram a necessidade de mais objetos virtuais em cena para enriquecer ainda mais o estudo de caso. Em relação às outras avaliações não houve comentários.

#### XI. Quanto a confiança de que o sistema já executou o que foi pedido

No gráfico da Figura 7.11, nota-se que a maioria dos entrevistados sentiram-se **satisfaitos** com a confiabilidade do software. Neste item não houve nenhum comentário. É importante ressaltar que, devido ao baixo desempenho do processamento dos computadores utilizados para a realização dos testes, às vezes o software não respondia adequadamente as chamadas realizadas por meio de colisão de objetos.

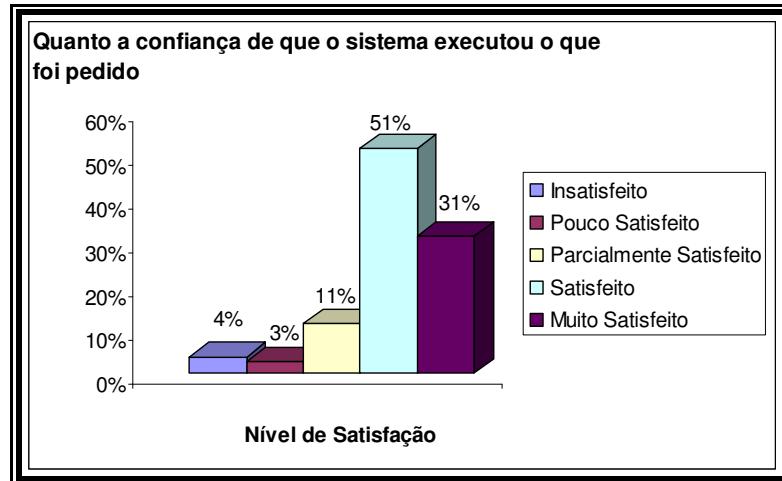


Figura 7.11. Gráfico: Confiança no sistema

## XII. Quanto o número de passos necessário para a realização de uma atividade

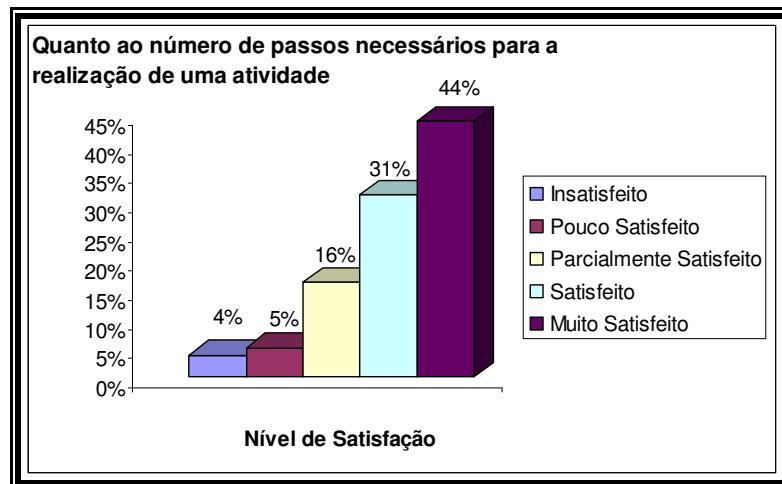
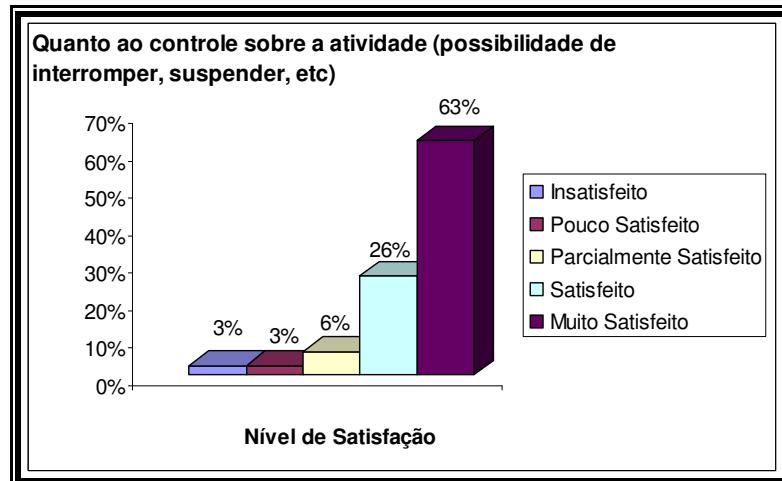


