Claiton de Oliveira

**Associação de Redes de Petri com Objetos
Virtuais e Reais para Controle de Ambientes
Virtuais Imersivos e Telepresença.**

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Manufatura.

Orientador: Prof. Dr. Arthur José Vieira Porto

São Carlos

2008

Folha de Julgamento

Aos meus pais...

Agradecimentos

Em tudo o que realizamos, sempre existem pessoas nos “bastidores” que contribuíram de diversas formas para o nosso sucesso. Eu tenho sido abençoado por Deus, por estar sempre cercado por um grupo de pessoas incríveis que sempre me apoiaram, me guiaram, me encorajaram e me inspiraram a continuar sempre adiante.

Gostaria de agradecer aos meus pais, por serem uma fonte de inspiração em tudo que faço, que compartilham comigo meus ideais e me incentivam em todas as minhas aspirações. Juntamente com os meus irmãos, são as pessoas mais importantes da minha vida.

Ao professor Dr^o. Arthur José Vieira Porto, pela oportunidade de realização deste trabalho, pela sua confiança, apoio e a amizade formada nestes dois anos de mestrado.

Não poderia deixar de expressar a minha gratidão à professora Dr^a. Jandira Guenka Palma, pela sua amizade, confiança e generosidade investidas em mim desde os tempos da graduação. Sinto-me realmente abençoado por ter tido a oportunidade de conhecer esta pessoa notável que é digna de todo o meu reconhecimento e admiração.

À Daniele, pelo carinho, atenção, afeto, companheirismo e cumplicidade. Você me deu forças para continuar nos momentos mais difíceis.

Aos amigos Heráclito e Rafael que sempre estiveram dispostos a ajudar sem pedir nada em troca, simplesmente por serem amigos.

À Beth, do Laboratório de Simulação, por sua amizade e solicitude e à Ana Paula, da Secretaria de Pós-graduação, por sua paciência em me atender sempre que preciso.

Ao CNPq pelo apoio financeiro através da bolsa de estudos concedida.

E, claro, à todos os amigos e colegas de laboratório, pelas amizades formadas e o companheirismo existente durante esses últimos anos.

Não posso deixar de dizer, enfim, o quanto considero difícil deixar de nominar todas as pessoas a quem eu gostaria de agradecer. No entanto, quando somos abençoados em ter tantas pessoas para agradecer, a maior gratidão e reconhecimento deve ser dada à Deus.

À Ele sou grato por tudo, sempre.

"God is able to do exceedingly abundantly above all that we ask or think..."

Ephesians 3:20

Resumo

OLIVEIRA, C. **Associação de redes de petri com objetos virtuais e reais para controle de ambientes virtuais imersivos e telepresença**. 2008. 123 p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

A telepresença, como um refinamento da teleoperação, permite que o indivíduo controle não apenas a simulação mas a própria realidade. Uma vez que, esta proporciona a habilidade de manipular a realidade física remotamente em tempo real, através de sua representação virtual. Através da tecnologia de realidade virtual, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações, possibilitando simulações de situações do mundo real em um ambiente ou mundo virtual. A aplicação de ambientes virtuais no processo de manufatura está relacionada tanto com a manutenção e treinamento, como também a criação e simulação de protótipos virtuais antes de sua utilização no chão-de-fábrica. O uso de redes de Petri para modelagem de ambientes virtuais, mostra-se como uma poderosa ferramenta gráfica/matemática que pode capturar naturalmente as principais características dos sistemas de realidade virtual, além de proporcionar resultados empíricos interessantes na verificação automática de concorrência e sistemas de tempo real. Por permitir modelar a computação não determinística e paralela, a modelagem de ambientes virtuais com redes de Petri é adequada não apenas para simulação, verificação e validação desses sistemas, mas para o próprio controle dos ambientes modelados. Ao associar os elementos de uma rede de Petri com os elementos do paradigma orientado à objetos de um conjunto de classes que constituem os objetos que irão compor os ambientes virtuais e os sistemas de telepresença, obtém-se uma valiosa ferramenta para o controle dos métodos dos objetos virtuais e reais a serem invocados. O objetivo deste trabalho é o desenvolvimento do modo de associação de redes de Petri com objetos virtuais e reais para controle de ambientes virtuais imersivos e telepresença, com base na descrição de um sistema de autoria de telepresença com controle em RP. Os resultados obtidos trarão avanços quanto ao desenvolvimento e implantação de aplicações de manufatura virtual em um ambiente de visualização 3D imersivo.

Palavras-Chaves: Redes de Petri, Ambientes Virtuais, Telepresença.

Abstract

OLIVEIRA, C. **Association of petri nets with virtual and real objects for controlling immersive virtual environments and telepresence.** 2008. 123 p. Thesis (Master) - School of Engineering of São Carlos, University of São Paulo, São Carlos, 2008.

The telepresence, as a refinement of teleoperation, allows the individual not only to control the simulation but also the own reality. Since this provides the ability to manipulate the physical reality remotely in real time, through its virtual representation. Through the technology of virtual reality, one can create a wide variety of environments suitable for all kinds of applications, enabling simulations of the situations in a real world environment or virtual world. The application of virtual environments in the manufacturing process is related to both the maintenance and training, and also the creation and simulation of virtual prototypes before their use in the shop-floor. The use of Petri nets for modeling virtual environments, shows itself as a powerful graphics / mathematics tool that can capture the main features of virtual reality systems, as well as offering interesting empirical results in automatic verification of concurrence and real time systems. By allowing to model non-deterministic and parallel computation, modeling of virtual environments with Petri nets is appropriate not only for simulation, verification and validation of these systems, but also for controlling the environments modeled. By associating the elements of a Petri net with the elements of the object oriented paradigm of a set of classes that are the objects that will compose the virtual environments and telepresence systems, one gets a valuable tool for controlling of the methods of virtual and real objects to be invoked. The objective of this work is the development of the mode of association of Petri nets with virtual and real objects for controlling immersive virtual environments and telepresence, based on the description of a telepresence authoring system controlled by PN. The results will bring progress on the development and deployment of virtual manufacturing applications in an immersive 3D visualization environment.

Keywords: Petri Nets, Virtual Environments, Telepresence.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

1	Introdução	p. 25
1.1	Justificativa	p. 28
1.2	Objetivos	p. 30
1.3	Organização do Trabalho	p. 30
2	Revisão Bibliográfica	p. 33
2.1	Redes de Petri	p. 33
2.1.1	Propriedades das RP	p. 35
2.1.2	Extensões de RP	p. 37
2.1.3	Redes de Petri Coloridas	p. 38
2.2	Ambientes Virtuais e Telepresença	p. 41
2.2.1	Características dos Ambientes Virtuais	p. 43
2.2.2	Tipos de Sistemas de Realidade Virtual	p. 45
2.2.2.1	Grau de Interação Proporcionado do Usuário	p. 46
2.2.2.2	Grau de Imersão Proporcionado ao Usuário	p. 47
2.2.3	Dispositivos de Realidade Virtual	p. 49
2.2.3.1	Dispositivos de Rastreamento	p. 50

2.2.3.2	Dispositivos de Interação	p. 51
2.2.3.3	Dispositivos visuais	p. 54
2.2.3.4	Dispositivos Auditivos	p. 58
2.2.3.5	Dispositivos Hápticos	p. 60
2.2.3.6	CAVE	p. 61
2.2.4	Dispositivos de Telepresença	p. 65
2.2.5	Bibliotecas de Desenvolvimento de Ambientes Virtuais	p. 66
2.2.6	Aplicações de Ambientes Virtuais e Telepresença	p. 69
2.2.6.1	Manufatura Virtual	p. 70
2.3	Redes de Petri e Ambientes Virtuais	p. 71
3	Sistema de Autoria de Telepresença com Controle em Redes de Petri	p. 73
3.1	Estrutura do Sistema	p. 74
3.2	Componentes do Sistema	p. 76
3.2.1	Biblioteca de Classes de Objetos	p. 76
3.2.2	Configurador de Ambientes Virtuais	p. 77
3.2.3	Editor de Redes de Petri	p. 78
3.2.4	Ferramenta de Autoria	p. 79
3.2.5	Módulo Central de Controle	p. 79
3.2.6	Sistema de Ambientes Reais	p. 80
3.2.7	Sistema de Ambientes Virtuais	p. 80
3.2.8	Sistema de Comunicação	p. 81
3.2.9	Protótipos Reais e Virtuais	p. 81
3.2.10	Sistema de Telepresença	p. 82
4	Associação de Redes de Petri com Objetos Virtuais e Reais	p. 83
4.1	Relação entre os Modelos de RP e os Ambientes Virtuais	p. 84

4.1.1	Classes de Objetos Virtuais e Reais	p. 86
4.1.2	Associação dos Objetos Virtuais e Reais com uma RP Ordinária	p. 86
4.1.3	Limitações na Abordagem de Palma (2001)	p. 88
4.2	Associação dos Objetos Virtuais e Reais com uma RP de Alto Nível	p. 91
4.2.1	Redes de Petri Coloridas (RPC)	p. 92
4.2.2	Associação dos Objetos Virtuais e Reais com uma RPC	p. 93
4.3	Estudo de Caso	p. 97
4.3.1	Associação de uma RPC com os Protótipos Botão/Esteira	p. 98
4.3.2	Associação de uma RPC com o Protótipo Guindaste	p. 101
4.4	Resultados e Discussão	p. 105
5	Conclusão	p. 109
5.1	Contribuições	p. 111
5.2	Trabalhos Futuros	p. 112
	Referências	p. 115

Lista de Figuras

1	Elementos de uma Rede de Petri.	p. 34
2	(a) Notação gráfica e (b) Notação matemática de uma mesma RP.	p. 35
3	Exemplo de uma Rede de Petri Colorida.	p. 38
4	Dispositivo de interação do tipo esteira chamado <i>CirculaFloor</i> (IWATA et al., 2004).	p. 53
5	Dispositivo de interação do tipo esfera da VirtuSphere.	p. 54
6	Dispositivo binocular suspenso.	p. 56
7	Dispositivos hápticos fixos: Falcon da Novint (NOVINT, 2008) e Phantom da SensAble (SENSABLE, 2008) respectivamente.	p. 61
8	Dispositivos hápticos CyberTouch, CyberGrasp, CyberForce e Haptic Workstation da Immersion Corporation (IMMERSION, 2008).	p. 61
9	Projetistas experimentando e avaliando o interior de um carro dentro de um CAVE.	p. 63
10	Esquema que descreve a posição dos projetores em um sistema do tipo CAVE.	p. 64
11	Estrutura do Modelo de Sistema de Autoria de Telepresença.	p. 75
12	Diagrama da estrutura das Classes de Objetos Virtuais e Reais.	p. 76
13	Configurador do Ambiente Virtual.	p. 78
14	Edição do elemento lugar (<i>place</i>) no Editor de Redes de Petri.	p. 79
15	Protótipos reais: guindaste, botão e esteira respectivamente.	p. 81
16	Objeto Virtual (Guindaste) que compõe o Ambiente Virtual 3D.	p. 82
17	Associação do Modelo de RP com o Ambiente Virtual.	p. 87
18	A tabela de conexão especifica a ligação dos elementos da RP com os elementos do objeto botão.	p. 88

19	Modelagem individual de 1 e 2 AGV's servindo uma máquina, respectivamente.	p. 90
20	Modelagem de 2 AGV's servindo uma máquina, sem identificação individual.	p. 90
21	(a) 2 AGV's servindo uma máquina, sem identificação individual. (b) 3 AGV's servindo uma máquina, com identificação individual.	p. 93
22	Composição de uma cena através do CAV.	p. 93
23	Arquivo XML que descreve o objeto botão pertencente a cena configurada no CAV.	p. 94
24	Associação do Objeto Virtual Botão com a Rede de Petri Colorida.	p. 95
25	Associação da RPC com o objeto virtual botão, representada em XML. . . .	p. 96
26	Arquivo XML que descreve os objetos (botão e esteira) que compõem a cena do ambiente virtual.	p. 99
27	Modelo em RPC dos objetos botão e esteira.	p. 100
28	Associação dos objetos (botão e esteira) com a RPC.	p. 101
29	Arquivo XML que descreve o objeto guindaste que compõe a cena do ambiente virtual.	p. 102
30	Modelo em RPC do objeto guindaste.	p. 103
31	Associação do objeto guindaste com a Rede de Petri Colorida.	p. 105

Lista de Tabelas

1	Crescimento do número de elementos da RP.	p. 90
2	Descrição dos lugares apresentados na RPC da Figura 27.	p. 100
3	Descrição das transições apresentados na RPC da Figura 27.	p. 100
4	Descrição das variáveis apresentada na RPC da Figura 27.	p. 101
5	Descrição dos lugares apresentados na RPC da Figura 30.	p. 103
6	Descrição das transições apresentados na RPC da Figura 30.	p. 104
7	Descrição das variáveis apresentada na RPC da Figura 30.	p. 104

Lista de Abreviaturas e Siglas

AGV	Automatic Guided Vehicle
AMD	Arm-Mounted Display
API	Application Programmer's Interface
AR	Ambiente Real
AV	Ambiente Virtual
BCOVR	Biblioteca de Classes de Objetos Virtuais e Reais
CAV	Configurador de Ambientes Virtuais
CAVE	CAVE Automatic Virtual Environment
CNC	Computer Numerical Control
DHM	Dexterous Hand Master
HMD	Head-Mounted Display
HRTF	Head Related Transfer Function
HUD	Head-Up Display
LED	Light Emitting Diode
MCC	Módulo Central de Controle
RP	Rede de Petri
RPAN	Rede de Petri de Alto Nível
RPC	Rede de Petri Colorida
RPCH	Rede de Petri Colorida Hierárquica
RPH	Rede de Petri Hierárquica
RV	Realidade Virtual
SAR	Sistema de Ambientes Reais
SAV	Simulador de Ambientes Virtuais
SID	Spatially Immersive Display
UDP	User Datagram Protocol
VRML	Virtual Reality Modeling Language
WIMP	Windows, Icons, Menus and Pointing Device
X3D	Extensible 3D
XML	eXtensible Markup Language

1 Introdução

“Realidade Virtual”, “Mundos Virtuais” ou “Ambientes Virtuais” são termos utilizados para definir ambientes 3D gerados por computador, que possibilitam ao usuário imergir e interagir em um ambiente sintético. A idéia de imersão está ligada ao sentimento de fazer parte do ambiente - o usuário vê-se envolvido no domínio da aplicação (realidade), tendo uma sensação de presença dentro do mundo virtual. Enquanto a interação está ligada à capacidade do sistema detectar as entradas do usuário e modificar instantaneamente o mundo virtual em função das ações efetuadas sobre ele.

Para Draper, Kaber e Usher (1999), um ambiente sintético permite que as capacidades perceptivas, cognitivas e psicomotoras humanas sejam projetadas em um ambiente distante, hostil ou simulado. Assim, Capin et al. (1999) definem a Realidade Virtual (RV) como uma tecnologia capaz de transportar um indivíduo para um ambiente diferente do real, sem movê-lo fisicamente, manipulando as informações destinadas aos órgãos sensoriais humanos de tal maneira que o ambiente percebido seja associado ao Ambiente Virtual (AV) desejado e não ao Ambiente Real (AR).

Steuer (1992), acrescenta que a definição de realidade virtual é baseada em conceitos de “presença” e “Telepresença”, ou seja, a chave para a definição da realidade virtual, em termos de experiência humana ao invés de tecnologias de *hardware* e *software*, é o conceito de presença. Onde a presença, pode ser entendida como a percepção de um ambiente físico, mediada apenas pela mente humana. No entanto, quando a percepção é mediada por alguma tecnologia de comunicação, a pessoa é obrigada a perceber simultaneamente dois ambientes distintos: o ambiente físico em que o mesmo se encontra presente, e o ambiente apresentado através do meio tecnológico. Logo, o termo “Telepresença” pode ser usado para descrever a última experiência como a sensação de estar em um ambiente mediado por algum meio de comunicação, ao invés do ambiente físico imediato.

Sendo a presença uma percepção natural de um ambiente e a telepresença a percepção de um ambiente mediado, este último pode ser tanto um ambiente real

temporalmente ou geograficamente distante como um ambiente virtual sintetizado por computador. Dessa forma, estudos sobre a telepresença têm explorado tecnologias de realidade virtual a fim de orquestrar a percepção humana a partir de um ambiente distante, uma vez que esta proporciona a habilidade de manipular a realidade física remotamente em tempo real, através de sua representação virtual (WILSON, 2003).

A telepresença, como um refinamento da teleoperação, permite que o indivíduo controle não apenas a simulação mas a própria realidade. Neste contexto, Goldberg (2000) afirma que a essência da telepresença consiste na antipresença, ou seja, o indivíduo não tem que estar fisicamente presente em um local para afetar a realidade do mesmo. Logo, Mair (1997) afirma que isto é muito mais do que uma simples teleconferência, uma vez que permite ao usuário sentir-se imerso em um ambiente remoto e ainda ser capaz, através da teleoperação, de manipular ou controlar eventos remotos.

Dessa forma, os sistemas de telepresença referem-se a um conjunto de tecnologias que permitem a sensação de presença em um local remoto, a partir de uma localização diferente da que pretende-se interagir (STEUER, 1992; FISHER, 1991). Muito similar à realidade virtual, onde luta-se para alcançar a ilusão de presença dentro de uma simulação gerada por computador, a telepresença luta para alcançar a ilusão de presença em um local remoto. Logo a idéia da telepresença, lembra a RV, exceto que na RV o usuário está imerso em um mundo virtual, ao passo que na telepresença o usuário está imerso em um mundo real remoto. Assim, Rosenberg (1995) afirma que no fim, tanto a telepresença como a realidade virtual são essencialmente o mesmo, uma interface humano-computador que permite ao usuário tirar vantagem de suas habilidades naturais quando interage com um outro ambiente que não aquele que o rodeia.