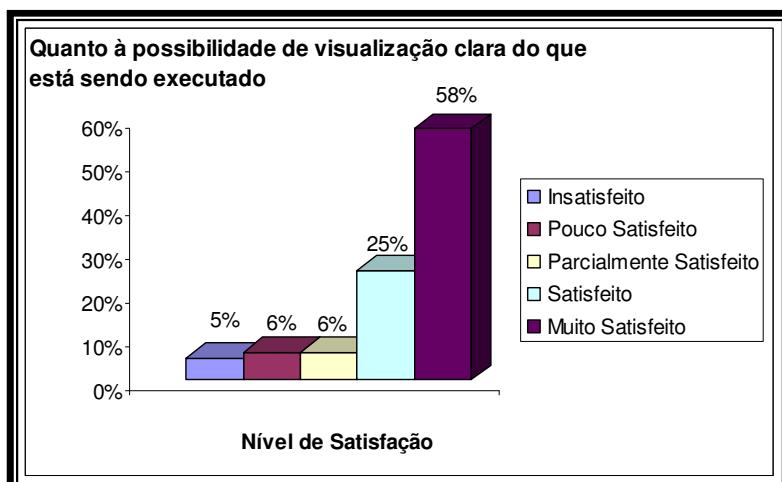
Figura 7.12. Gráfico: Número de passos necessários para realizar uma atividade

No gráfico da Figura 7.12, a maioria dos entrevistados sentiram-se **muito satisfeitos** com a quantidade de passos necessários para a realização de uma atividade no software. Neste item não houve comentários. Entende-se que a quantidade de passos para a interação na interface de Realidade Aumentada é pequena e objetiva e, a interface de distribuição realiza todo o processo automaticamente.

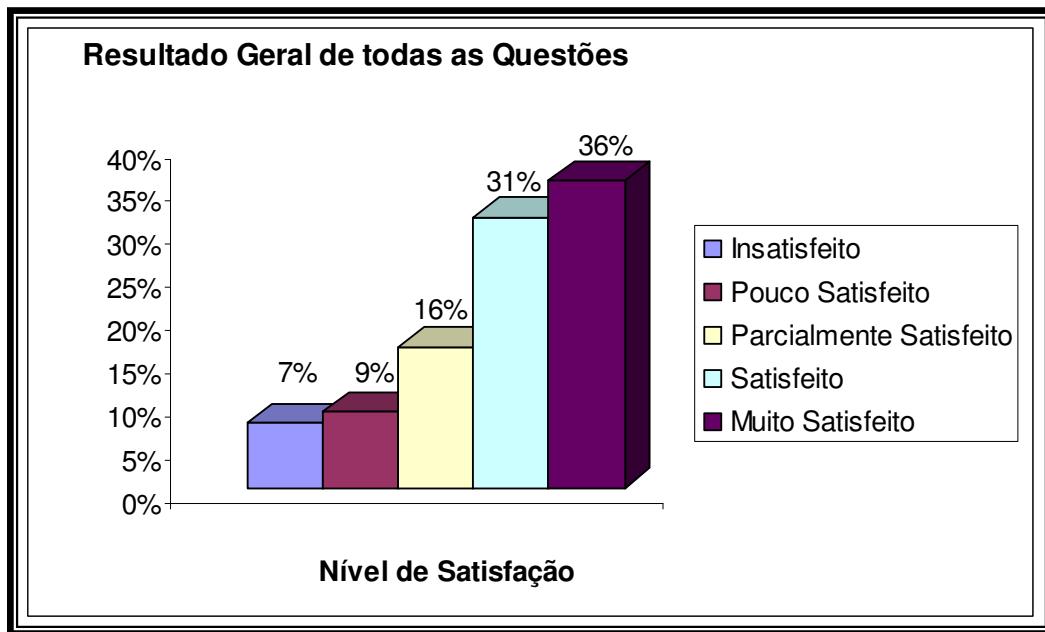
**XIII. Quanto o controle sobre a atividade (possibilidade de interromper, suspender, etc)****Figura 7.13. Gráfico: Controle sobre as atividades****XIV. Quanto à possibilidade de visualização clara do que está sendo executado**

No gráfico da Figura 7.14 a maioria das pessoas sentiram-se **muito satisfeitas** com a visualização clara do que está ocorrendo dentro da cena de Realidade Aumentada.

Porém, alguns comentários realizados pelos profissionais da biologia relatam à necessidade de um nível maior de detalhamento dos objetos virtuais. Entende-se que o nível de processamento para ambientes de Realidade Aumentada dos equipamentos utilizados para desenvolver esta aplicação não permitiu esse nível maior de detalhamento dos objetos virtuais.

**Figura 7.14. Gráfico: Visualização clara do que está sendo executado**

## Visão Geral do Sistema por meio dos questionários aplicados



**Figura 7.15. Gráfico: Visão Geral do Sistema**

No gráfico da Figura 7.15 ilustra a visão geral do resultado da aplicação dos questionários aplicados. Desta forma, entende-se que por meio destes questionários, é possível afirmar que o software **R.A Distribuída** apresentado nesta dissertação é viável e, que pode ser utilizado em aplicações educacionais de forma geral, porém, cabe uma análise mais aprofundada das questões pedagógicas/educacionais.

### 7.3. Recursos não implementados

Os tópicos a seguir mostram algumas limitações do sistema.

- ✓ Não há possibilidade de distribuição de dois objetos virtuais ao mesmo tempo.
- ✓ Os objetos virtuais são inseridos manualmente pelos usuários.
- ✓ A arquitetura de distribuição não permite a comunicação cliente/cliente.
- ✓ A interface de Realidade Aumentada não suporta equipamentos de HMD.

### 7.4. Considerações Finais

Este capítulo apresentou a avaliação em relação aos testes com usuários, mostrando os resultados em forma de gráficos, e por fim, faz-se um relato sobre as limitações da aplicação. Apesar de alguns recursos não terem sido implementados o sistema atendeu os objetivos propostos.

Assim, dentro deste aspecto, algumas sugestões em relação às análises realizadas, foram colocadas em prática, tal como:

- ✓ **Quanto à colaboração entre os usuários (é perceptível):** Atendendo às sugestões solicitadas neste item, o software foi alterado. Assim, o objeto virtual distribuído é visualizado em um marcador previamente cadastrado. Desta forma, o objeto virtual pode ser levado para qualquer ponto na tela gráfica do sistema de Realidade Aumentada. Antes desta implementação, a visualização dos objetos virtuais era realizada no mesmo ponto onde foi concebida os parâmetros de distribuição.
- ✓ **Quanto ao nível de interação com os objetos disponíveis na cena:** Atendendo às sugestões solicitadas neste item, o software foi alterado. Assim, o objeto virtual pode sofrer todas as transformações geométricas por meio de um menu interativo, utilizando-se de marcadores, ou seja, foi criado um marcador para cada transformada geométrica. Antes as transformações geométricas eram realizadas apenas por meio de teclado.
- ✓ **Quanto aos objetos virtuais na cena:** Atendendo às sugestões solicitadas neste item, o software foi alterado. Assim, criou-se a possibilidade de escolher dois objetos virtuais por menu interativo. Desta forma, por meio de um menu suspenso, é possível a escolha do objeto virtual a ser visualizado na cena gráfica e posteriormente ser selecionado. Antes havia a possibilidade de apenas um objeto virtual por menu.
- ✓ **Camada de Comunicação:** Foi necessário a criação de uma camada de comunicação para fazer a comunicação entre a Interface de Realidade Aumentada e a Camada de Distribuição.