As fronteiras entre os sistemas de RV e sistemas de telepresença estão cada vez mais tênues e lentamente se fundem em um espectro muito amplo de tecnologias utilizadas para visualizar e interagir com mundos tridimensionais em tempo real. Através das tecnologias de RV e telepresença, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações, possibilitando simulações de situações do mundo real em um ambiente ou mundo virtual. A telepresença e operadores remotos que empregam a RV tanto para educação, treinamento, simulação como entretenimento, têm sido explorada por cientistas e desenvolvedores de sistemas complexos como apresentados em (HANÁK et al., 2007; VALLIUS; RÖNING, 2006; YANG et al., 2005).

A aplicação de ambientes virtuais (AV) no processo de manufatura está relacionada tanto com a manutenção e treinamento, como também a criação e simulação de protótipos

virtuais antes de sua utilização no chão-de-fábrica. A criação de protótipos virtuais pode proporcionar diversas vantagens aos processos industriais, como a redução do *time-to-market*, ou seja, do tempo decorrente entre o início do processo de desenvolvimento de um produto à sua introdução no mercado, e a redução dos custos de desenvolvimento, pois protótipos virtuais permitem reduzir o número de protótipos físicos necessários, conseqüentemente reduzindo a quantidade de ferramentas e materiais utilizados para a sua construção.

Wang, Shen e Lang (2004) listam trabalhos relacionados com manufatura virtual envolvendo a área automotiva, aeroespacial, indústria pesada e a indústria de ferramentas e máquinas, onde sistemas são projetados entre outras funcionalidades, para monitoramento e controle de chão-de-fábrica, operação de máquinas CNC (*Computer Numerical Control*) remotas ou automação de fábrica avançada.

Dentre o benefícios oferecidos pela virtualização de equipamentos de manufatura, Shridhar e Ravi (2002) citam: a verificação e modificação rápida dos projetos, maior confiabilidade às decisões, avaliação e testes de propriedades estruturais e funcionais de máquinas e produtos projetados, desenvolvimento de um produto ergonomicamente funcional e confiável sem a necessidade de construir um modelo em escala real, projeção da estética do produto de acordo com as preferências do consumidor, além do desenvolvimento e avaliação de processos assegurando a manufaturabilidade e a montabilidade, sem produzir de fato o produto em escala comercial.

A necessidade crescente da utilização de recursos de modelagem para projetar ambientes virtuais gerou alguns ambientes baseados em Redes de Petri (RP). A representação gráfica das Redes de Petri, além de ser de fácil compreensão, permite o encapsulamento de detalhes, oferecendo um método adequado para a modelagem dos diferentes níveis de coordenação de ambientes virtuais. Além disso, as RPs oferecem um forte suporte matemático para a análise do comportamento dos ambientes virtuais e técnicas de simulação complementares. Com o modelo baseado em RPs, é possível prever e testar o comportamento de um ambiente antes mesmo de sua implementação.

Em meio à ampla variedade de sistemas de RV disponíveis, Sherman e Craig (2002) consideram a tecnologia CAVE (*Computer Automatic Virtual Environment*) como a mais completa para visualização de dados. A caverna digital ou CAVE - um título inspirado pela "Caverna de Platão" que retrata a exploração da percepção, ilusão e realidade através do uso de metáforas, contada em sua obra "A República" - consiste em um ambiente de multiprojeção estereoscópico voltado para aplicações imersivas de realidade virtual (CRUZ-NEIRA, 1995). Ou seja, um sistema de multiprojeção estereoscópico montado na forma de cubo, onde imagens

de alta resolução são projetadas em cada uma das faces do mesmo e visualizadas com óculos estéreos, permitindo que usuários sejam totalmente imersos em uma simulação gerada por computador.

Através de uma poderosa ferramenta de visualização como a CAVE aliada a um sistema de autoria de telepresença com controle baseado em Redes de Petri, torna-se possível o desenvolvimento de um vasto número de aplicações em diversos campos do conhecimento. Uma vez que o controle seja definido em relação a um nível mais abstrato dos objetos que compõem o ambiente virtual, pode-se associar os elementos de uma RP que descrevem o intertravamento de uma máquina, por exemplo, com o objeto virtual que representa esta máquina a ser projetada na CAVE, e assim, poder controlá-la tanto no mundo virtual como no mundo real.

Logo, a associação das RPs com os objetos que compõem um ambiente virtual ou real, permite que o modelo que descreve os ambientes virtuais e reais não seja utilizado apenas para simulação e análise do sistema, mas para o próprio controle do mesmo através da definição de prioridades de invocação de métodos de objetos, durante a execução das aplicações de RV imersivas e de telepresença.

1.1 Justificativa

O *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos* - um projeto financiado pela FAPESP, sob coordenação do Professor Arthur José Vieira Porto - disponibiliza uma CAVE recém instalada, com projeção em três faces: duas paredes laterais e no chão. Através deste projeto, os pesquisadores da região de São Carlos têm uma infraestrutura de visualização 3D imersiva disponível para o desenvolvimento de pesquisas, principalmente no contexto de desenvolvimento da Manufatura Virtual. Logo, uma das justificativas deste trabalho baseia-se no estudo, desenvolvimento e controle de ambientes de realidade virtual e de telepresença para o sistema CAVE.

A complexidade na construção e controle de ambiente virtuais inibe a aplicação da tecnologia de RV e da telepresença. No entanto, Palma (2001) propôs um metamodelo que facilita o emprego de protótipos reais e virtuais em aplicações envolvendo ambientes virtuais imersivos e telepresença. A idéia central deste metamodelo é disponibilizar ao projetista de simulação um ambiente de desenvolvimento que encapsule a complexidade de programação de sistemas de realidade virtual e telepresença, fornecendo elementos que permitam a construção de Ambientes Virtuais de “forma intuitiva”.

O metamodelo de Palma (2001) envolve a modelagem do controle de ambientes virtuais e reais, através de um modelo gráfico/matemático (Redes de Petri) com execução do controle centralizado. Com o desenvolvimento de diversos trabalhos relacionados ao metamodelo, foi possível obter um sistema de autoria que contém uma coleção de aplicativos a fim de editar o modelo de controle, configurar o ambiente de telepresença (virtual e real), executar o controle do ambiente (servidor) e gerar os ambientes (abstração do mundo em 3D ou o mundo real) através de instâncias em vários clientes com objetos pré-definidos (OLIVEIRA et al., 2007a, 2007b).

A ferramenta resultante do metamodelo oferece um sistema virtual que tem associado ou não o sistema real. Assim, ela pode ser utilizada em várias aplicações de realidade virtual como a telemedicina e teleoperação de equipamentos presentes em ambientes hostis ao ser humano; para treinamento dos operadores em equipamentos de alto custo ou com alta periculosidade ao operador; para demonstração de produtos (*marketing*); além de possibilitar a geração de ambientes com uma coleção de máquinas que atuam de forma integrada, como por exemplo um sistema de manufatura.

O trabalho utiliza os conceitos de telepresença como o controle de sistemas em tempo real, ambientes virtuais, ambientes reais e teleoperação, a fim de prover uma ferramenta para o desenvolvimento de sistemas de telepresença. Além disso, é baseada em linguagem de especificação formal, Redes de Petri (RP), visto que, a implementação e o teste de ambientes virtuais são complicados devido às concorrências e as características de tempo real.

A RP é uma ferramenta bem apropriada para a modelagem de ambientes virtuais, pois pode capturar naturalmente as principais características de sistemas de RV, e proporcionar resultados empíricos muito bons na verificação automática de concorrência e sistemas de tempo real, por permitir modelar facilmente a computação não determinística e paralela (MASCARENHAS et al., 1998).

Como o controle dos métodos dos objetos virtuais e reais a serem invocados é centralizado em um Módulo Central de Controle (MCC) a partir de uma RP, justifica-se estabelecer o modo de associação da RP com os objetos virtuais e reais para que o gerenciamento dos ambientes que compõem o sistema de telepresença seja o mais correto possível.

Através da especificação do modo de associação da RP com o ambiente virtual, pode-se complementar e enriquecer o trabalho iniciado por Palma (2001), consolidando as funcionalidades da ferramenta resultante do metamodelo proposto, com a inserção de melhorias ao mesmo.

1.2 Objetivos

Dentro do objetivo geral deste trabalho, encontra-se o desenvolvimento do modo de associação de Redes de Petri (RP) com objetos virtuais e reais para controle de ambientes virtuais imersivos e telepresença, com base na descrição de um sistema de autoria de telepresença com controle em RP.

Dentro dos objetivos específicos encontram-se :

- a melhoria no processo de associação da RP com o ambiente virtual do sistema de autoria de telepresença com controle em RP, precursor deste trabalho;
- a definição de uma extensão de RP que possibilite não apenas a associação com objetos virtuais mas também a redução da complexidade (extensão da rede) enfrentada na abordagem com RP Ordinárias;
- a especificação do processo de associação dos elementos de um modelo de RP com os elementos dos objetos virtuais e reais que compõem um ambiente virtual imersivo e um sistema de telepresença controlado por esta rede.
- a contribuição com o crescimento do projeto *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos*, possibilitando o desenvolvimento de aplicações para ambientes virtuais imersivos com controle em RP;

1.3 Organização do Trabalho

O texto desta dissertação está organizado em uma estrutura de 5 capítulos. Onde, o capítulo 1 refere-se à introdução, o capítulo 2 à revisão bibliográfica, o capítulo 3 ao sistema de autoria de telepresença com controle em Redes de Petri, enquanto o capítulo 4 refere-se à associação de redes de Petri com objetos virtuais e reais e, por fim, o capítulo 5 à conclusão.

Após a introdução que contextualiza, justifica e apresenta os objetivos do trabalho, o capítulo 2 apresenta os principais conceitos relacionados às Redes de Petri, suas propriedades e extensões. Além disso, o capítulo 2 ainda introduz os conceitos envolvidos em Ambientes Virtuais e Telepresença, suas características, tecnologias empregadas e variações.

No capítulo 3 é descrito um sistema de autoria de telepresença com controle em Redes de Petri, que refere-se à um trabalho precursor deste, cujo tema é uma proposta de um

metamodelo para modelagem e simulação de sistema a eventos discretos, baseado em redes de petri e realidade virtual.

A questão da associação dos elementos da Rede de Petri com os elementos dos objetos virtuais e reais é apresentada no capítulo 4, onde discorre-se sobre às limitações do método proposto ao sistema já existente, e propõe-se uma nova abordagem no controle de ambientes virtuais e reais que utilizam o paradigma Orientado à Objetos, através da associação de uma RP de Alto Nível com os objetos que compõem estes ambientes.

Por fim, no capítulo 5 é apresentado a conclusão do trabalho, onde são feitas as considerações finais, analisadas as contribuições do trabalho, assim como as perspectivas para sua continuidade, através da sugestão de trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo são apresentados os principais aspectos referentes aos métodos e tecnologias envolvidas direta ou indiretamente neste trabalho. Inicialmente, é introduzida uma breve apresentação do formalismo de Redes de Petri (RP), onde aborda-se as principais definições, características, propriedades e extensões de modelagem presentes na literatura. Em seguida, são descritos os conceitos de Ambientes Virtuais e Telepresença, introduzindo definições e apresentando suas características e classificações, além dos dispositivos envolvidos neste tipo de tecnologia. Os conceitos de Orientação a Objetos, presentes nesta proposta de especificação da associação de RP com objetos virtuais e reais, não serão aqui abordados. Este paradigma de análise, projeto e programação de sistemas é amplamente e corriqueiramente utilizado por projetistas e desenvolvedores, além de estar bem fundamentado pela literatura.

2.1 Redes de Petri

De acordo com Murata (1989), uma Rede de Petri (RP) é uma ferramenta de modelagem gráfica e matemática aplicável a muitos sistemas, principalmente na descrição e estudo de sistemas concorrentes, assíncronos, distribuídos, paralelos, não-determinísticos ou estocásticos. Protocolos de rede, sistemas *fuzzy* e aplicações de RV são exemplos de sistemas que podem ser modelados e simulados utilizando esta ferramenta.

O nome “Redes de Petri” é uma homenagem a Carl Adam Petri, criador deste modelo em 1962. Em sua tese, em alemão, intitulada *Kommunikation mit Automaten* (em inglês *Communication with Automata*) (PETRI, 1962), seu objetivo era desenvolver um modelo em que máquinas de estado fossem capazes de se comunicar.

Graficamente, uma RP é um tipo particular de grafo direcionado, como pode ser visto na Figura 1, composta por quatro tipos de elementos:

- **Lugares ou Places:** representados por elipses, são vértices que representam os estados do sistema. São componentes passivos da RP;

- **Fichas, Marcas ou Tokens:** simbolizados por pontos, representam a situação atual, ou seja, em que estado encontra-se o sistema modelado. As marcas iniciais informam o estado primitivo do sistema;
- **Transições:** vértices que representam às ações (eventos) do sistema, simbolizados por traços ou barras (horizontais ou verticais). Modelam o comportamento dinâmico do sistema;
- **Arcos:** interligam lugares a transições, e vice-versa, por meio de setas direcionais, indicando a seqüência de execução da rede. Transportam marcas entre os vértices do grafo, indicando as pré e pós-condições de uma transição. Também podem ter rótulos que definem o número de marcas a transportar de um vértice a outro.

Figura 1: Elementos de uma Rede de Petri.

Formalmente, uma RP pode ser definida como uma quintupla (P, T, A, w, M_0) , onde:

- $P = \{p_1, p_2, \dots, p_m\}$ é um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$ é um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arcos;
- $w : F \rightarrow \{1, 2, \dots\}$ é uma função que dá valor aos arcos;
- $M_0 : P \rightarrow \{0, 1, 2, \dots\}$ é a marcação inicial¹ da rede, com $(P \cap T) = \emptyset$ e $(P \cup T) \neq \emptyset$.

O comportamento de um sistema modelado por RP é descrito em termos de seus estados e suas mudanças (MURATA, 1989). Raposo (2000) destaca que, em um modelo de RP, os estados estão associados aos lugares e suas marcações, e os eventos às transições.

¹A marcação inicial da rede corresponde ao número inicial de marcas (*tokens*) em cada lugar.

Uma transição t está habilitada se cada um de seus lugares de entrada P_i possuir pelo menos $w(P_i, t)$ marcas, onde $w(P_i, t)$ é o peso do arco ligando P_i a t . Estando habilitada, uma transição pode ser disparada quando o evento associado a ela ocorrer. O disparo de t remove $w(P_i, t)$ marcas de cada um de seus lugares de entrada P_i e adiciona $w(t, P_o)$ tokens a cada lugar de saída P_o .

Uma comparação da notação gráfica com a notação matemática das RPs, pode ser vista na Figura 2. Por definição, um arco não marcado tem peso 1.

Figura 2: (a) Notação gráfica e (b) Notação matemática de uma mesma RP.

2.1.1 Propriedades das RP

Conforme Murata (1989), as propriedades das RP podem ser divididas em dois grupos: propriedades estáticas (ou estruturais) e propriedades dinâmicas (ou comportamentais). O primeiro não depende das marcas e não se modifica durante o estágio de execução da rede. Já o segundo grupo depende das marcas, e tende a estar em constante atualização. Na sequência, são apresentados os tipos de propriedades que compõem cada um desses grupos, e o que cada um deles procura evidenciar na rede.

As propriedades estruturais referem-se à:

- **Limitação** (*structural boundedness*): uma rede é estruturalmente limitada se o número de marcas em cada lugar sempre permanece o mesmo;
- **Conservação** (*conservation*): ocorre quando a soma dos valores dos arcos que ligam as pré-condições de uma transição for igual à soma das pós-condições;
- **Repetitividade** (*repetitiveness*): essa propriedade se evidencia quando uma marcação habilita ilimitadamente uma sequência de transições;

- **Consistência** (*consistency*): uma RP apresenta essa propriedade quando uma seqüência de transições é disparada, a partir da marcação inicial, retornando-a a esta configuração quando todas as transições da rede foram disparadas ao menos uma vez.

Já as propriedades comportamentais referem-se à:

- **Alcançabilidade** (*reachability*): indica a possibilidade de atingir-se um determinado lugar pelo disparo de um número finito de transições, a partir de uma marcação inicial. Satisfazendo esta propriedade, diz-se que a rede é alcançável, e todos os estados, em algum momento da execução do sistema, serão satisfeitos;
- **Limitação** (*boundedness*): define o número de marcas que cada lugar pode acumular. A passagem de um estado para outro depende do peso dos arcos associados às transições;
- **Segurança** (*safeness*): uma RP é definida como segura se todos os lugares desta rede podem contar uma ou nenhuma marca. Esta propriedade está ligada à limitação da rede;
- **Vivacidade** (*liveness*): usada para constatar a ausência de deadlocks. Assim, esta propriedade é evidenciada quanto for possível executar todas as suas transições a partir de qualquer um dos estados alcançáveis da rede;
- **Cobertura** (*coverability*): identifica se uma transição é potencialmente disparável, isto é, se uma marca pode ser obtida a partir de uma outra, anterior a ela;
- **Persistência** (*persistence*): ocorre quando o disparo de uma transição não desabilita o disparo de outra. Este tipo de comportamento está presente em sistemas paralelos e circuitos digitais com atividades assíncronas;
- **Reversibilidade** (*reversibility*): ocorre quando o lugar inicial ou um grupo específico destes pode ser novamente alcançado;
- **Justiça** (*fairness*): uma RP é dita justa se, para quaisquer duas transições, o número de vezes que uma é executada enquanto a outra não é executada é finito.