# CAPÍTULO VIII

## 8. CONCLUSÕES E TRABALHOS FUTUROS

### 8.1 Introdução

Este capítulo tem como objetivo apresentar os principais pontos estudados nesta dissertação, relacionar os possíveis trabalhos futuros advindos desta pesquisa e avaliar a principal contribuição deste trabalho para a área científica.

### 8.2 Conclusões

Durante esta pesquisa, constatou-se que existem diversos ambientes virtuais distribuídos. Porém, a maioria desses ambientes apresenta foco na distribuição e comunicação de réplicas e, ainda, a maioria dos ambientes é em Realidade Virtual. Ainda, um fator de grande relevância observado é a falta de flexibilidade dos objetos virtuais em relação à sua interface e em relação a distribuição destes objetos.

Este trabalho apresenta um sistema com grande vocação educacional que usa Realidade Aumentada de forma distribuída permitindo que vários usuários manipulem ao mesmo tempo objetos virtuais dentro de cenários reais capturados por câmeras de vídeo. Sendo assim, foi apresentada uma arquitetura para o sistema proposto e implementado um estudo de caso que apresenta o processo de transporte de solutos (seiva). Para tal, foram criadas três camadas de software para proporcionar um único ambiente de Realidade Aumentada Distribuída.

Desta forma, a Camada de Comunicação é baseada em módulos e faz apenas a comunicação entre a camada de interface em Realidade Aumentada e a Camada de Distribuição.

Como conclusões da implementação da arquitetura, pode-se citar que:

- ✓ A plataforma CORBA 2.0 se mostrou capaz para a realização da distribuição de ambientes de Realidade Aumentada.
- ✓ A interface de Realidade Aumentada implementada proporciona um bom nível de interatividade e, possibilita a distribuição de objetos virtuais.

#### 8.2.1 Contribuições do Trabalho

Uma das contribuições deste estudo refere-se à construção de uma arquitetura para

desenvolvimento de um Ambiente Distribuído de Realidade Aumentada, que seja compatível com a metodologia proposta. Esta contribuição pode ser melhor detalhada pelos tópicos que estão ordenados de acordo com a evolução do estudo:

**1** – No levantamento efetuado no decorrer deste trabalho, verificou-se que, até o momento da elaboração desta dissertação, poucos AVD's era propriamente dito uma distribuição de ambientes virtuais de Realidade Aumentada por meio da arquitetura CORBA.

**2** – Este trabalho veio ressaltar outros estudos realizados a eficácia dos ambientes de Realidade Aumentada e Distribuída na criação de ferramentas educacionais.

**3** – O protótipo de interface interativa de Realidade Aumentada construído também representa uma contribuição, podendo ser utilizados principalmente como ferramenta de ensino.

**4** – Outra contribuição resulta da construção de uma aplicação que pode ser reutilizada para distribuir qualquer tipo de ambiente virtual ou de Realidade Aumentada, desde se utilize os mesmos parâmetros e metodologias utilizadas para a construção deste protótipo. Assim, sem muito esforço computacional e, sem alteração do código da camada de distribuição, pode-se utilizar a **Camada de Comunicação** por meio do ARToolKit ou de outro software de R. A. para distribuir objetos virtuais em uma rede de computadores, bastando para isso, implementar apenas as rotinas para gravação e leitura dos parâmetros de distribuição na camada de comunicação.

Percebe-se que não há diferenças na distribuição entre ambientes de Realidade Virtual e Realidade Aumentada. Entende-se e constata-se por meio desta dissertação que a diferença básica está na implementação das interfaces de Realidade Aumentada e Realidade Virtual, pois o que é distribuído são passagens de parâmetros relativos aos objetos virtuais ou à interações que poderão ser realizados nas interfaces de R.A.

Neste contexto, uma implementação de interface bem feita, resulta em um ambiente virtual muito mais interativo. Assim, essas interatividades podem ser distribuídas em uma rede de computadores em forma de parâmetros. Esses parâmetros podem referir-se a objetos virtuais, transformações geométricas ou a qualquer outro aspecto relacionado aos ambientes virtuais. A distribuição de ambientes virtuais de Realidade Aumentada, quando trabalhada corretamente, com todas as requisições e restrições de usuários para inserção e manipulação de objetos virtuais na cena, pode proporcionar um ambiente altamente interativo e colaborativo. Assim, entende-se que a principal contribuição deste trabalho é a junção de um Pipeline de interações da interface de Realidade Aumentada com a Arquitetura de Distribuição feita por meio de uma **Camada de Distribuição**. Assim, pode-se proporcionar

um ambiente virtual de Realidade Aumentada Distribuído, onde as camadas de interface de Realidade Aumentada e a Aplicação de Distribuição são conectadas por uma Camada de Comunicação.

### 8.3. Trabalhos Futuros

De acordo com a evolução deste trabalho, alguns aspectos foram considerados importantes na consolidação de um produto final com todos os requisitos necessários para utilização do mesmo, tanto no meio acadêmico/científico quanto comercial. O objetivo deste trabalho é validar um aspecto específico, portanto não tem pretensão de resolver todos os problemas ou necessidades advindas desta pesquisa, possibilitando a este estudo ser referência para outros trabalhos futuros. Desta forma destacam-se algumas contribuições futuras:

- 1** – Realizar testes em relação à latência e escalabilidade do protótipo.
- 2** – Implementar a utilização de dispositivo HMD.
- 3** – Estabelecer novos estudos de casos baseados em ambientes para treinamento.
- 4** – Implementar o reconhecimento de áreas diversas como forma de marcador, reconhecer uma caneta como um marcador, por exemplo.
- 5** – Fazer a integração de ambientes de Realidade Virtual com Realidade Aumentada.
- 6** – Criar uma arquitetura que permita implementar a distribuição dos ambientes virtuais pela internet, assim, o usuário pode fazer download o software e fazer colaboração com usuários em todo o mundo.
- 7** – Implementar a arquitetura de comunicação cliente-para-cliente, sem a passagem da mensagem pelo servidor.
- 8** – Testar a aplicação com outras arquiteturas de distribuição e, por fim fazer uma comparação entre as mesmas.
- 9** – Implementar a inserção de um objeto virtual novo, não cadastrado, em tempo real, sem a necessidade de re-iniciar a aplicação.
- 10** – Criar ambientes virtuais utilizando o OpenSceneGraph.

Desta forma, criou-se nesta dissertação um Ambiente de Realidade Aumentada Distribuída, tendo como estudo de caso a fisiologia vegetal. O **R.A. Distribuída** possui fortes características educacionais, e é implementado pelo ARToolKit e, ainda utilizando-se da plataforma CORBA 2.0 para realizar a distribuição dos objetos virtuais. Entende-se que a partir deste estudo pode-se criar interfaces de Realidade Aumentada mais interativas e distribuídas, implementando inclusive a distribuição das ações dos objetos virtuais.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALBURQUERQUE, F. **TCP/IP Internet: Programação de sistemas distribuídos – HTML, Javascript e Java.** Rio de Janeiro: Axcel Books do Brasil Editora, 2001.

AWAD, Marcel et al. **Introdução à Fisiologia Vegetal.** São Paulo: Nobel, 1993.

AZUMA, R, et al. **Recent Advances in Augmented Reality.** IEEE Computer Graphics and Applications, November/December 2001, vol. 21, p. 34-37.

AZUMA, R. T.. **A survey of augmented reality, Presence: Teleoperators and virtual environments**, vol. 6, p. 355- 385. 1997

BARANAUSKAS, M. C. C.; ROCHA, H. V. (2003) **Design e Avaliação de Interfaces Humano–Computador.** Campinas – SP: NIED/UNICAMP, 2003.

BILLINGHURST, M. **Augmented Reality in Education.** New Horizons for Learning. fevereiro de 2005. Disponível em: <<http://www.newhorizons.org>> .Acesso em: 10 fevereiro de 2008.