2.1.2 Extensões de RP

Nas redes de Petri apresentadas nas Figuras 1 e 2, as fichas associadas aos lugares representam dados binários. Entretanto, há diversas limitações ao utilizá-las para modelar sistemas complexos. Uma delas é a necessidade de replicação da estrutura da rede para modelar processos idênticos, o que tende a aumentar o tamanho dos modelos. Essa replicação ocorre porque suas fichas não carregam tipos de dados, assim não podemos diferenciá-las. Para suprir tais limitações e facilitar a modelagem de sistemas maiores, extensões foram propostas às redes de Petri. Surgindo assim, a classe das redes de Petri de alto nível (RPAN).

Dessa forma, Jensen (1997) afirma que pode-se agrupar as redes de Petri em duas grandes classes: as Ordinárias e de Alto Nível. As redes ordinárias se caracterizam pelo tipo de suas marcas, ou seja, suas marcas são do tipo inteiro e não negativo, enquanto que as de alto nível possuem marcas de tipos particulares.

As redes ordinárias, podem ser subdivididas em:

- **Rede Petri Binária:** é a rede mais elementar dentre todas. Essa rede só permite no máximo um *token* em cada lugar, e todos os arcos possuem valor unitário.
- **Rede Petri Lugar-Transição:** é o tipo de rede que permite o acúmulo de marcas no mesmo lugar, assim como valores não unitários para os arcos.

As RPAN, por sua vez, por permitirem associar tipos de dados às fichas. As fichas podem carregar informações complexas, que são manipuladas conforme as inscrições dos arcos. O fato das fichas expressarem informações complexas aumenta o poder de descrição dessas redes e, conseqüentemente, podemos construir modelos mais compactos. As RPAN não aumentam o poder de representação de um modelo. Entretanto, elas permitem uma maior clareza e um maior (ou menor) nível de abstração ao modelo.

Como exemplos bem difundidos das RPAN, temos as Redes de Petri Coloridas (RPC) e as Redes de Petri Hierárquicas (RPH). As RPC associam cores às marcas nos lugares, para carregar informações sobre as mesmas, possibilitando a diferenciação dos recursos aos quais as marcas representam. Já as RPH proporcionam uma descrição estruturada dos sistemas, possibilitando diferentes níveis de abstração que auxiliam na interpretação dos modelos.

Do ponto de vista teórico, a hierarquia é uma conveniência gráfica que não adiciona poder computacional, contudo permite uma modelagem compacta em níveis de abstração variados de sistemas de grande porte. Nessa extensão, um conjunto de lugares e transições

pode ser uma subrede da rede global. Para tanto, lugares e transições são utilizados como interface entre uma rede de mais alto nível e suas respectivas subredes.

Nesta seção será abordada apenas a extensão de RP do tipo Rede de Petri Colorida (RPC), por ter sido a extensão utilizada na realização do trabalho.

2.1.3 Redes de Petri Coloridas

As redes de Petri coloridas têm por objetivo reduzir o tamanho do modelo, permitindo que as marcas sejam individualizadas, através de cores atribuídas à elas; assim, diferentes processos ou recursos podem ser representados em uma mesma rede. As cores não significam apenas cores ou padrões. Elas podem representar tipos de dados complexos, usando a nomenclatura de colorida apenas para referenciar a possibilidade de distinção entre os *tokens* (JENSEN, 1997).

Ainda segundo Jensen (1997), uma RPC é uma representação gráfica e intuitiva que facilita a visualização da estrutura básica de um modelo complexo, além de permitir a compreensão individual do comportamento dos processos do sistema. A Figura 3 apresenta uma rede colorida, onde as cores utilizadas para as marcas são representadas por letra. Da mesma forma, os arcos são rotulados com as cores (a, b, c).

Figura 3: Exemplo de uma Rede de Petri Colorida.

No exemplo da Figura 3, utiliza-se o modo mais elementar de redes coloridas, no qual se associa ao arco uma determinada cor, assim, a marca se destinará ao arco cuja cor for idêntica a própria marca. Observando-se essa figura, pode-se perceber que as marcas de $p0$ não habilitarão a transição $t0$, pois o arco que liga $P0$ a $t0$ só aceita cores do tipo “a”, e o lugar $p0$ só possui marcas do tipo “d”. Em contrapartida, $p1$ possui marcas do tipo “a”, podendo habilitar a transição $t1$.

Conforme o exemplo apresentado, percebe-se que RPC provê mecanismos que possibilitam efetuar uma escolha determinística. Esse poder de escolha já significa um grande

avanço em direção a uma representação mais clara de um modelo. No entanto, o exemplo dado ainda é requer algumas considerações que vieram dar maior adequação às redes coloridas, com relação à representação das escolhas não-determinísticas. A seguir, são apresentadas algumas características pertencentes às redes coloridas, que as tornam mais poderosas do que a simples associação de cores das marcas com as cores dos arcos.

As redes de Petri coloridas são compostas por três diferentes partes :

- **estrutura:** é um grafo dirigido com dois tipos de vértices (lugares e transições). Os lugares são representados graficamente por círculos (ou por elipses) e as transições por retângulos. Essa representação herda a propriedade das redes coloridas originais de poder armazenar em cada lugar marcas de tipos diferentes, além de poder representar valores associados a tipos de dados mais complexos.
- **declarações:** compreendem a especificação dos conjuntos de cores e declarações de variáveis.
- **inscrições:** estas variam de acordo com o componente da rede. Os lugares possuem três tipos de inscrições: nomes, conjunto de cores e expressão de inicialização (marcação inicial). As transições têm dois tipos de inscrições: nomes e expressões de guarda, e os arcos apenas um tipo de inscrição dado pela expressão.

De acordo com Jensen (1997), a definição formal de uma RPC é uma nômupla $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, onde:

- Σ é um conjunto finito, não vazio, de tipos, chamado **conjunto de cores**;
- P é um conjunto finito de **lugares**;
- T é um conjunto finito de **transições**;
- A é um conjunto finito de **arcos** onde: $P \cap T = P \cap A = T \cap A = \emptyset$;
- $N : A \rightarrow P \times T \cup T \times P$ é uma função **nó**;
- $C : P \rightarrow \Sigma$ é uma função **cor**;
- $G : T \rightarrow Exp$ é uma função **guarda**, onde Exp é da forma: $\forall t \in T : [Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma]$;

- $E : A \rightarrow Exp$ é uma função de **expressão de arco**, onde Exp é da forma:
 $\forall a \in A : [Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$, onde $p(a)$ é um lugar $N(a)$;
- $I : P \rightarrow Exp$ é uma função de **inicialização**, onde Exp é da forma: $\forall p \in P : [Type(I(p)) = C(p)_{MS}]$.

Os nomes, quando associados a lugares, não têm significado formal, apenas facilitam a identificação. Já as expressões de guarda associadas às transições são proposições lógicas (expressões booleanas) dentro de colchetes “[...]” que, se verdadeira, habilita o disparo da transição a qual ela se associa. Estas condições, chamadas de “guardas”, também podem estar associadas a limitações impostas pelas transições, visando restringir o conjunto de dados a trabalhar.

Os lugares da RPC podem ser associados a conjuntos específicos de cores. O uso de conjuntos de cor em RPC é análogo ao uso de tipos em linguagem de programação: cada lugar tem um conjunto de cor ligado a este, significando que cada marca presente neste lugar deve ser necessariamente pertencente ao conjunto de cor deste lugar.

A expressão de um arco pode conter variáveis, constantes, funções e operações que são definidas nas declarações da rede. Quando as variáveis de uma expressão de arco são unificadas, isto é são substituídas por cores dos conjuntos de cores correspondentes, a expressão de arco deve ser avaliada para a cor que pertence ao conjunto de cor do lugar ligado ao arco.

Para cada elemento de unificação pode-se verificar se a transição, associada a essa unificação, está habilitada para a marcação corrente. Isso pode ser realizado através da avaliação da expressão de guarda de todos os arcos de entrada. Quando a expressão de guarda está omitida, conclui-se que ela seja verdadeira.

Quando uma transição está habilitada ela pode disparar e então remover as marcas de seus lugares de entrada e adicionar marcas em seus lugares de saída. O número de marcas adicionadas/removidas e suas cores são determinadas pelo valor da expressão de arco correspondente.

Pela disponibilidade de ferramentas de *software* de suporte às Redes de Petri Coloridas (RPC), estas são uma das redes mais utilizadas dentre as redes de alto nível.

2.2 Ambientes Virtuais e Telepresença

Os primeiros trabalhos envolvendo a Realidade Virtual (RV) iniciaram-se na indústria de simulação, com os simuladores de voo que a força aérea do Estados Unidos passou a construir logo após a Segunda Guerra Mundial (JACOBSON, 1994). No entanto, o termo “Realidade Virtual” é creditado a Jaron Lanier, que nos anos 80 juntamente com Thomas Zimmerman desenvolveu a primeira luva de dados, chamada *DataGlove*, usada para manipular dados em aplicações de RV (ZIMMERMAN et al., 1987), além da introdução dos primeiros capacetes de visualização (HMD *head-mounted displays*) de forma comercial (BEIER, 1998). Outros termos relacionados incluem “Ciberespaço” (GIBSON, 1984), “Realidade Artificial” (KRUEGER; GIONFRIDDO; HINRICHSSEN, 1985), e mais recentemente, “Mundos Virtuais” e “Ambientes Virtuais” (FISHER; TAZELAAR, 1990; ELLIS, 1994).

O termo, “Realidade Virtual” é utilizado em uma variedade de formas e, algumas vezes, de uma maneira confusa e distorcida. Beier (1998) afirma que, originalmente, o termo se referia a “Realidade Virtual Imersiva”, uma vez que, a utilização do termo, por Jaron Lanier, era uma forma de diferenciar as simulações computacionais tradicionais de simulações envolvendo usuários imersos em um ambiente artificial, gerado por computador (RHEINGOLD, 1991; ARAUJO, 1996).

Ao longo do tempo, o termo “Realidade Virtual” tornou-se abrangente e passou a ser utilizado não apenas para aplicações imersivas, como também para aplicações parcialmente imersivas ou ainda sem imersão alguma. Um exemplo disso é o sistema QuickTime VR (CHEN, 1995), que utiliza fotografias para a modelagem tridimensional do mundo e fornece as capacidades de um pseudo *look-around*² e *walk-around*³ em um monitor. Termos alternativos como “Mundos Virtuais” ou “Ambientes Virtuais” (AV), refletem mais adequadamente a vasta gama de aplicações possíveis e o fato de a tecnologia ainda estar distante de simular completamente a “realidade”. Sendo, “Ambientes Virtuais” um dos termos preferidos pela comunidade acadêmica (BEIER, 2000).

Steuer (1992), acrescenta que a definição de realidade virtual é baseada em conceitos de “presença” e “Telepresença”, ou seja, a chave para a definição da realidade virtual, em termos de experiência humana ao invés de tecnologias de *hardware* e *software*, é o conceito de presença. Onde a presença, pode ser entendida como a percepção de um ambiente físico, mediada apenas pela mente humana. No entanto, quando a percepção é mediada por alguma

²*Look-around* refere-se a um tipo de interação onde o usuário pode olhar o ambiente virtual em várias direções.

³*Walk-through* refere-se a um tipo de interação onde o usuário pode “andar” ou mover-se pelo ambiente virtual restrito a caminhos ou lugares pré-definidos.

tecnologia de comunicação, a pessoa é obrigada a perceber simultaneamente dois ambientes distintos: o ambiente físico em que o mesmo se encontra presente, e o ambiente apresentado através do meio tecnológico. Logo, o termo “Telepresença” pode ser usado para descrever a última experiência como a sensação de estar em um ambiente mediado por algum meio de comunicação, ao invés do ambiente físico imediato.

Sendo a presença uma percepção natural de um ambiente e a telepresença a percepção de um ambiente mediado, este último pode ser tanto um ambiente real temporalmente ou geograficamente distante como um ambiente virtual sintetizado por computador. Dessa forma, estudos sobre a telepresença têm explorado tecnologias de realidade virtual a fim de orquestrar a percepção humana a partir de um ambiente distante, uma vez que esta proporciona a habilidade de manipular a realidade física remotamente em tempo real, através de sua representação virtual (WILSON, 2003).

O uso do termo “telepresença” foi cunhado por Minsky (1980), em referência aos sistemas de teleoperação para manipulação de objetos físicos remotos. Sheridan (1992) continuou esta tradição, adotando o termo presença (ao invés de telepresença), para uma revista dedicada ao estudo de teleoperação e ambientes virtuais, para referir-se à percepção de estar em um ambiente artificial ou remoto, reservando “telepresença” apenas para os casos que envolvem teleoperação. No entanto, na mesma seção da revista, Held e Durlach (1992) utilizam a expressão “telepresença” para referirem-se à experiência comum a ambos teleoperação e ambientes virtuais. Sendo assim, o termo é amplo e serve para destacar as semelhanças entre teleoperação e ambientes virtuais.

A telepresença, como um refinamento da teleoperação, permite que o indivíduo controle não apenas a simulação mas a própria realidade. Neste contexto, Goldberg (2000) afirma que a essência da telepresença consiste na antipresença, ou seja, o indivíduo não tem que estar fisicamente presente em um local para afetar a realidade do mesmo. Logo, Mair (1997) afirma que isto é muito mais do que uma simples teleconferência, uma vez que permite ao usuário sentir-se imerso em um ambiente remoto e ainda ser capaz, através da teleoperação, de manipular ou controlar eventos remotos.

As fronteiras entre os sistemas de RV e sistemas de telepresença estão cada vez mais tênues e lentamente se fundem em um espectro muito amplo de tecnologias utilizadas para visualizar e interagir com mundos tridimensionais em tempo real. Através das tecnologias de RV e telepresença, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações, possibilitando simulações de situações do mundo real em um ambiente ou mundo virtual. A telepresença e operadores remotos que empregam a RV

tanto para educação, treinamento, simulação como entretenimento, têm sido explorada por cientistas e desenvolvedores de sistemas complexos como apresentados em (HANÁK et al., 2007; VALLIUS; RÖNING, 2006; YANG et al., 2005).

2.2.1 Características dos Ambientes Virtuais

Os ambientes virtuais (AV), assim como qualquer tipo de sistema computacional, necessitam atender a requisitos não-funcionais (SOMMERVILLE; SAWYER, 1997) relacionados a aspectos de *software*, *hardware* ou fatores externos, que determinem condições ou restrições ao comportamento do sistema pretendido. Entre estes requisitos destacam-se:

- **Desempenho:** refere-se ao atendimento de exigências de processamento, tempo de resposta, consumo de recursos, *throughput* e eficiência de processos contidos no escopo do sistema;
- **Usabilidade:** compreende o atendimento de critérios relacionados ao entendimento e facilidade de uso do sistema, além do nível de satisfação dos usuários durante a navegação e interação com o AV;
- **Portabilidade:** normalmente o ambiente virtual desenvolvido é fortemente ligado ao ambiente de desenvolvimento. As aplicações devem comportar facilidades para execução em diferentes plataformas e/ou sistemas operacionais;
- **Flexibilidade:** a forma em que o ambiente foi desenvolvido deve permitir que possíveis alterações, assim como a remoção e inserção de novos dispositivos seja implementada com o mínimo possível de alteração de código.

Além dos aspectos destacados anteriormente, Stuart (1996) e Vince (2004) acrescentam que um AV agrega peculiaridades adicionais que podem caracterizá-lo como um ambiente:

- **Sintético:** o AV é gerado em tempo real por um sistema computacional, e não pré-gravado como em sistemas multimídia;
- **Multisensorial:** significa que o ambiente pode incluir recursos que estimulem diferentes sentidos humanos, como visão, audição, tato, senso de espaço, profundidade, etc;

- **Tridimensional:** o ambiente é representado em 3 dimensões e, além disso, existem recursos que dão a idéia de que o ambiente possui profundidade e possibilita movimento através dele;
- **Imersivo:** o sistema deve permitir que o usuário sinta-se “dentro” do mundo virtual, seja com o seu corpo físico ou com uma representação qualquer (avatar⁴, vídeo, simulação de cabine, etc). É importante que, de alguma forma, o usuário seja “envolvido” pelo ambiente;
- **Interativo:** corresponde à capacidade do sistema detectar as entradas do usuário e, instantaneamente, promover alterações ou ações em cenas e objetos do ambiente virtual;
- **Realístico:** determina a precisão com que o AV reproduz os objetos reais, as interações com os usuários e o próprio modelo do ambiente;
- **Com Presença:** permitir a sensação de estar presente fisicamente em um ambiente artificial.

Os AV também podem ser caracterizados como a coexistência integrada de três idéias básicas: imersão, interação e envolvimento (MORIE, 1994). Embora não seja necessário que as três coexistam em simultâneo. É importante ressaltar que nem todo AV é imersivo (LESTON, 1996; ROBERTSON; CARD; MACKINLAY, 1993). A visualização de uma cena 3D em um monitor convencional, por exemplo, é considerada não-imersiva. Na RV não-imersiva o usuário é transportado, parcialmente, para um mundo virtual, através de uma projeção ou um monitor, mas continua a sentir-se, predominantemente, no mundo real. Entretanto, Robertson, Card e Mackinlay (1993) comentam que dispositivos baseados nos demais sentidos humanos, podem introduzir algum grau de imersão inclusive à AV's em monitores.

A idéia de imersão está ligada ao sentimento de fazer parte do ambiente - o usuário vê-se envolvido no domínio da aplicação (realidade), tendo uma sensação de presença dentro do mundo virtual. Normalmente, um sistema imersivo é obtido com o uso de capacete de visualização, ou cavernas virtuais - sistemas imersivos baseados em salas com projeções das visões nas paredes, teto, e piso (CRUZ-NEIRA, 1995). Além do fator visual, dispositivos ligados aos demais sentidos também são importantes para o sentimento de imersão, como o som

⁴Avatares são representações visuais de usuários ou agentes autônomos dentro do mundo virtual que são controlados; podem ser imagens tridimensionais ou vídeos que servem como entrada padrão para o usuário que o controla.

(BEGAULT, 1994), o posicionamento automático da pessoa e dos movimentos da cabeça, controles reativos, etc.

A interação está ligada à capacidade do computador detectar as entradas do usuário e modificar instantaneamente o mundo virtual em função das ações efetuadas sobre ele (capacidade reativa). As pessoas são cativadas por uma boa simulação em que as cenas mudam em resposta aos seus comandos, que é característica mais marcante dos vídeo games. Para que um sistema de RV pareça mais realista, o ambiente virtual inclui objetos simulados. Outros artifícios para aumentar o realismo são empregados, por exemplo, a texturização dos objetos do ambiente e a inserção de sons tanto ambientais quanto sons associados a objetos específicos (ARAUJO, 1996).

A idéia de envolvimento, por sua vez, está ligada ao grau de motivação para o engajamento de uma pessoa em determinada atividade. O envolvimento pode ser passivo, como ler um livro ou assistir televisão, ou ativo, ao participar de um jogo com algum parceiro. A RV tem potencial para os dois tipos de envolvimento ao permitir a exploração de um ambiente virtual e propiciar a interação do usuário com o mundo virtual dinâmico. Embora a percepção visual seja nosso sentido primário, outros sentidos também devem ser estimulados para proporcionar uma completa imersão; entre os quais o retorno auditivo, o tato e a força de reação.

Sob o conceito de ambientes virtuais alberga-se um vasto número de tecnologias, na sua maioria constituindo tecnologias recentes ainda sujeitas a novos desenvolvimentos sem que tenha-se, ainda, uma percepção muito clara do verdadeiro alcance que as mesmas podem ou virão a deter. Dessa forma, a seção seguinte apresenta algumas classes de RV, com o intuito de obter-se uma melhor compreensão sobre os diferentes tipos de sistemas de RV.

2.2.2 Tipos de Sistemas de Realidade Virtual

Como descrito anteriormente, Realidade Virtual (RV) é um termo usado para ambientes 3D gerados por computador, que possibilitam ao usuário entrar e interagir em um ambiente sintético. Para Draper, Kaber e Usher (1999), um ambiente sintético permite que as capacidades perceptivas, cognitivas e psicomotoras humanas sejam projetadas em um ambiente distante, hostil ou simulado.

Assim, Capin et al. (1999), a definem como uma tecnologia capaz de transportar um indivíduo para um ambiente diferente do real, sem movê-lo fisicamente, manipulando as informações destinadas aos órgãos sensoriais humanos de tal maneira que o ambiente percebido

seja associado ao Ambiente Virtual (AV) desejado e não ao Ambiente Real (AR).

Diversos artigos e livros que abordam os conceitos e definições sobre realidade virtual ou ambiente virtual (KALAWSKY, 1993; LATTA; OBERG, 1994; EARNSHAW; VINCE; JONES, 1995; VINCE, 2004; KIRNER; TORI, 2004). Como mencionado, há várias definições aceitas, o que é devido, em parte, à natureza interdisciplinar da área e também à sua evolução, pois de uma maneira ou de outra, os sistemas de RV acabaram vindo de sistemas computacionais de mesa, simuladores e sistemas de teleoperação, entre outros (KIRNER; TORI, 2004).

Os sistemas de RV diferem entre si de acordo com os níveis de imersão e de interatividade proporcionado ao participante. Esses níveis são determinados pelos tipos de dispositivos de entrada e saída de dados do sistema, além da velocidade e potência do computador que o hospeda. Ainda não existe um critério claro de classificação dos sistemas de RV.

De acordo com o nível de interação, os sistemas de RV podem ser classificados como passivos, exploratórios ou interativos (ADAMS, 1994). Quanto ao grau de imersão, os sistemas ou estilos de RV podem ser classificados como RV de Simulação, RV de Projeção, *Visually Coupled Displays* ("Visores Visualmente Acoplados"), RV de Mesa, Realidade Aumentada, Virtualidade Aumentada e Hiper-realidade (PIMENTEL; TEIXEIRA, 1994; MILGRAM et al., 1994; TIFFIN; TERASHIMA, 2001). As duas próximas seções apresentam as classificações de RV com relação ao grau de interação e imersão do usuário com o AV de forma mais detalhada.

2.2.2.1 Grau de Interação Proporcionado do Usuário

Um ambiente virtual é composto por um determinado espaço e situações, ambos delimitados, incluindo todos os componentes neles inseridos, como o conjunto de condições e objetos possíveis de serem percebidos e com os quais é possível interagir. Um ambiente virtual é, portanto, um ambiente interativo gerado por computador e disponibilizado através de um sistema de realidade virtual .

Como mencionado anteriormente, os sistemas de RV podem ser classificados de acordo com o modelo de participação do usuário, ou seja, de acordo com o nível de interação que este sistema oferece. Assim, Adams (1994) sugere que um aplicativo de RV pode proporcionar uma sessão sob três formas diferentes:

- **Passiva:** uma sessão de RV passiva proporciona ao usuário uma exploração do ambiente automática e sem interferência. A rota e os pontos de observação são explícitos e controlados exclusivamente pelo software. O usuário não tem controle

algum, exceto talvez, para sair da sessão;

- **Exploratória:** uma sessão de RV exploratória proporciona uma exploração do ambiente dirigida pelo usuário. O participante pode escolher a rota e os pontos de observação, mas não pode interagir de outra forma com entidades contidas na cena;
- **Interativa:** uma sessão de RV interativa proporciona uma exploração do ambiente dirigida pelo usuário e, além disso, as entidades virtuais do ambiente respondem e reagem às ações do participante. Ou seja, ela oferece todas as possibilidades que eram restritas nas participações passiva e exploratória: o usuário tem liberdade de escolher a rota, o ponto de vista e interagir com os objetos do ambiente.

A navegação em ambientes virtuais é controlada pelo posicionamento do observador no mundo virtual, o chamado ponto de observação (*viewpoint*). A navegação é, provavelmente, a forma mais simples de interação encontrada em muitas aplicações de RV. Os dispositivos, associados a técnicas e linguagens computacionais, viabilizam a interação do usuário com a aplicação, permitindo a exploração do ambiente virtual e a modificação de objetos de maneira fácil e rápida - por exemplo, apontar, pegar, arrastar, rotacionar e visualizar os objetos sob diferentes pontos de vista.

2.2.2.2 Grau de Imersão Proporcionado ao Usuário

Dada a natureza artificial dos ambientes virtuais, é possível que os usuários e os elementos do próprio ambiente comportem-se como os complementos físicos do mundo real (HARRISON; DOURISH, 1996). Os usuários podem ser representados por avatares, que são objetos sintéticos inseridos no ambiente virtual, que podem navegar livremente nesse espaço, encontrar artefatos, objetos ou dados e interagir utilizando canais visuais de comunicação (CHURCHILL; SNOWDON, 1998). Logo, existem diferentes formas de inserção ou imersão do usuário em relação ao ambiente ou mundo virtual.

Assim, Pimentel e Teixeira (1994) consideram que, de acordo com o grau de imersão, os sistemas ou estilos de RV podem ser classificados como RV de Simulação, RV de Projeção, *Visually Coupled Displays* ("Visores Visualmente Acoplados"), RV de Mesa e Realidade Aumentada. Milgram et al. (1994) abordam uma classe chamada Realidade Misturada como generalização da Realidade Aumentada e da Virtualidade Aumentada. Enquanto, Tiffin e Terashima (2001) descrevem uma outra classe de RV, chamada Hiper-realidade, como um paradigma para o terceiro milênio.

A RV de Simulação corresponde ao tipo mais antigo, originado com os simuladores de vôo desenvolvidos pelos militares americanos após a Segunda Guerra Mundial (JACOBSON, 1994). Um sistema desse tipo basicamente imita o interior de um carro, avião ou jato, colocando o participante dentro de uma cabine com controles. Na cabine, telas de vídeo e monitores apresentam um mundo virtual que reage aos comandos do usuário. Uma vez que o sistema de RV de Simulação não processa imagens em estéreo, as imagens são geradas de forma bastante rápida. Em alguns sistemas as cabines são montadas sobre plataformas móveis, e os controles oferecem *feedback* tátil e auditivo.

A RV de Projeção também é conhecida como Realidade Artificial, e foi criada nos anos 70 por Myron Krueger. Na RV de Projeção o usuário está fora do mundo virtual, mas pode se comunicar com personagens ou objetos nele contidos. O sistema VIDEOPLACE (KRUEGER; GIONFRIDDO; HINRICHSEN, 1985), capturava a imagem do(s) usuário(s) e projetava-a em uma grande tela que representava um mundo virtual no qual os usuários podiam interagir uns com os outros ou com objetos. Krueger usou o termo Realidade Artificial para descrever o tipo de ambiente criado pelo seu sistema, que não exigia que o participante vestisse ou usasse dispositivos de entrada de dados (JACOBSON, 1994).

Os Visores Visualmente Acoplados (*Visually Coupled Displays*) correspondem a uma classe de sistemas na qual as imagens são exibidas diretamente ao usuário, que está olhando em um dispositivo que deve acompanhar os movimentos de sua cabeça. Esse dispositivo geralmente permite imagens e sons em estéreo, além de conter sensores especiais que detectam a movimentação da cabeça do usuário e usam essa informação para realimentação da imagem exibida (PIMENTEL; TEIXEIRA, 1994).

A RV de Mesa (*Desktop VR*) é um subconjunto dos sistemas tradicionais de RV em que, ao invés de capacetes de visualização (*headmounted displays* - HMD) são utilizados grandes monitores ou algum sistema de projeção para apresentação do mundo virtual. Alguns sistemas permitem ao usuário ver imagens 3D no monitor com óculos obturadores (*shutter glasses*), polarizadores ou filtros coloridos (PIMENTEL; TEIXEIRA, 1994).

A Realidade Aumentada é usualmente definida como a sobreposição de objetos virtuais tridimensionais, gerados por computador, no ambiente real, por meio de algum dispositivo tecnológico (AZUMA, 1993; BOMAN, 1995). Onde, o ambiente predominante é real, porém enriquecido com objetos virtuais que sobrepõem o mesmo. Além de permitir que objetos virtuais possam ser introduzidos em ambientes reais, a Realidade Aumentada proporciona também ao usuário, o manuseio desses objetos com as próprias mãos, possibilitando uma interação atrativa e motivadora com o ambiente (BILLINGHURST; KATO; POUPYREV, 2001;

ZHOU et al., 2004). No entanto, para que os objetos virtuais façam parte do ambiente real e sejam manuseados, deve-se utilizar um software com capacidade de visão do ambiente real e de posicionamento dos objetos virtuais, além de acionar dispositivos tecnológicos apropriados para Realidade Aumentada. Como exemplo, pode-se utilizar dispositivos visuais transparentes presos à cabeça (*heads-up-displays* (HUDs)) nos quais os dados são projetados. Pelo fato desses *displays* ou visores serem transparentes, o usuário pode ver dados, diagramas, animações e gráficos 3D sem deixar de enxergar o mundo real.

A Virtualidade Aumentada é a inserção de representações de elementos reais no mundo virtual, usando uma interface que permite ao usuário interagir com o ambiente virtual (KIRNER; KIRNER, 2007). Logo, sua principal característica consiste na predominância de um ambiente virtual envolvente. Além dos objetos estáticos pode-se, também, levar objetos reais dinâmicos para o cenário (ambiente virtual). Estes elementos reais podem ser pré-capturados ou ser capturados em tempo real por câmeras, neste caso o processo computacional faz a sua identificação e respectiva reconstrução no ambiente virtual. Milgram et al. (1994) acreditam que a conceituação de Realidade Aumentada e Virtualidade Aumentada é muito geral e só fica clara com sua inserção em um contexto mais amplo: o da Realidade Misturada.

A Realidade Misturada, misturando 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. Dessa forma, Milgram et al. (1994) descrevem estes conceitos, em um contexto de Realidade-Virtualidade contínua, onde existe uma relação contínua entre o ambiente puramente virtual e o ambiente puramente real. Logo, a Realidade Misturada nada mais é que uma generalização da Virtualidade Aumentada e da Realidade Aumentada.

Por fim, a Hiper-realidade é a capacidade tecnológica de combinar realidade virtual, realidade física, inteligência artificial e inteligência humana, integrando-as de forma natural para acesso do usuário (TIFFIN; TERASHIMA, 2001). Na Hiper-realidade o virtual é combinado com o real, incrementando-se novos elementos para facilitar e potencializar a interação do usuário com os recursos de que necessita-se para realização das tarefas.

2.2.3 Dispositivos de Realidade Virtual

Um convincente mundo virtual deve parecer natural aos participantes e a interação deve ser tanto intuitiva e transparente quanto possível. Para que isto seja alcançado, os sistemas de RV utilizam diversos dispositivos que permitam maximizar o grau de interação e imersão do usuário com o ambiente. Estes dispositivos, de modo geral, permitem a captura

ou rastreamento de movimentos, além de fornecerem estímulos aos sentidos humanos. Logo, os dispositivos de RV atuam basicamente de duas formas: captando as entradas do usuário (geralmente movimentos do corpo) e estimulando sentidos como visão, audição e tato.

2.2.3.1 Dispositivos de Rastreamento

Também conhecidos como *tracking devices* ou ainda *motion trackers*, os dispositivos de rastreamento ou de trajetória permitem que um sistema de realidade virtual possa acompanhar a posição e orientação de partes selecionadas do corpo do usuário. Muitos dispositivos visuais e de interação incorporam algum dispositivo de rastreamento, a fim de medir a posição e orientação da parte do corpo ao qual estão associados, para ampliar a interação e a sensação de imersão em um AV.

Com base nas informações recebidas de um dispositivo de rastreamento, pode-se definir a perspectiva do usuário no mundo virtual, determinando qual parte do AV que deve ser “renderizada” para a visualização, bem como a geração de estímulos táteis e auditivos. Um *motion tracker* anexado a um HMD, por exemplo, mede a posição e orientação da cabeça continuamente para que a imagem gerada pelo computador se ajuste à representação da cena para a visão atual. Enquanto em uma luva, mede a posição e orientação da mão para que as respostas necessárias para a manipulação de objetos virtuais seja a mais natural possível.

Os dispositivos de rastreamento operam baseados na diferença de posição ou orientação em relação a um ponto ou estado de referência, de maneira ativa ou passiva. Enquanto no *tracking* passivo o alvo é rastreado a partir de uma ou mais câmeras que monitoram os movimentos do usuário a uma certa distância, no *tracking* ativo, são anexados sensores ao alvo a ser rastreado. Em termos de *hardware*, estes dispositivos geralmente são formados por três componentes: uma fonte que gera um sinal, um sensor que recebe o sinal e um controle para processar o mesmo e comunicar-se com o computador. Dependendo da tecnologia empregada, ou a fonte ou o sensor é anexada ao corpo e a outra parte a um local fixo no ambiente, servindo como ponto de referência. Assim, pode-se medir as coordenadas de posição (*x*, *y*, *z*) e as coordenadas de orientação (*yaw*, *pitch*, e *roll*) obtendo dados suficientes para operar com dispositivos de interação com 6DOF (seis graus de liberdade).

Para localizar os movimentos do usuário com precisão dentro do espaço virtual, os atuais dispositivos de rastreamento são baseados em tecnologias mecânicas, acústicas, ópticas ou eletromagnéticas, cuja apresentação de cada uma destas abordagens segue abaixo:

- **Rastreadores Mecânicos:** estes dispositivos medem a posição e orientação

através de uma conexão mecânica entre um ponto de referência e o alvo. Tipicamente, um braço mecânico articulado, conecta uma caixa de controle à cabeça, e codificadores localizados nas juntas do braço medem a mudança de posição e orientação com relação ao ponto de referência.

- **Rastreadores Acústicos:** os rastreadores acústicos utilizam ondas sonoras ultra-sônicas (alta frequência) para medir a posição e orientação do objeto alvo. Com o uso de transmissores e receptores, pode-se medir o tempo de viagem (*time-of-flight*) do som entre a fonte emissora dos pulsos ultra-sônicos e a captação por um conjunto de receptores dispostos usualmente de maneira triangular e, assim, calcular a distância do alvo para cada um dos sensores.
- **Rastreadores Óticos:** estes rastreadores usam uma combinação de marcadores, tais como diodos emissores de luz (LED's), câmeras de vídeo e técnicas de processamento de imagens. Uma variante utiliza algumas mini-câmeras que são posicionadas sobre o alvo a ser rastreado (cabeça ou mão do usuário) e um conjunto de LED's (diodos emissores de luz) são fixados em uma estrutura acima do mesmo (no teto por exemplo). Em uma outra variante, ocorre o inverso, as câmeras são montadas no teto, ou uma estrutura fixa, e alguns LED's são colocados em posições fixas e conhecidas sobre o alvo. Em ambos os casos, as projeções dos LED's sobre os planos das imagens das câmeras permitem identificar a posição e orientação do objeto alvo que está sendo rastreado.
- **Rastreadores Eletromagnéticos:** os rastreadores eletromagnéticos possuem um transmissor ou fonte que emite campos eletromagnéticos ao longo de três eixos ortogonais que são detectados por um ou mais sensores. A partir dos sensores pode-se determinar o tamanho e a direção destes campos e, por conseguinte, determinar a posição e orientação da fonte emissora (geralmente o usuário).