BILLINGHURST, Mark; KATO, Hirokazu & POUPYREV, Ivan. **The MagicBook: A Transitional AR Interface.** IEEE Computer Graphics and Applications. Projects in VR, 2001.

BOTELHO, Fábio Pereira, **Realidade Virtual Aplicada a Aquisição de Informações do Subsolo.** Grupo de Realidade Virtual e Multimídia do Centro de Informática - UFPE, Junho de 2002. Disponível em: <<http://www.cin.ufpe.br/~grvm>>. Acesso em 10 Junho 2007.

CAMPOS, Fernanda. **Dez etapas para o desenvolvimento de software educacional do tipo hipermídia.** Núcleo de Informática na Educação Especial, UFRGS, 1998. Disponível em: <<http://www.niee.ufrgs.br/ribie98>>. Acesso em 16 jan. 2008.

CARDOSO, Alexandre, et al. **Conceitos de Realidade Virtual e Aumentada.** In: Tecnologias para o desenvolvimento de sistemas de Realidade Virtual e Aumentada. Recife: Ed. Universitária da UFPE, 2007.

COSTA, Samira Rachid da. **Objetos distribuídos:** Conceitos e padrões. Instituto Nacional de Pesquisas Espaciais. Ministério da Ciência e Tecnologia, 2000.

DAINESSE, C.A., GARBIN, T.R. e KIRNER, C. **Sistema de Realidade aumentada para o Desenvolvimento da Criança Surda,** In:VI Symposium on Virtual Reality:, 2003. Ribeirão Preto - SP. SBC, 2003. p.273-281.

FREITAS, Marlene Roque de; SILVA, Luciano F; LAMOUNIER, Edgard; CARDOSO, Alexandre; GOULART, Reane F. **Proposta de Integração de Jogos e Realidade Virtual no Ensino Aprendizagem de Matemática - Projeto Ludostop.** In: CONFERÊNCIA

## Referências Bibliográficas

---

INTERNACIONAL EM EDUDAÇÃO EM ENGENHARIA E COMPUTAÇÃO, 2007, Mongagua. 2007.

GARBIN, T.R., DAINESE, C.A., KIRNER, C., SANTOS, A.M., JESUS, M.A. **Avaliação de Interface de um Sistema de Realidade Aumentada para a Criança Surda com base no Construcionismo** Anais do 1º Workshop de Realidade Aumentada, Piracicaba, SP, maio de 2004, p. 33-36.

GLUT. **GLUT - The OpenGL Utility Toolkit**. Eye-Sys, Visualizationm, evolved. OpenGL. Org, Janeiro de 2007. Disponível em: <<http://www.opengl.org/resources/libraries/glut/>> . Acesso em: maio de 2008.

HESINA, GERD; SCHMALSTIEG, Dieter. **Distributed Applications for Collaborative Aumented Reality**. IEEE Computer Society. Virtual Reality, 2005.

JING, Zhang; et al. **Researches on Model of Navigation in CVE**. GIS Development - Technology Laboratory, May 2005. Disponível em: <<http://www.gisdevelopment.net>> . Acesso em: 13 janeiro de 2008.

KALAWSKY, R.S. **Exploring Virtual Reality Techniques in Education and Training: Technological Issues**. Advanced VR Research Centre, Loughborough, University of Tecnology, 1993.

KATO, H.; BILLINGHURST, M. et. al. **ARToolKit version 2.52**: A software Library for Augmented Reality Applications. HitLab – Human Interface Technology Laboratory. Junho de 2005. Disponível em: <<http://www.hitl.washington.edu/>>. Acesso em: 10 janeiro de 2007.

KIRNER, C. ; TORI, R. **Introdução à Realidade Virtual, Realidade Misturada e Hiperealidade**. In: Cláudio Kirner; Romero Tori. (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. 1ed. São Paulo, 2004, v. 1, p. 3-20.

KIRNER, Cláudio. **ARToolKit versão 2.52**. Realidade Virtual e Aumentada, Abril de 2008. Disponível em: <<http://www.realidadevirtual.com.br>>. Acesso:maio 2008.