2.2.3.2 Dispositivos de Interação

As aplicações de *desktop* baseadas no paradigma WIMP (*Windows, Icons, Menus and Pointing Device*) usam dispositivos de entrada convencionais como o teclado, *mouse* e *joystick*. Estes dispositivos interativos básicos são fáceis de usar, mas eles também tendem a atrapalhar a sensação de imersão. Para que o usuário interaja com o sistema sem que perceba os dispositivos envolvidos, os pesquisadores e a indústria de RV procuram desenvolver dispositivos mais intuitivos como luvas, roupas, bolas isométricas, sensores biológicos, esteiras e esferas.

Os dispositivos isométricos ou bolas isométricas, em geral, consistem de uma bola sobre uma plataforma com botões que são configurados via *software*. Parecidos com um *mouse* ou um *joystick*, são bastante fáceis de manipular. Os sensores de entrada biológicos processam atividades denominadas indiretas, como comando de voz e sinais elétricos musculares. Em sistemas de RV o reconhecimento de comandos de voz pode facilitar a execução de tarefas no mundo virtual, principalmente quando as mãos estiverem ocupadas em outra tarefa e impossibilitadas de acessar o teclado. Já os dispositivos que utilizam sinais elétricos musculares detectam a atividade muscular por meio de eletrodos colocados sobre a pele, permitindo ao usuário movimentar-se no mundo virtual pelo simples movimento dos olhos, por exemplo.

Muitas vezes chamadas de *datagloves* ou *powergloves*, apesar de ambas as expressões se referirem a modelos particulares de luvas para interação em AV's, os diferentes tipos de luvas de RV compartilham o mesmo propósito: permitir ao usuário manipular dados de computador de uma forma intuitiva. Usando uma luva com fios, o usuário pode interagir com objetos virtuais fazendo vários gestos manuais como se estivesse interagindo com objetos reais. Geralmente o usuário é capaz de ver sua mão virtual no mundo virtual no qual se encontra inserido e assim poder coordenar seus movimentos naturalmente.

Algumas luvas medem a extensão do dedo através de uma série de cabos de fibra óptica. A luz passa através dos cabos a partir de um emissor para um sensor. A quantidade de luz que atinge o sensor muda dependendo de como o usuário mantém seus dedos: se ele curva seus dedos para o punho, menos luz atinge o sensor, que, por sua vez, envia esses dados ao computador do sistema de RV. Em geral, esse tipo de luva precisa ser calibrado para cada usuário para que funcione adequadamente. Outras luvas usam faixas de material flexível revestidas com uma tinta condutora de eletricidade para medir a posição do dedo do usuário. À medida que o usuário dobra ou estende seus dedos, a resistência elétrica ao longo da faixa muda. O computador interpreta as mudanças na resistência e responde apropriadamente. Essas luvas são menos precisas que as luvas de fibra óptica, mas também costumam ser mais baratas. Para maior precisão e sensibilidade, foram desenvolvidas as luvas do tipo *dexterous hand master* (DHM). A DHM usa sensores anexados a cada junta dos dedos. O usuário conecta os sensores à cada uma de suas juntas através de *links* mecânicos, o que significa que a luva se parece com um exoesqueleto. Essas luvas são mais precisas do que as luvas de fibra óptica ou aquelas que usam material condutor de eletricidade, mas também são incômodas e desajeitadas.

Na tentativa de aumentar a sensação de imersão do usuário durante a navegação no ambiente virtual, alguns dispositivos de interação oferecem recursos para que o usuário

possa caminhar dentro do AV sem sair de um local fixo (BOUGUILA; ISHII; SATO, 2002). Nesse contexto, dois dispositivos têm sido desenvolvidos e explorados recentemente: a esteira e a esfera.

Uma esteira é útil porque o usuário permanece parado em relação ao mundo real, mas sente-se como se estivesse, de fato, andando em um ambiente virtual. Uma limitação óbvia das esteiras normais é que você só pode andar em duas direções: para trás e para frente. Assim, algumas empresas desenvolveram esteiras multidirecionais, que ao contrário das esteiras normais que usam um único motor para exercer força tanto para frente quanto para trás em relação ao usuário, as esteiras multidirecionais usam dois motores: da perspectiva do usuário, a esteira pode exercer força para frente, para trás, para a esquerda e para a direita. Um exemplo desta tecnologia é o *CirculaFloor* desenvolvido por Iwata et al. (2004), que consiste num conjunto de placas móveis, equipadas com um mecanismo holonômico que propicia um movimento omnidirecional, como pode ser visto na Figura 4. O movimento dos pés do usuário é acompanhado por sensores de posição: o piso se move na direção oposta da direção do seu andar, de forma que o deslocamento do passo é cancelado. Seguindo esta linha de pesquisa, Iwata, Yano e Tomiyoshi (2007) ainda desenvolveram a *String Walker*, uma interface de locomoção que utiliza oito cordas acionadas por mecanismos de motor-polia montados sobre uma placa giratória em forma de anel, que permite ao usuário andar em um ambiente virtual em diversas direções, enquanto a sua posição é mantida.

Figura 4: Dispositivo de interação do tipo esteira chamado *CirculaFloor* (IWATA et al., 2004).

A empresa VirtuSphere, Inc.⁵ oferece uma maneira ainda mais versátil de locomoção dentro de um ambiente virtual. O dispositivo consiste de uma esfera de plástico de gigante que se parece com uma roda de *hamster* em tamanho humano. Ao entrar na esfera, o usuário portando um capacete de visualização (HMD) pode andar, deitar, rolar, pular ou se arrastar

⁵VIRTUSPHERE - A New Locomotion Technology for Immersing in Virtual Reality - Disponível em: <<http://www.virtusphere.com>>

sobre ela para locomover-se pelo mundo virtual. A esfera, que pode ser vista na Figura 5, é posicionada sobre uma plataforma estável que tem várias rodas que se apóiam contra a mesma, o que permite que ela gire em qualquer direção ao mesmo tempo que permanece em uma posição fixa. Sensores nas rodas informam ao computador em qual caminho o usuário está andando e a visão dentro do HMD do usuário muda de acordo com o caminho. Outras tecnologias semelhantes, como *Cybersphere* e o *Cyberwalk* são descritas respectivamente em (FERNANDES; RAJA; EYRE, 2003; SCHWAIGER; THÜMMEL; ULBRICH, 2007).

Figura 5: Dispositivo de interação do tipo esfera da VirtuSphere.

2.2.3.3 Dispositivos visuais

Os dispositivos visuais são dispositivos que apresentam visualmente para o usuário o mundo 3D e influenciam fortemente a percepção do nível de imersão de um sistema de RV. Alguns autores chegam a dividir ou classificar as pesquisas sobre RV de acordo com os dispositivos de visualização empregados (CRUZ-NEIRA, 1995). Com base nos dispositivos visuais, os sistemas de RV podem ser monoscópicos ou estereoscópicos: no primeiro caso, uma única imagem é “renderizada”⁶ e exibida para os dois olhos, no segundo, cada olho observa uma imagem ligeiramente diferente, cada qual “renderizada” separadamente (PIMENTEL; TEIXEIRA, 1994).

Os sistemas de exibição estereoscópica ou 3D permitem ao observador/usuário ver os objetos ou o ambiente gerado além das dimensões da tela do computador, acrescentando profundidade à visualização e, conseqüentemente, tornando-a mais realista e próxima da forma que o mundo real é visto. Em linhas gerais, pode-se dizer que a estereoscopia, conceito no

⁶“Renderização” é uma adaptação o termo em inglês “rendering” e, refere-se ao processo de geração de uma imagem a partir de um modelo, através de técnicas de computação gráfica.

qual esses sistemas de exibição são baseados, trabalha com o fato de que cada um dos olhos humanos, estando ligeiramente afastados um do outro, vê imagens ligeiramente diferentes. Essas imagens são reunidas no cérebro e compreendidas com características de profundidade, distância, posição e tamanho (MACHADO, 2003).

Para a exibição estereoscópica das imagens existem diversos dispositivos utilizados, sendo que a escolha de um determinado tipo de sistema de exibição dependerá da finalidade do sistema de RV. Pimentel e Teixeira (1994) separam estes dispositivos em duas classes: a primeira composta pelos capacetes de visualização (*head-mounted displays*) e os dispositivos binoculares suspensos *head-coupled displays* ou *arm-mounted displays* e, a segunda composta pelos monitores de vídeo e telas de projeção. O que diferencia essas duas classes é que, na primeira, o dispositivo possui sensores para detectar os movimentos do usuário, enquanto que na segunda isso não ocorre, e o rastreamento depende dos comandos do usuário via outro dispositivo de entrada.

Os capacetes de visualização ou *head-mounted displays* (HMD's) são dispositivos para serem acoplados/vestidos à cabeça e são constituídos de duas telas/monitores pequenos, um à frente de cada olho, um conjunto de lentes especiais e um sistema óptico que canaliza as imagens aos olhos do observador. As lentes ajudam a focalizar imagens que estão a alguns milímetros dos olhos do usuário, ajudando também a ampliar o campo de visão do vídeo auxiliando na apresentação estéreo de um mundo virtual. Geralmente integram sistemas de áudio e detecção de trajetória, com sensores de rastreamento que medem a posição e orientação da cabeça. A imagem do mundo virtual é mostrada de acordo com o ponto de visão do usuário, enquanto o mesmo explora o ambiente. A medida que o computador recebe os dados de posição e orientação, é gerado uma seqüência de imagens por quadro correspondente às ações e perspectivas do usuário, fornecendo um alto grau de imersão ao mesmo (GRADECKI, 1995).

Outra forma de imergir num ambiente de Realidade Virtual é utilizando os dispositivos binoculares suspensos (*head-coupled displays*), que consiste em um par de lentes binoculares montado sobre um braço mecânico articulado, com sensores ligados a este braço e controles próximos ao *display* (lentes) (PIMENTEL; TEIXEIRA, 1994). Também conhecidos como *arm-mounted displays* (AMD's), estes dispositivos integram detecção de movimento no braço mecânico que os suporta, permitindo movimentos em até 6 graus de liberdade, com movimentos restritos pelo tamanho do braço (ARAUJO, 1996). O formato do binoculares suspensos, Figura 6, permite uma transição entre a visualização do mundo virtual e a interação com teclados, monitores e outros dispositivos que possam auxiliar no controle da simulação.

Figura 6: Dispositivo binocular suspenso.

Os dispositivos visualização baseados em monitores de vídeo representam a categoria de dispositivos do tipo *desktop displays*. Este paradigma evoluiu-se a partir dos monitores de computadores gráficos tradicionais, onde estações de trabalho gráficas acopladas com acessórios para as mãos e cabeça permitem que as imagens exibidas na tela respondam aos movimentos do usuário na frente do monitor. Diretamente derivado das aplicações gráficas computacionais interativas tradicionais, este sistema posiciona o observador externamente ao ambiente virtual, usando a tela do monitor como uma janela para o mundo virtual, por conseguinte provendo um sentido muito limitado de imersão (FISHER, 1991).

Quanto aos sistemas baseados no uso de telas de projeção, estes podem ser de constituídos de uma tela simples ou múltiplas telas. Os sistemas de projeção baseado em tela simples, em sua maioria, usam a metáfora de uma mesa (*desk-style display*), na qual os objetos são projetados em sua superfície. Em geral, sistemas de tela simples são boas opções para aplicações que requerem manipulação de objetos diretamente localizados na frente do usuário. Estes sistemas são menores que as cavernas digitais (múltiplas telas), o que oferece uma certa mobilidade no sentido de que eles podem ser facilmente desmontados e transportados. Neste tipo de dispositivo, o usuário ficam em frente a uma tela vertical inclinável, como uma tela de pintura, com tamanhos típicos entre 150-240 cm, onde o usuário pode interagir com os objetos virtuais projetados sobre e “acima” da tela (HOFFMANN; STEFANI; PATEL, 2006). Um grande exemplo deste tipo de sistema é o ImmersaDesk (CZERNUSZENKO et al., 1997).

Os sistemas de projeção baseado em múltiplas telas surgiram com uma nova proposta de interface para sistemas de RV. Neste tipo de sistema as imagens são projetadas em telas dispostas de uma maneira que envolvam o observador ou usuário do sistema, dando a sensação de imersão. Um grande exemplo deste sistema de projeção é o ambiente de realidade virtual CAVE (*Cave Automatic Virtual Environment*), onde a ilusão de imersão é criada pela projeção de gráficos estereoscópicos de computador nas paredes e chão de uma sala em forma de cubo,

que envolve completamente o usuário em uma espécie de “caverna” digital (CRUZ-NEIRA, 1995). As paredes, teto e chão são telas semitransparentes onde as imagens são projetadas a partir de projetores posicionados atrás das telas. Sistemas do tipo CAVE também incorporam projeção acústica tridimensional, dispositivos de rastreamento de posição e de interação, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual e interajam com os objetos virtuais projetados neste ambiente. A projeção das imagens é feita por projetores posicionados atrás das telas e pode ser estereoscópica, exigindo dos usuários o uso de óculos obturadores ou estereoscópicos (*shutter/stereoscopic glasses*). Mais detalhes a respeito desse sistema serão abordados na seção 2.2.3.6.

Tanto para os monitores como para as telas de projeção, existem três processos principais utilizados para a exibição de imagens estereoscópicas: anaglifo, luz polarizada ou luz intermitente. No processo por anaglifo as imagens de cada olho são exibidas em cores complementares, como vermelho e azul (ou vermelho e verde). As imagens são observadas com óculos que tem a mesma correspondência de cores (os chamados filtros), permitindo a cada olho ver a sua respectiva imagem. Esta técnica é empregada em alguns tipos de cinema 3D, onde uma componente direita da imagem, geralmente na cor vermelha, é sobreposta à componente esquerda em uma cor contrastante, geralmente o verde ou azul, para produzir um efeito 3D quando o observador olha através dos filtros coloridos (óculos). No entanto, é o conhecido que o uso prolongado desses óculos cansam os olhos e causam dor de cabeça (LIPTON, 1982; VINCE, 2004). No processo por luz polarizada, as imagens do par estereoscópico são projetadas em planos ortogonais, e no processo por luz intermitente cada imagem do par estereoscópico é apresentada de forma alternada em uma frequência rápida e imperceptível ao ser humano. Por apresentarem duas imagens diferentes no mesmo plano de exibição, independente do método de exibição, estes dispositivos exigem o uso de algum tipo de dispositivo periférico de visualização pelo usuário para a separação das imagens, tal como óculos coloridos (ou de filtros), óculos obturadores (ou estereoscópicos) (*shutter/stereoscopic glasses*) ou óculos polarizadores (NETTO; MACHADO; OLIVEIRA, 2002).

Sistemas baseados no uso de monitores auto-estereoscópicos também já estão disponíveis (KAUFMANN; LIU; BURRIS, 2000; DODGSON, 2005). Os monitores auto-estereoscópicos por sua vez já têm um filtro aplicado à frente desdobrando a visualização da imagem de forma a apresentar uma imagem diferente para cada olho quando olhamos para qualquer ponto do monitor. Assim, esses dispositivos auto-estereoscópicos permitem visualização estéreo ou 3D sem a necessidade do uso de óculos especiais ou algum outro periférico adicional para acoplar à cabeça (BENTON, 2001).

2.2.3.4 Dispositivos Auditivos

Vivemos em um mundo com um fluxo constante de sons que nos dizem muito sobre o meio envolvente. O eco e reverberação, por exemplo, nos dão pistas sobre a direção e distância de objetos, além de uma noção a respeito das proporções do espaço físico ao qual estamos inseridos. Através do som, podemos localizar objetos que estão fora de nossa vista, como um controle remoto ou um celular. Não obstante, o som nos oferece informações sobre o tipo de material, a textura e a dureza de uma superfície. Logo, a habilidade de sintetizar o som espacial aliada a informação visual traz muitos benefícios para a realidade virtual imersiva.

Entre os sistemas de áudio existentes, encontram-se os sistemas para espacialização e os sistemas de auralização. Os primeiros sugerem campos sonoros envolventes e uma sensação de espacialidade, enquanto os últimos têm como objetivo primário simularem ou reproduzirem com o máximo de fidelidade possível campos sonoros 2D/3D legítimos. Dessa forma, os sistemas de auralização são usualmente mais precisos na capacidade de prover a percepção de direção e distância (posição) dos objetos sonoros dispostos na cena, bem como reproduzir com mais realismo as características acústicas de uma cena (FARIA, 2005).

A fim de adquirir uma compreensão clara do som espacial, é importante distinguir o som mono, estéreo, *surround* e binaural de som 3D. Uma gravação de som mono é uma gravação de som com um microfone. Não existe um senso de posicionamento do som presente neste tipo de gravação. O som estéreo é gravado com dois microfones distantes um do outro, separados por um espaço vazio (ar) simulando o posicionamento dos ouvidos. Quando uma gravação estéreo é reproduzida, a gravação de um microfone entra no ouvido esquerdo, enquanto que a gravação a partir do outro microfone é canalizada para o ouvido direito. Isto dá uma idéia da posição em que o som foi gravado pelos microfones. No entanto, geralmente tem-se a sensação de que a fonte emissora do som esteja dentro da cabeça do ouvinte - isto acontece porque os seres humanos geralmente não ouvem os sons da maneira como eles foram gravados em estéreo, separados por um espaço vazio, e a cabeça humana atua como um filtro na captura dos sons. No som envolvente (*surround*), adiciona-se mais microfones e mais alto-falantes para adquirir uma melhor ilusão de posicionamento do som, no entanto ainda não é possível obter-se a localização precisa dos objetos sonoros.