KIRNER, Cláudio; GERALDI, Thiago Humberto. **Ambientes Colaborativos com Realidade Aumentada**. In: II Workshop de Realidade Aumentada – WRA'2005, 2, 2005, Piracicaba-SP. *Proceedings*. Piracicaba-SP, 2005.p.13-16.

KIRNER, Cláudio; SISCOUTTO, Robson. **Realidade Virtual e Aumentada: Conceitos, Projetos e Aplicações**. Porto Alegre: SBC, 2007.

LÉVY, Pierre. **O que é virtual?**, Tradução de Paulo Neves, São Paulo: Ed. 34, 1999.

LIMA, Rayfran Rocha. **Tutorial CORBA**. Programa de Pós-Graduação da UFRJ. Janeiro de 2002. Disponível em <[http://www.gta.ufrj.br/grad/00\\_2/corba](http://www.gta.ufrj.br/grad/00_2/corba)>. Acesso em: 10 de abril de 2008.

## Referências Bibliográficas

---

MACEDONIA, M.R.; ZYDA, M.J.A **Taxonomy for Networked Virtual Environments**. IEEE Computer Society. Janeiro de 2005. Disponível em: <<http://doi.ieeecomputersociety.org/10.1109/93.580395>>. Acesso em: 20 de Novembro de 2007.

MILGRAM, P. et. al. **Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum**. Telemanipulator and Telepresence Technologies, SPIE, V.2351, 1994.

OLIVEIRA, J. C. **Collaborative Virtual Environment for Industrial Training and e-Commerce**. In: Workshop on Application of Virtual Reality Tecnologies for Future Telecommunication Systems. IEEE Globecom'2000 Conference. Proceedings... San Francisco, 2000.

OMG, **OMG and the CORBA default**. The Object Management Group (OMG). Fevereiro de 2008. Disponível em: <<http://www.omg.org/>> acesso em: 25 de fevereiro de 2008.

PINHO, Márcio Serolli. **Realidade Virtual como Ferramenta de Informática na Educação**. Grupo de Realidade Virtual da PUC-RS, Outubro de 1996. Disponível em: <<http://grv.inf.pucrs.br/pagina/educa/educa.htm>>. Acesso em: 02, jun. 2007.

PORTAL BIOLOGIA. **Introdução da Fisiologia Vegetal**. Porta Educação, Fevereiro de 2007. Disponível em:<<http://www.portalbiologia.com.br/biologia/principal>>. Acesso em: 25 de Maio de 2008.

PRUGMPER. **Test it: ISONORM 9241/10**. In: H.-J. BULLINGER & J. ZIEGLER Eds. Proceedings of HCI International, Munich, 22-27 August 1999. Mahwah, NJ: Lawrence Erlbaum. 1999.

RAJ, G.S. **A Detailed Comparison of CORBA, DCOM and Java/RMI**. Gopalan. Janeiro de 2004. Disponível em <<http://my.execpc.com/~gopalan>>. Acesso em: 02 de outubro de 2007.

RAPOSO, Alberto; et. al. **Distributed Augmented Chess System**. Rio de Janeiro, TECGRAF – Computer Graphics Technology – PUC-Rio, 2003.

RIBEIRO, Marcos Wagner de Souza. **Arquitetura para Distribuição de Ambientes Virtuais Multidisciplinares**, Uberlândia, Faculdade de Engenharia Elétrica - UFU, 2006, 176p.

RICCIONI, Paulo Roberto. **Introdução a Objetos Distribuídos com CORBA**. Florianópolis: Visual Books, 2000.

RINALDI, Cristina Alves; et al. **Ambientes Virtuais Distribuídos e Compartilhados**. In TORI, Romero; KIRNER, Cláudio; SISCOUTTO, Robson. Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Porto Alegre. Editora SBC, 2006.

SEMENTILLE, A. C. **A utilização da Arquitetura CORBA na Construção de Ambientes Virtuais Distribuídos**. 1999. 186f; Tese (Doutorado em Física Computacional) – Instituto de Física de São Carlos, USP. São Carlos, 1999.