A maior parte dos sistemas comerciais em utilização no mercado presta-se mais à criação de efeitos *surround* e espacialização para filmes e entretenimento de modo geral do que propriamente à reprodução de campos sonoros 2D/3D coerentes. Uma vez que, para este mercado, não é importante a localização precisa dos objetos sonoros. Já para os sistemas de realidade virtual, somente a espacialização de eventos sonoros não é suficiente para impor o

realismo necessário. Falta a percepção da diretividade e da localização dos objetos no espaço, sugerindo o uso de sistemas para geração de sons 3D.

Faria (2005) destaca que entre os métodos de síntese de áudio tridimensional (3D), a percepção de bidimensionalidade e tridimensionalidade é maior na medida em que se usam técnicas de auralização binaurais ou multicanais. As técnicas de auralização permitem simular com mais eficácia a sensação de imersão em campos sonoros 3D. A gravação binaural parece mais realista, uma vez que o som é gravado de uma maneira mais próxima do sistema acústico humano, envolvendo modelos de propagação sonora formais e simulações para obter o efeito desejado. Existem várias técnicas de gravação e reprodução binaural, como apresentadas em (LOKKI et al., 2002; SUSNIK et al., 2003; MCDERMOTT, 2007). McDermott (2007) descreve uma das técnicas bastante utilizadas, que envolve o conceito de modelagem do ouvinte. Ou seja, um modelo exato da cabeça humana é usado para “ouvir” a uma variedade de sons a partir uma série de locais e orientações diferentes em torno do avatar. O modelo pode ser um “boneco” ou até mesmo uma pessoa comum. Os microfones são então embutidos nas orelhas do modelo para captar os sinais sonoros, representando a forma como o ouvido humano recebe sons. Esses sinais são analisados e o percurso dos sons entre a fonte e cada ouvido são adequadamente modelados por meio de funções de transferência relativas à cabeça (HRTF ou *Head Related Transfer Function*). Através desta técnica, procura-se obter uma série de filtros que podem ser aplicados aos sons, a fim de reproduzir a espacialização percebida.

O uso de som tridimensional pode melhorar as capacidades dos sistemas de RV dramaticamente, contribuindo para a sensação de imersividade e orientação em um ambiente virtual (DOERR et al., 2007). Os sistemas de som 3D duplicam artificialmente os ativadores naturais que auxiliam o cérebro a localizar o som, além de recriarem eletronicamente esses efeitos em tempo real (JACOBSON, 1994). O usuário pode ser capaz de interagir de forma mais eficaz dentro de um ambiente de RV se suas ações forem acompanhadas por sons apropriados que parecem ser emitidos a partir de sua própria localização (BEGAULT, 1994). Assim, os dispositivos auditivos de RV, que nada mais são do que dispositivos geradores de som 3D, auxiliam na aquisição de informações sobre sua localização além do que é fornecido pelo canal visual, tornando a experiência muito mais realista.

Segundo Jacobson (1994), as gravações de som tridimensional baseiam-se em um processo de manipulação auditiva que permite que o artista ou o engenheiro de gravação “posicione” os sons no espaço, controlando sua direção, distância e profundidade. Existem diversas placas de som projetadas para trabalhar com conjuntos de ferramentas que constroem mundos virtuais auditivos. Algumas ferramentas que são usualmente utilizadas em sistemas

e ambientes voltados para aplicações de realidade virtual imersivas, como a Avango, FMOD, VSS, VRJuggler Sound Library, Bergen Sound Server & Library, Wave Engine, AuSIM3D e Lake Huron, são apresentadas em (FARIA, 2005)

2.2.3.5 Dispositivos Hápticos

Em oposição aos dispositivos convencionais como teclado, *mouse* e monitor, os dispositivos hápticos procuram ampliar a interação homem-computador na exploração de mundos virtuais, através de uma interação direta com os nossos sentidos. Para uma melhor interação com um ambiente, deve haver algum tipo de retorno ou resposta (*feedback*) ao pegar ou tocar um objeto virtual, por exemplo. Este tipo de retorno é chamado de retorno háptico (*haptic feedback*) (JEONG; HASHIMOTO; MAKOTO, 2004).

Os dispositivos hápticos incorporam sensores e atuadores que permitem o monitoramento das ações do usuário e podem fornecer-lhe um retorno tanto tátil como de força. O retorno tátil está associado à natureza do contato com o objeto virtual (OV), como a percepção de geometria, temperatura, textura e vibração. Enquanto o retorno de força refere-se ao senso de posição e movimentação, reproduzindo as forças direcionais resultantes de um movimento, o peso ou pressão e a inércia de um OV. Assim, ao tocar ou pegar os objetos do mundo 3D têm-se a sensação de estar manipulando um objeto semelhante ao objeto real, aumentando o senso de imersão no ambiente. A tecnologia de atuadores é bastante variada, sendo que os motores elétricos, pistões hidráulicos e bolsas pneumáticas são os tipos de atuadores mais utilizados e permitem o envio de sensação de força, temperatura e diversas propriedades táteis (BURDEA, 1996).

Os dispositivos hápticos que fornecem sensação de força podem ser divididos em duas categorias básicas: fixos (*ground-based*) ou móveis (*body-based*) (KUNZLER; RUNDE, 2005). Os dispositivos hápticos fixos, como os *joysticks*, são aqueles que estão fisicamente atrelados a uma plataforma ou superfície estável que permite o envio de reações de força ao usuário. Já os dispositivos hápticos móveis utilizam um ponto de conexão do próprio dispositivo para fornecer a reação de força e apresentam a vantagem de poderem ser portáteis na maioria das vezes, como as luvas, roupas e exoesqueletos.

Entre os dispositivos hápticos fixos, encontram-se os dispositivos baseados em braços mecânicos articulados, como pode ser visto na Figura 7. O braço mecânico rastreia a posição da terminação e é capaz de exercer força e vibração na ponta da mesma. Como alternativa à este tipo de sistema, Cai, Ishii e Sato (1996) e Kim, Koike e Sato (2002) apresentam dispositivos a base de tensão, onde ao invés da força ser aplicada através das articulações

de um braço mecânico, cabos são conectados ao ponto de contacto, a fim de exercer uma força sobre o mesmo. Através de motores para criar uma tensão no cabo e codificadores para determinar o comprimento de cada cabo, obtém-se um retorno de força durante a manipulação dos objetos.

Figura 7: Dispositivos hápticos fixos: Falcon da Novint (NOVINT, 2008) e Phantom da SensAble (SENSABLE, 2008) respectivamente.

Com relação aos dispositivos hápticos móveis, encontram-se os dispositivos baseados em um exoesqueleto mecânico que se encaixa no corpo do usuário, fazendo com que determinados movimentos permitam-lhe sentir o peso ou a resistência do material de um objeto no mundo virtual. Alguns sistemas transmitem reação de força apenas para as mãos e braços em formas de luvas, como pode ser visto na Figura 8.

Figura 8: Dispositivos hápticos CyberTouch, CyberGrasp, CyberForce e Haptic Workstation da Immersion Corporation (IMMERSSION, 2008).

2.2.3.6 CAVE

Os sistemas de exibição estereoscópica⁷ ou 3D permitem ao observador/usuário ver os objetos ou o ambiente gerado além das dimensões da tela do computador, acrescentando profundidade à visualização e, conseqüentemente, tornando-a mais realista e próxima da forma que o mundo real é visto. Desde o desenvolvimento dos primeiros capacetes de visualização (HMD's) no fim dos anos 80 até o início dos anos 90, vários dispositivos de exibição de imersão espacial, denominados SID's (*Spatially Immersive Displays*), foram desenvolvidos como uma proposta de interface para sistemas de RV.

⁷A visualização estereoscópica permite a percepção de profundidade e a sensação de espaço.

No início de 1991, os pesquisadores Carolina Cruz-Neira, Thomas A. DeFanti e Daniel J. Sandin, do Laboratório de Visualização Eletrônica (*Electronic Visualization Laboratory*) da Universidade de Illinois em Chicago, projetaram um ambiente de multiprojeção estereoscópico voltado para aplicações imersivas de realidade virtual, baseado em grandes telas de projeção. Pelo projeto, as imagens seriam projetadas em telas dispostas na forma de um cubo, para envolver o observador ou usuário do sistema e prover a sensação de imersão. Este sistema foi anunciado e demonstrado em uma conferência de computação gráfica e técnicas interativas, o SIGGRAPH'92, com o título: "*The CAVE: Audio Visual Experience Automatic Virtual Environment*" (CRUZ-NEIRA et al., 1992).

Cruz-Neira, Sandin e DeFanti (1993) afirmam que o nome "CAVE" é tanto um acrônimo recursivo (*CAVE Automatic Virtual Environment*) quanto uma referência ao "Mito da Caverna", narrado por Platão no livro VII de sua obra "A República", que retrata a exploração da percepção, ilusão e realidade através do uso de metáforas, como a projeção de sombras nas paredes da caverna. Logo, a concepção deste sistema, consiste de uma estrutura em formato cúbico, como uma sala ou uma "caverna digital", onde as paredes e o chão são formadas por telas de projeção. Através de projetores posicionados atrás das telas que formam as paredes e acima do chão deste ambiente, são projetadas imagens estereoscópicas de alta resolução, que quando vistas através de óculos obturadores ou estereoscópicos (*shutter/stereoscopic glasses*), os usuários que encontram-se dentro do volume das telas obtêm uma sensação imediata e intensa de imersão em um ambiente virtual. Além da projeção de imagens 3D, a CAVE também incorpora projeção acústica tridimensional, dispositivos de rastreamento de posição e de interação, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual e interajam com os objetos virtuais projetados neste ambiente.

Os sistemas de visualização do tipo CAVE apresentam ao usuário uma representação estereoscópica em escala natural de um ambiente tridimensional gerado por computador. Logo, o ambiente possui uma relação direta com as dimensões humanas que através de uma visão referenciada pelo *tracking* da cabeça e a interação realística com objetos virtuais, via *data glove* e outros dispositivos, viabiliza a manipulação, operação e controle de mundos virtuais. Ou seja, o sistema oferece uma interface para a navegação no espaço 3D com as capacidades de *look-around*⁸, *walk-around*⁹ e *fly-through*¹⁰ em um mundo virtual.

⁸*Look-around* refere-se a um tipo de interação onde o usuário pode olhar o ambiente virtual em várias direções.

⁹*Walk-through* refere-se a um tipo de interação onde o usuário pode "andar" ou mover-se pelo ambiente virtual restrito a caminhos ou lugares pré-definidos.

¹⁰*Fly-through* é semelhante ao *walk-around*, porém sem a restrição dos movimentos quanto a caminhos ou lugares específicos no mundo virtual.

Apesar do objetivo inicial do desenvolvimento da CAVE ter sido traçado como uma ferramenta útil de visualização científica (CRUZ-NEIRA, 1995), ela acabou sendo utilizada não apenas no meio acadêmico como também na indústria, governos, museus e entretenimento. Por ser uma das mais completas tecnologias de visualização de dados (SHERMAN; CRAIG, 2002), a CAVE tornou-se uma importante ferramenta de pesquisa, cuja vasta gama de aplicações abrange as áreas de engenharia, arquitetura, medicina, arte, negócios e muitas outras. Algumas empresas de produção, por exemplo, utilizam os recursos de um dispositivo CAVE para exibirem seus produtos à clientes potenciais ou então corrigir eventuais problemas relacionados com seus produtos antes que eles sejam submetidos ao processo de manufatura (CREAGH, 2003). A Figura 9, exibe projetistas experimentando e avaliando o interior de um carro, dentro de um sistema CAVE no Centro de Desenvolvimento da Mercedes-Benz¹¹.

Figura 9: Projetistas experimentando e avaliando o interior de um carro dentro de um CAVE.

Uma grande vantagem da CAVE em relação a outros sistemas imersivos é a possibilidade de compartilhamento do mundo virtual por vários usuários simultaneamente. A quantidade de pessoas que podem compartilhar do mesmo ambiente, está limitada diretamente às dimensões físicas da caverna utilizada. A estrutura computacional envolvida no acionamento de CAVE's é bastante avançada e deve processar os pares estereoscópicos das imagens, além de gerenciar os dispositivos de interação, auditivos e projetores. A Figura 10 mostra um esquema que descreve um sistema CAVE com o posicionamento dos projetores. Na figura apresentada, tem-se o uso de espelhos para ajustar a distância dos projetores em relação às telas.

Por ser um sistema de projeção cúbica, estes sistemas podem ser implementados a partir de diferentes configurações, que podem variar de 3 a 6 faces (ou paredes) de projeção. Raskar et al. (1998) propuseram a criação do “escritório do futuro”, onde as paredes do escritório seriam utilizadas para projeção das imagens e o usuário poderia interagir tanto

¹¹<http://www.mercedesbenz.com>

Figura 10: Esquema que descreve a posição dos projetores em um sistema do tipo CAVE.

com seu próprio ambiente de trabalho como com outros ambientes remotos, servindo como estudo para aplicações na telepresença e realidade aumentada. Gross et al. (2003) descrevem um sistema de telepresença chamado *blue-c*, que utilizando um sistema CAVE, um conjunto de câmeras 3D e telas de vidro semitransparentes podem criar a impressão de total imersão para desenvolvimento de produtos e colaboração. Nesta mesma linha, Rhee e Park (2007) adicionam câmeras de vídeo de baixo custo a um sistema CAVE existente, com o auxílio de algoritmos de imagem sofisticados que viabilizem a sensação de presença e colaboração por um custo reduzido. Modelos mais modestos como o de Jacobson (2002), por exemplo, são formados por apenas 3 faces, onde as imagens são projetadas em 2 paredes posicionadas em forma de V, além de uma projeção direta no chão. Este tipo de caverna é mais simples e barata, porém a sensação de imersão é menor (do que um sistema com 6 faces) e oferece menos mobilidade ao usuário, pois o mesmo fica limitado a um ponto central do ambiente.

A construção de um sistema de multiprojeção imersivo demanda uma série de conhecimentos interdisciplinares, como a implementação e controle do sistema de projeção, desenvolvimento de softwares de rede, controle de múltiplos equipamentos, renderização das imagens em tempo real, *softwares* de vídeo e áudio, projeção sincronizada, arquitetura e design, etc. Além disso, por se tratar de uma tecnologia recente e de alto custo, muitos trabalhos envolvendo esta tecnologia ainda estão em fase exploração e desenvolvimento.

O *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos* - um projeto financiado pela FAPESP, sob coordenação do Professor Arthur Jose Vieira Porto - disponibiliza um CAVE recém instalado, onde através deste projeto, os pesquisadores da região de São Carlos têm uma infraestrutura de visualização 3D imersiva disponível para o desenvolvimento de pesquisas, principalmente no contexto de desenvolvimento da Manufatura Virtual.

Dentre as várias possibilidades de sistemas de multiprojeção imersivos, o Laboratório de Simulação da Escola de Engenharia de São Carlos optou por montar uma CAVE de 3 faces, onde 2 delas correspondem à 2 paredes laterais e a terceira corresponde ao chão. As paredes são formadas por telas de projeção posicionadas em forma de V, com um ângulo de 90 graus entre elas. Para cada face existe um par de projetores responsáveis pela projeção das imagens estereoscópicas. A projeção das imagens nas paredes laterais é realizada a pela parte de trás das telas, enquanto o chão recebe uma projeção direta pela parte superior. Assim, o sistema é composto por 6 projetores de alta intensidade luminosa e alta resolução, cujas imagens projetadas nas telas são geradas por 3 computadores com placas gráficas de alto poder de processamento, um para cada 2 projetores. A estrutura do cubo é toda feita em madeira e plástico, envolvendo o uso de metal apenas em alguns parafusos estruturais e projetores, buscando minimizar a distorção dos campos magnéticos gerados por algum dispositivo eletromagnético que venha ser usado, como rastreadores de posição.

2.2.4 Dispositivos de Telepresença

Os sistemas de telepresença referem-se a um conjunto de tecnologias que permitem a sensação de presença em um local remoto, a partir de uma localização diferente da que pretende-se interagir (STEUER, 1992; FISHER, 1991). Muito similar à realidade virtual, onde luta-se para alcançar a ilusão de presença dentro de uma simulação gerada por computador, a telepresença luta para alcançar a ilusão de presença em um local remoto. Logo a idéia da telepresença, lembra a RV, exceto que na RV o usuário está imerso em um mundo virtual, ao passo que na telepresença o usuário está imerso em um mundo real remoto. Assim, Rosenberg (1995) afirma que no fim, tanto a telepresença como a realidade virtual são essencialmente o mesmo, uma interface humano-computador que permite ao usuário tirar vantagem de suas habilidades naturais quando interage com um outro ambiente que não aquele que o rodeia.

Como a telepresença é a habilidade de perceber e agir em um ambiente remoto e não apenas percebê-lo, o usuário precisa receber estímulos a partir do local remoto para que o mesmo possa ter a habilidade de exercer o controle ou a manipulação de objetos dentro deste ambiente distante (CHAPIN; LACEY; LEIFER, 1994). Dessa forma, recursos como vídeo, áudio, dispositivos hápticos, sistemas de visualização, como HMDs (*Head Mounted Displays*), telas de exibição auto-estereoscópicas¹², auto-falantes estéreo, luvas e outros dispositivos de RV, podem ser utilizados pelo operador do sistema remoto.