## Referências Bibliográficas

---

SEMENTILLE, A. C., et. al. **Ambientes Virtuais Distribuídos usando CORBA:** um estudo de caso. In: III Workshop de Realidade Virtual – WRV’2000, 3, 2000, Gramado-RS. *Proceedings*. Gramado-S, 2000.p.145-156.

SINGHAL, S; ZYDA, M. **Networked Virtual Environment:** Design and Implementation, New York: ACM Press, 331 p., Siggraph Series.

SIQUEIRA, Luiz Leonardo de. **Estudo comparativo entre plataformas de suporte a Ambientes Virtuais Distribuídos**, Uberlândia, Faculdade de Engenharia Elétrica – UFU, 2005, 89p.

TAIZ, Lincoln. **Fisiologia Vegetal**. São Paulo: ARTMED, 2006.

TEIXEIRA, S. **Delphi 6, o guia do desenvolvedor**. Rio de Janeiro: Campus, 2002.

TORI, Romero; KIRNER, Cláudio; SISCOUTTO, Robson. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Porto Alegre: SBC, 2006.

VALENTE, J.A. **Aprendendo para a vida: o uso da informática na educação especial**. In VALENTE, J.A. Aprendendo para a vida: os computadores na sala de aula. São Paulo. Cortez Editora, p.29-42, 2001.

ZHOU, ZhiYing; et al. **Interactive Entertainment Systems Using Tangible Cubes**. IEEE Computer Society. Virtual Reality, 2004.

ZORZAL, Ezequiel; KIRNER, Cláudio; CARDOSO, Alexandre; LAMOUNIER JR., Edgard. **Visualização de Dados Relativos a Redes de Computadores Usando Realidade Virtual e Aumentada**. In: IX Symposium on Virtual and Augmented Reality – SVR’2007, 9, 2007, Petrópolis-RJ. *Proceedings*. Petrópolis-RJ, 2007.

**ANEXO**

**QUESTIONÁRIO BASEADO NA ISONORM**

# Avaliação do Sistema – R.A. Distribuída

Avaliador: \_\_\_\_\_

Data da Avaliação: \_\_\_\_ / \_\_\_\_ / 2008

Nível de Escolaridade: ( ) Ensino Médio ( ) Ensino Superior ( ) Pós-Graduação

Nome do Curso: \_\_\_\_\_

## I. Quanto à Finalidade (Eficiência):

Insatisfeito           Muito satisfeito

## II. Quanto à Interface (Usabilidade): entendimentos dos comandos

Insatisfeito           Muito satisfeito

## III. Quanto à facilidade de uso (Funcionalidade)

Insatisfeito           Muito satisfeito

## IV. Quanto ao nível de interação com os objetos disponíveis na cena

Insatisfeito           Muito satisfeito

## V. Quanto a manipulação dos objetos virtuais na cena (facilidade de uso)

Insatisfeito           Muito satisfeito

## VI. Quanto a manipulação dos menus interativos (facilidade de uso)

Insatisfeito           Muito satisfeito

## VII. Quanto a colaboração entre os usuários (é perceptível)

Insatisfeito           Muito satisfeito

## VIII. Quanto a aprendizagem (obteve-se aquisição de conhecimento?)

Insatisfeito           Muito satisfeito

## VIII. Quanto aos objetos virtuais na cena (permite a concepção da experiência proposta)

Insatisfeito           Muito satisfeito

## XI. Quanto ao número de funções existentes no sistema

Insatisfeito           Muito satisfeito

## XI. Quanto a confiança de que o sistema já executou o que foi pedido

Insatisfeito           Muito satisfeito

## XII. Quanto o número de passos necessário para a realização de uma atividade

Insatisfeito           Muito satisfeito

## XIII. Quanto o controle sobre a atividade (possibilidade de interromper, suspender, etc)

Insatisfeito           Muito satisfeito

## XIV. Quanto a possibilidade de visualização clara do que está sendo executado

Insatisfeito           Muito satisfeito

## Justificativas, comentários e sugestões