¹²Dispositivos de exibição autoestereoscópicos permitem a visualização e percepção de imagens 3D sem a necessidade de óculos especiais.

As tecnologias dos sistemas de telepresença, do ponto de vista do usuário, requerem que o controle seja exercido pelo operador humano e dispositivos tais como sistemas de rastreamento (*tracking*) de cabeça e corpo, *joysticks*, luvas e estruturas em forma de exoesqueleto, entre outros controladores de aplicação específica. Quanto a tecnologia utilizada no ambiente remoto, grande parte desta é semelhante ao encontrado na área da robótica, mais precisamente a telerobótica, uma vez que os sistemas de telepresença não exigem operação autônoma. Os sistemas de telepresença por definição requerem um laço humano na operação ou interação com o ambiente. Dessa forma, os equipamentos que podem ser encontrados no ambiente remoto incluem sensores, microfones, câmeras, manipuladores entre outros.

2.2.5 Bibliotecas de Desenvolvimento de Ambientes Virtuais

Com a avanço no estudo de tecnologias de realidade virtual, programas e bibliotecas vêm sendo desenvolvidos para os mais diversos tipos de sistemas de RV. Algumas interfaces de programação de aplicação (API - *Application Programmer's Interface*), como CAVELib e VR Juggler são específicas para dispositivos de exibição imersivos, enquanto outras como OpenGL e Direct3D são de propósito geral para programação 3D.

A OpenGL (*Open Graphics Library*) é uma biblioteca gráfica aberta composta por funções e estruturas de dados que constitui uma interface portátil de alto nível para acesso ao *hardware* gráfico. Essa interface consiste basicamente de funções que podem ser utilizadas para especificar objetos e operações necessárias para produzir aplicações 3D interativas, oferecendo ao desenvolvedor um pequeno conjunto de primitivas geométricas. A biblioteca Direct3D, faz parte da tecnologia DirectX da Microsoft, e fornece um conjunto de funcionalidades de grande utilidade para o desenvolvimento de aplicações 3D, sendo considerada padrão na plataforma Windows. Ela apresenta uma interface de alto nível, permitindo acesso aos recursos avançados de hardware das placas gráficas 3D, sem a necessidade de conhecer as particularidades de cada *hardware*.

Como a OpenGL e a Direct3D, são bibliotecas de baixo nível, que se concentram basicamente na renderização das cenas, foram desenvolvidas bibliotecas baseadas nesta tecnologia que se concentram na estruturação das cenas 3D, como a API Java 3D. O Java 3D consiste em uma API baseada nas bibliotecas gráficas OpenGL e DirectX, e refere-se a uma hierarquia de classes Java que serve como interface para o desenvolvimento de sistemas gráficos tridimensionais, onde o programador necessita trabalhar apenas com construtores de alto nível para criar, manipular e visualizar objetos geométricos, tendo os detalhes de visualização gerenciados automaticamente (BROWN; PETERSEN, 1999). Ainda com base na

OpenGL e a Direct3D, foram desenvolvidas as chamadas *engines*, um pacote de software orientado à cena, que permite a construção de aplicações 3D de um modo mais fácil e flexível. As *engines* abstraem os detalhes e funcionalidades de bibliotecas de baixo nível como a OpenGL e a Direct3D e oferece uma interface baseada em objetos e classes mais intuitivas.

Um grande exemplo de uso das *engines*, está no desenvolvimento de jogos de computadores. Através de um *game engine* (motor de jogo ou motor 3D), têm-se um pacote de *software* que lida com as principais funcionalidades do jogo. Ou seja, um código que define e processa todas as possibilidades de um determinado jogo, tais como: gráficos, sons, inteligência artificial, rede, controle entre outras. Algumas das particularidades deste conceito são a sua modularidade, extensibilidade e facilidade de reutilização. Além disso, os melhores motores de jogos permitem ainda que os usuários possam modificar cenários, modelos, sons e outras componentes do jogo (os chamados *mods*), tornando as possibilidades ainda maiores. Dessa forma, alguns autores e pesquisadores de aplicações de realidade virtual, como Jacobson e Lewis (2005), Tan, Hashimoto e Sato (2005), Ryu et al. (2006), utilizam ferramentas baseadas em motores 3D de jogos para implementação de mundos virtuais em um dispositivo de exibição imersivo.

Com relação as linguagens de RV destacam-se a VRML e o X3D. A VRML (*Virtual Reality Modeling Language*), como o próprio nome sugere é uma linguagem para modelagem de realidade virtual, que consiste em um formato de arquivo onde é possível criar objetos tridimensionais através da descrição ou definição de cor, transparência, brilho e textura. Os objetos podem ser formas básicas, como esferas, cubos, ovóides, hexaedros, cones, cilindros, ou formas criadas pelo próprio programador, como as extrusões (CAREY, 1998). Tido como o sucessor da VRML, o X3D (*Extensible 3D*) tem como sua principal característica a capacidade de codificar a cena usando a sintaxe da XML (*eXtensible Markup Language*). O X3D é um padrão aberto para distribuir conteúdo 3D. Ele não é uma API de programação, tão pouco um formato de arquivo para troca de geometrias. Ele combina ambos, geometria e descrições de comportamentos instantâneos em um simples arquivo que tem inúmeros formatos de arquivos disponíveis para isso, incluindo o XML (DALY; BRUTZMAN, 2007). Alguns trabalhos envolvendo o desenvolvimento de ambientes virtuais utilizando X3D, podem ser encontrados em (HAIQING; GUOFU; JIE, 2006; BOURAS et al., 2007)

Inicialmente, muitas aplicações 3D para RV eram programadas diretamente a partir da biblioteca gráfica OpenGL ou alguma outra biblioteca similar. Posteriormente, foram desenvolvidos *kits* ou sistemas de autoria de ambientes virtuais ou ainda bibliotecas específicas

pra RV, como o ARToolkit. O ARToolkit¹³ é uma biblioteca desenvolvida no Laboratório de Tecnologias de Interface Humana da Universidade de Washington, que permite a criação de aplicações de realidade aumentada. Para isso, ela utiliza as capacidades de rastreamento de vídeo a fim de calcular a posição e orientação de uma câmera com relação a marcadores físicos em tempo real.

O ambiente de desenvolvimento de programas aplicativos para sistemas de multiprojeção estereoscópicas é complexo, pois leva em conta aspectos como a representação de grandes estruturas de dados, o gerenciamento de periféricos e o mapeamento das imagens nos projetores. Neste contexto, encontra-se a CAVELib¹⁴, que é a API original desenvolvida pelo Laboratório de Visualização da Universidade de Illinois em Chicago, criadores da CAVE. Esta API provê o suporte geral para a construção de ambientes virtuais para sistemas de multiexibição imersivos e equipamentos de realidade virtual. A CAVELib permite a configuração dos dispositivos de exibição, a sincronização dos múltiplos processos, mapeamento das vistas estereoscópicas, comunicação com os dispositivos de rastreamento, suporte ao modelo de câmera virtual baseado no ponto de vista do observador e os recursos básicos de comunicação em rede entre ambientes de realidade virtual remotos.

Em alternativa à CAVELib, foram desenvolvidas outras bibliotecas para os sistemas de exibição imersivos, em cujo cenário destaca-se a VR Juggler (BIERBAUM et al., 2001). O VR Juggler é um projeto de pesquisa do Centro de Aplicações para RV da Iowa State University, e trata-se de uma biblioteca de realidade virtual, de código aberto, multiplataforma e independente de dispositivo. Ela fornece uma abstração dos dispositivos de entrada e saída, podendo ser utilizada em conjunto com qualquer biblioteca de grafo de cena (OpenGL Performer, OpenSceneGraph, OpenSG) para renderizar a cena.

Com o uso do VR Juggler, o desenvolvedor não precisa se preocupar com o dispositivo de entrada que o usuário irá utilizar para interagir com a aplicação, ou como ele irá visualizá-la. A escolha dos dispositivos é feita através de arquivos de configuração. Baseado nesses arquivos, a biblioteca carrega os *drivers* adequados. A grande vantagem dos arquivos de configuração é permitir que diferentes dispositivos sejam combinados de várias maneiras sem necessidade de recompilar a aplicação. Assim, a aplicação que usa o VRJuggler não precisa ser desenvolvida levando em consideração o tipo ou o número de projetores ou telas usadas para a exibição, nem da disposição delas, nem precisa ser previamente preparada para ser controlada por um determinado dispositivo. O usuário pode configurar a aplicação para rodar com um *joystick* em uma CAVE ou utilizar um *mouse* 3-D imerso em um capacete de realidade virtual, apenas

¹³<http://www.hitl.washington.edu/artoolkit/>

¹⁴<http://www.mechdyne.com>

mudando alguns parâmetros do arquivo. Além de sistemas do tipo CAVE, o VR Juggler suporta uma variedade de dispositivos de RV, como mesas de projeção, monitores simples de estações de trabalho e os tradicionais HMDs.

2.2.6 Aplicações de Ambientes Virtuais e Telepresença

Um ambiente virtual pode representar qualquer mundo tridimensional real ou abstrato. Isso inclui sistemas reais como edifícios, paisagens, submarinos, naves espaciais, locais de escavação arqueológica, anatomia humana, esculturas, reconstrução da cena de um crime, sistemas solares, e assim por diante. Estes mundos virtuais podem ser animados, interativos, compartilhados e podem expor comportamento e funcionalidade.

Muito similar aos ambientes virtuais, onde luta-se para alcançar a ilusão de presença dentro de uma simulação gerada por computador, a telepresença luta para alcançar a ilusão de presença em um local remoto com a habilidade de perceber e agir em um ambiente distante através da manipulação e controle dos objetos reais remotos (CHAPIN; LACEY; LEIFER, 1994). Assim, a combinação de realidade virtual e telepresença (teleoperação e telerobótica) permitem a criação e utilização de um vasto número de aplicativos que podem ser utilizados nos mais diversos campos do pesquisa, trabalho, exploração, inspeção e manutenção.

Portanto, tal como as tecnologias de realidade virtual e telepresença evoluem, as aplicações de RV e telepresença tornam-se literalmente ilimitadas. Adams (1994), acredita que a RV é uma reformulação da interface entre pessoas e tecnologia da informação, oferecendo novas formas de comunicação de informações, a visualização de processos, bem como a expressão criativa das idéias. Assim, as aplicações da RV podem incluir a formação de uma variedade de áreas: militar, médica, equipamentos de operação, educação, concepção de avaliação (prototipagem virtual), estudos ergonômicos e arquitetônicos, simulação de seqüências de montagem e manutenção tarefas, estudo e tratamento de fobias (por exemplo, medo de altura), entretenimento, e muito mais.

Publicações de trabalhos que empregam os sistemas de realidade virtual e telepresença têm sido descritos para os mais diversos fins, como a simulação de missões críticas na medicina (SORID; MOORE, 2000; PARK et al., 2007); modelagem e testes de colisão na indústria automobilística (FELEZ; VERA; MARTINEZ, 1999; SCHWEBELA; GAINESA; SEVERSON, 2008); estudos de biologia molecular (GILLET et al., 2004); educação e entretenimento (HUGHES et al., 2005); tratamento de fobias (JUAN et al., 2005; TAKACS; SIMON, 2007); simulação do comportamento de satélites no espaço (KIMURA; IBARAKI, 1999) e simulação, inspeção e treinamento com aeronaves (VORAA et al., 2002).

Buscando novas formas de obter uma melhoria organizacional das empresas e propiciar a modernização do sistema produtivo, alguns pesquisadores desenvolveram aplicações baseadas em tecnologia de RV e telepresença para treinamento de funcionários (PARK; JANG; CHAI, 2006); controle e planejamento da produção (SCHENK; STRASSBURGER; KISSNER, 2005); visualização de *layouts* de fábricas e prototipação de produtos (HUANG et al., 2006)); processos de engenharia (CORSEUIL et al., 2003) e controle de estoques (MONTEIRO; PAIVA, 2002). O potencial das ferramentas de desenvolvimento de ambientes de RV tem permitido modelar maquinários, veículos e dispositivos, e simular o comportamento real destes equipamentos, diminuindo os custos envolvidos em simulações reais. A seção seguinte aborda os ambientes virtuais e a telepresença no contexto da indústria de manufatura.

2.2.6.1 Manufatura Virtual

A aplicação de ambientes virtuais (AV) no processo de manufatura está relacionada tanto com a manutenção e treinamento, como também a criação e simulação de protótipos virtuais antes de sua utilização no chão-de-fábrica. Wang, Shen e Lang (2004) listam trabalhos relacionados com manufatura virtual envolvendo a área automotiva, aeroespacial, indústria pesada e a indústria de ferramentas e máquinas, onde sistemas são projetados entre outras funcionalidades, para monitoramento e controle de chão-de-fábrica, operação de máquinas CNC (*Computer Numerical Control*) remotas ou automação de fábrica avançada.

Prototipação virtual é um passo importante em direção ao desenvolvimento eficiente do produto final. Baseados nas informações de geometria e topologia do projeto, nos resultados da simulação obtidos por ferramentas de modelagem combinados com os cálculos de cinemática, o material, a tolerância e outras informações disponíveis sobre o produto, será possível gerar protótipos no computador para apresentações realistas, diminuindo os custos com protótipos reais e com o tempo de disponibilização para testes, permitindo ainda interações com o produto até mesmo nos estágios iniciais de desenvolvimento.

Dentre os benefícios oferecidos pela virtualização de equipamentos de manufatura, Shridhar e Ravi (2002) citam a verificação e modificação rápida dos projetos, maior confiabilidade às decisões, avaliação e testes de propriedades estruturais e funcionais de máquinas e produtos projetados, desenvolvimento de um produto ergonomicamente funcional e confiável sem a necessidade de construir um modelo em escala real, projeção da estética do produto de acordo com as preferências do consumidor, além do desenvolvimento e avaliação de processos assegurando a manufaturabilidade e a montabilidade, sem produzir de fato o produto em escala comercial. Porto et al. (2002) acrescentam a realização de treinamento

por intermédio de um ambiente virtual e distribuído para os operadores, técnicos e gerência no uso das instalações de manufatura, onde os custos de treinamento e produção podem, assim, serem reduzidos.

A criação de protótipos virtuais pode proporcionar diversas vantagens aos processos industriais, como a redução do *time-to-market*, ou seja, do tempo decorrente entre o início do processo de desenvolvimento de um produto à sua introdução no mercado, e a redução dos custos de desenvolvimento, pois protótipos virtuais permitem reduzir o número de protótipos físicos necessários, conseqüentemente reduzindo a quantidade de ferramentas e materiais utilizados para a sua construção. Assim, Porto et al. (2002) resumem as principais vantagens da prototipação virtual, para os processos industriais, em três pontos:

- **Redução de Tempo:** o parâmetro tempo nos dias de hoje é um dos mais importantes fatores para a indústria. O *time-to-market* é a chave do marketing que diferencia os competidores.
- **Diminuição de Custos:** os protótipos virtuais podem reduzir a necessidade de se fazer um grande número de protótipos físicos, isto possibilita uma diminuição no tempo de desenvolvimento e do trabalho humano empregado no projeto. Também há diminuição da quantidade de ferramentas e materiais utilizados para a confecção do protótipo físico. Os resultados do protótipo virtual são obtidos mais rapidamente e por isto possibilitam um feedback para o projeto, antes dos custos da produção estarem fixados.
- **Melhoria da Qualidade:** a aplicação de diferentes alternativas para um projeto pode ser realizada mais rapidamente, permitindo uma melhoria da validação das soluções apropriadas que satisfaça os parâmetros especificados pelo cliente, com um menor custo.

2.3 Redes de Petri e Ambientes Virtuais

A representação gráfica das Redes de Petri (RP), além de ser de fácil compreensão, permite o encapsulamento de detalhes, oferecendo um método adequado para a modelagem dos diferentes níveis de coordenação de ambientes virtuais. Além disso, as RPs oferecem um forte suporte matemático para a análise do comportamento dos ambientes virtuais e técnicas de simulação complementares. Com o modelo baseado em RPs, é possível prever e testar o comportamento de um ambiente antes mesmo de sua implementação.

A necessidade crescente da utilização de recursos de modelagem para projetar ambientes virtuais gerou alguns ambientes baseados em Redes de Petri. Mascarenhas et al. (1998) utilizaram RPs para a modelagem e análise de um sistema de RV de um CAVE (sistema de baseado na projeção nas paredes, chão e teto de imagens de um ambiente sintético geradas pelo computador). Enquanto, Ying e Gracanin (2004) propõem a modelagem de um mundo virtual através do formalismo de RP, com o intuito de apoiar a subjetividade e proporcionar a colaboração e interações consistentes entre os usuários.

Ishii et al. (1997) apresentam o desenvolvimento de um sistema de treinamento baseado em realidade virtual para ensino de procedimentos de desmontagens de máquinas mecânicas utilizadas em instalações de usinas nucleares. Neste sistema foi empregado a as RP como método de modelagem para descrição das ações dos funcionários envolvidos no treinamento.

Já Horvath e Rudas (1999) utilizam as RP em uma pesquisa que integra o processo de modelagem de sistemas de manufatura com a modelagem de manufatura virtual. Enquanto Smith e Duke (1999) fazem uma breve descrição de pesquisas baseadas em técnicas de modelagem de sistemas híbridos, com enfoque em técnicas de interações em Ambientes Virtuais utilizando Redes de Petri de alto nível.

(WILLIAMS; MICHAEL; SMITH, 2001) descrevem o comportamento de um Ambiente Virtual como um sistema híbrido. Ele descreve o comportamento dos objetos desse ambiente através do formalismo FlowNet. Este formalismo utiliza Redes de Petri para descrever o processo discreto e elementos de notação para modelagem de sistemas dinâmicos para o processo contínuo. Uma das vantagens desse formalismo é permitir modelar o comportamento do sistema sem considerar transformações de baixo nível.

Raposo (2000) descreve a coordenação em ambientes colaborativos usando RPs, enquanto Zhou, Murata e DeFanti (1999) desenvolveram um trabalho para a modelagem de ambientes virtuais em rede, onde utilizaram uma extensão de Redes de Petri a RP *fuzzy*-temporizada para a modelagem e simulação de ambientes de realidade virtuais em rede. Ainda com o uso de RP *fuzzy*-temporizada, Yu, Wu e Wu (2004) apresentam um modelo que integra RP *fuzzy*-temporizadas extendidas com a tecnologia de agentes para ambientes virtuais colaborativos.

3 Sistema de Autoria de Telepresença com Controle em Redes de Petri

Esta seção concentra-se na descrição de um trabalho iniciado por Palma (2001) cujo tema é uma proposta de um metamodelo para modelagem e simulação de sistema a eventos discretos, baseado em redes de petri e realidade virtual. A partir de sua proposta inicial, Palma continuou desenvolvendo trabalhos de pesquisa na Universidade Estadual de Londrina em um projeto para geração de uma “Ferramenta para o Desenvolvimento de Sistema de Telepresença Colaborativa com controle em Redes de Petri”. Com o desenvolvimento deste projeto, foi possível alcançar o foco desta seção: um Sistema de Autoria de Telepresença com Controle em Redes de Petri.

Palma (2001) propõe um metamodelo para a construção de ferramentas que viabilizam a criação de um sistema de autoria para Ambientes Virtuais (AV) sem a necessidade de programação. A idéia central deste metamodelo é disponibilizar ao projetista de simulação um ambiente de desenvolvimento que encapsule a complexidade de programação de sistemas de realidade virtual e telepresença, fornecendo elementos que permitam a construção de Ambientes Virtuais de “forma intuitiva”.

O trabalho utiliza os conceitos de telepresença como o controle de sistemas em tempo real, ambientes virtuais, ambientes reais e teleoperação, a fim de prover uma ferramenta para o desenvolvimento de sistemas de telepresença, que permite a modelagem do controle de ambientes virtuais e reais, através de um modelo gráfico/matemático com execução do controle centralizado.

A ferramenta resultante da proposta inicial é um sistema de autoria que contém uma coleção de aplicativos a fim de editar o modelo de controle, configurar o ambiente de telepresença (virtual e real), executar o controle do ambiente (servidor) e gerar os ambientes (abstração do mundo em 3D ou o mundo real) através de instâncias em vários clientes com

objetos pré-definidos (OLIVEIRA et al., 2007a, 2007b).

Além disso, é baseada em linguagem de especificação formal, Redes de Petri (RP), visto que, a implementação e o teste de ambientes virtuais são complicados devido às concorrências e as características de tempo real. Portanto, o desenvolvimento de métodos formais para a modelagem e análise é altamente desejado, uma vez que eles devem modelar a sincronização dos processos assíncronos, que é muito comum em sistemas de realidade virtual (RV).

A RP é uma ferramenta bem apropriada para a modelagem de ambientes virtuais, pois pode capturar naturalmente as principais características de sistemas de RV, e proporcionar resultados empíricos muito bons na verificação automática de concorrência e sistemas de tempo real, por permitir modelar facilmente a computação não determinística e paralela (MASCARENHAS et al., 1998).

A ferramenta desenvolvida oferece um sistema virtual que tem associado ou não o sistema real. Assim, ela pode ser utilizada em várias aplicações de realidade virtual como a telemedicina e teleoperação de equipamentos presentes em ambientes hostis ao ser humano; para treinamento dos operadores em equipamentos de alto custo ou com alta periculosidade ao operador; para demonstração de produtos (*marketing*); além de possibilitar a geração de ambientes com uma coleção de máquinas que atuam de forma integrada, como por exemplo um sistema de manufatura.

Nas seções seguintes, serão apresentados, em linhas gerais, a estrutura do sistema de autoria de telepresença bem como a descrição de cada um dos componentes que formam a ferramenta.

3.1 Estrutura do Sistema

A proposta inicial de Palma (2001) gerou uma ferramenta-linguagem de 4º geração (4GL), portadora de um nível de abstração mais elevado, para fornecer suporte no desenvolvimento de ambientes de telepresença. Dessa forma, a ferramenta caracteriza-se como um sistema de autoria de telepresença, por permitir que o usuário especifique um ambiente em um nível elevado integrado com um ambiente em um nível inferior (OLIVEIRA et al., 2007a, 2007b). Ou seja, a ferramenta auxilia a criação de Ambientes Virtuais (AV) integrados com Ambientes Reais (AR).

Através deste sistema de autoria, pode-se desenvolver a modelagem do controle de

ambientes virtuais e reais por meio de um modelo gráfico/matemático chamado Redes de Petri. Além disso, a composição do ambiente de telepresença pode ser realizada por meio de objetos pré-definidos numa Biblioteca de Objetos Virtuais e Objetos Reais. A integração do modelo de controle com os objetos virtuais/reais, a execução do controle centralizado e a execução em cada cliente do ambiente virtual/real, completam as características do sistema aqui apresentado.

Uma vez estabelecidas as características do sistema, a condução dos estudos realizados para o desenvolvimento do trabalho centraram em: conceitos de telepresença e teleoperação, ambientes reais e virtuais, comunicação, controle e Redes de Petri. Cujas descrição geral do sistema pode ser resumida como na forma apresenta na Figura 11.

Figura 11: Estrutura do Modelo de Sistema de Autoria de Telepresença.

Um sistema baseado no metamodelo descrito acima fornece ao usuário final um ambiente que:

- não utiliza linguagem de programação em nenhuma fase do processo de simulação;
- suporta a edição e verificação do modelo de controle do sistema a ser simulado;
- permite a definição do Ambiente Virtual e Real;
- efetua a ligação lógica entre o Ambiente Virtual e o modelo de simulação; e
- executa a simulação em um Ambiente Virtual comandado pelo modelo, que permite a interação do usuário.

3.2 Componentes do Sistema

O sistema de autoria consiste em uma ferramenta que contém uma coleção de aplicativos a fim de editar o modelo de controle, configurar o ambiente de telepresença (virtual e real), executar o controle do ambiente (servidor) e gerar os ambientes (abstração do mundo em 3D ou o mundo real) através de instâncias em vários clientes com objetos pré-definidos. Esta coleção de aplicativos contém um repositório comum de arquivos para fornecer dados de entrada e armazenar dados de saída necessários para a integração, onde, o arquivo de saída de um aplicativo pode ser a entrada de outro. Sendo assim, esta seção objetiva descrever individualmente os módulos que compõem o sistema, introduzidos por meio da Figura 11, exibida na seção anterior.

3.2.1 Biblioteca de Classes de Objetos

A Biblioteca de Classes de Objetos Virtuais e Reais (BCOVR) é um repositório dos códigos das abstrações de objetos aptos a comporem uma cena virtual e real. A Biblioteca deve ser previamente concebida dentro de um determinado contexto antes de fornecer ao usuário final a ferramenta de autoria. As classes contidas na BCOVR foram criadas em um compilador, o código fonte é desenvolvido em linguagem de Programação Orientada a Objetos e as classes foram baseadas na abstração dos objetos reais.

Para cada entidade a biblioteca tem duas abstrações, uma que faz a abstração tridimensional do objeto e a outra refere-se aos comandos de *hardware* do objeto real. A Figura 12 exibe um diagrama da estrutura das classes de objetos virtuais e reais.

Figura 12: Diagrama da estrutura das Classes de Objetos Virtuais e Reais.

Para o desenvolvimento da biblioteca empregou-se os conceitos de abstração, hierarquia e polimorfismo da orientação a objetos, de forma que a classe pai de uma entidade é abstrata, e as classes filhas redefinem os métodos da classe pai (métodos virtuais) dentro de

dois escopos: real e virtual. Por meio destas propriedades foi possível trabalhar em um nível elevado de abstração no Módulo Central de Controle (MCC), pois este não precisa distinguir se o cliente é um ambiente que instanciou objetos reais ou virtuais da entidade e ainda permeia a sincronização dos ambientes reais com os virtuais.

As classes “_real” servem para atuar diretamente nos objetos reais, cujos métodos destas transmitem comandos ou recebem dados dos protótipos reais através das portas (seriais, paralelas ou outras). As classes de objetos reais estão relacionadas aos protótipos motorizados que compõem um ambiente real (físico) para simulação.

Quanto as classes “_virtual”, estas referem-se aos objetos virtuais que são modelados em 3D. Os métodos são manipulações de objetos gráficos, ou seja uma composição de transformações geométricas que correspondem ao comportamento do objeto real e as instâncias das classes são os objetos que compõem a cena do ambiente virtual.

Com este tipo de organização de classes de objetos, a adição de protótipos ao sistema é obtida com a elaboração de uma nova classe abstrata e as suas derivações: virtual e real do objeto em questão. Dessa forma a atualização do sistema é facilitada e há uma ampliação das possibilidades de simulação com o acréscimo de novos objetos à biblioteca. Caso a biblioteca tenha uma boa coleção de classes de forma a cobrir todas as entidades de um determinado escopo, então habilitará a criação do ambiente desejado, caso contrário, é responsabilidade do fornecedor do sistema desenvolver novas classes, pois exigirá programação e conhecimentos específicos de RV.

3.2.2 Configurador de Ambientes Virtuais

O Configurador de Ambientes Virtuais (CAV) tem como objetivo simplificar o trabalho, fornecendo uma interface para a criação de Ambientes Virtuais, como pode ser visto na Figura 13. O CAV não provê funções para representar a lógica e nem a dinâmica de execução do AV, seu objetivo é permitir criar a cena do Ambiente Virtual de tal forma que o usuário poderá definir a disposição dos objetos no ambiente e o estado inicial dos seus atributos. As informações são empregadas pelo Editor de Redes de Petri, pelo Simulador de Ambientes Virtuais (SAV) e pelo Sistema de Ambiente Real (SAR).

O objetivo do CAV é apresentar ao usuário uma interface gráfica simples para a montagem de Ambientes Virtuais, configurando os objetos virtuais diretamente. Esta ferramenta utiliza uma representação icônica, cujos objetos e atributos podem ser manipulados diretamente por uma representação visual através de caixas de diálogo. Ou seja, os objetos

Figura 13: Configurador do Ambiente Virtual.

estão em forma de ícones para criar ou representar as instâncias das classes de objetos virtuais da Biblioteca de Classes de Objetos. A coleção de classes da biblioteca pode variar na quantidade de classes ou no escopo dos ambientes que se deseja configurar, enquanto o CAV adapta-se automaticamente às mudanças da biblioteca.

3.2.3 Editor de Redes de Petri

O Editor de Redes de Petri tem como objetivo simplificar o trabalho de modelagem do sistema, sendo responsável em representar a lógica de execução do AV e fornecer uma interface para a criação de um modelo em Redes de Petri, como pode ser visto na Figura 14. O editor permite a manipulação de objetos gráficos na janela de edição (documento) e gera um arquivo de saída representando o modelo de controle do ambiente em Redes de Petri.

Através das informações fornecidas pelo Configurador de Ambientes Virtuais (CAV) à respeito dos atributos e métodos que compõem as classes dos objetos virtuais e reais, o analista ou usuário do sistema de telepresença pode representar os objetos das classes através de um modelo de RP.

O Editor gera um modelo que será empregado pelo Módulo Central de Controle (MCC), que por sua vez executará a dinâmica do AV e do AR regida pelo modelo de Redes de Petri. Ao editar um modelo de Redes de Petri, é realizada a relação do objeto instanciado no CAV com os elementos desta rede.

Figura 14: Edição do elemento lugar (*place*) no Editor de Redes de Petri.

3.2.4 Ferramenta de Autoria

Através de uma ferramenta de autoria, o próprio projetista do sistema pode criar o ambiente virtual sem o auxílio de um técnico em informática. É claro que, muitas vezes, estes programas tornam-se muito simples, devido aos limitados recursos oferecidos pelas ferramentas, o que não influencia no conteúdo do *software*. O projetista dos sistemas pode buscar apoio de técnicos em informática, para a criação de novos objetos para a Biblioteca relativos aos novos equipamentos (protótipos) que venham fazer parte do ambiente.

Os arquivos gerados na ferramenta de autoria alimentam o sistema de telepresença onde:

- a cena do Ambiente é editada no CAV tal como a disposição do ambiente real (organização dos protótipos) e o arquivo gerado pelo CAV serve para vários computadores/clientes instanciarem o Ambiente (um Simulador de Ambiente Virtual (SAV) ou Sistema de Ambiente Real (SAR));
- a lógica de controle do ambiente é editada no Editor de Redes de Petri, e o arquivo gerado alimenta o Módulo Central de Controle, onde os objetos são controlados em um computador caracterizado como servidor.

3.2.5 Módulo Central de Controle

O Módulo Central de Controle (MCC) é um sistema que é executado como um servidor e controla todas as instâncias de ambientes reais ou virtuais (clientes), como se fossem um único ambiente compartilhado por vários usuários. Na memória do servidor fica a Rede de Petri modelada pelo usuário no Editor de Redes de Petri e que é responsável pelo

controle do Ambiente. O ambiente é único, mas pode ser instanciados em várias máquinas e pode ser virtual ou real, de forma que todas as ações executadas em uma máquina sejam refletidas e executadas em outras, pois o MCC é o servidor e o único que comanda todas as instâncias do ambiente (um para cada cliente conectado) como um único ambiente. Como o MCC trabalha em um nível mais abstrato não precisa diferenciar o que é real ou virtual. Assim, as interferências ocorridas nos objetos reais são refletidas nos objetos virtuais e vice-versa.

Para ser executado o MCC precisa do modelo, dos dados da interligação do modelo com o AV para se comunicar com o AV e da configuração do Ambiente Virtual que vai ser simulado.

3.2.6 Sistema de Ambientes Reais

O Sistema de Ambientes Reais (SAR) é composto por protótipos reais que são comandados pela serial ou paralela enviado pelo computador na qual estão conectados. No computador que está executando o Sistema de Ambiente Real (SAR), as instâncias dos objetos reais são controladas pelo Módulo Central de Controle (MCC).

A configuração do AR é feita manualmente pelo usuário do sistema. O usuário define um espaço físico que representará o seu mini-mundo e dispõe os objetos reais de acordo com o que se deseja que seja simulado no AV.

3.2.7 Sistema de Ambientes Virtuais

O Sistema de Ambientes Virtuais (SAV) é composto por objetos virtuais 3D que são réplicas dos objetos reais que irão compor o ambiente real. Através do Configurador de Ambiente Virtual (CAV) é possível editar a cena com os objetos virtuais que irão compor o ambiente virtual. E, através do Editor de Redes de Petri será estabelecida a lógica de controle deste ambiente editado no CAV.

Durante a execução do sistema, apenas os comandos relativos à navegação no ambiente não depende do controle ou autorização do MCC. Dessa forma, cada usuário pode ter um ponto de vista diferenciado do ambiente virtual em execução. Exceto os comandos de navegação, qualquer outra solicitação efetuada tanto no ambiente virtual como no ambiente real, será encaminhada ao MCC que conduz o sistema em execução.

3.2.8 Sistema de Comunicação

A fim de realizar a comunicação entre o Simulador de Ambientes Reais (SAR), o Simulador de Ambientes Virtuais (SAV) e o Módulo Central de Controle (MCC) foi criada uma biblioteca, denominada Netix, feita em cima do protocolo UDP. Algumas políticas de averiguações de chegada e descarte de pacotes foram implementadas, e as instâncias dos objetos de comunicação são amplamente utilizadas no MCC e no de Sistema de Ambientes Virtuais e Reais.

3.2.9 Protótipos Reais e Virtuais

Para composição do ambiente real, foram desenvolvidos protótipos com sucata criando-se os seguintes equipamentos: um guindaste, um botão e uma esteira, como pode ser visto na Figura 15.

Figura 15: Protótipos reais: guindaste, botão e esteira respectivamente.

O computador se comunica com os protótipos utilizando a porta paralela ou serial, tanto para o recebimento quanto para o envio de dados, pois o controle dos dispositivos é realizado pelo computador. Os comandos são emitidos aos protótipos e estes efetivam as ações solicitadas e também envia as interações do usuário ao computador - como o apertar de um botão - para o computador acionar o comando associado a tal interação.

O protótipo virtual deve representar as funcionalidades e as características físicas do real, ou seja, deve obedecer a critérios de escala de tamanho, espaço, velocidade e tempo para que os resultados sejam os mais exatos possíveis. Isto é, para que a simulação ocorrida no ambiente com os protótipos virtuais produza uma situação idêntica a que ocorrerá no ambiente real e vice-versa.

Portanto, cada protótipo teve que passar por um estudo minucioso do tamanho, cronometragem dos tempos para realização de cada tarefa, de tal forma que o protótipo virtual fosse uma abstração do real em termos de proporcionalidade e funcionalidades. A Figura 16 apresenta o protótipo virtual guindaste, que é uma réplica do protótipo real correspondente.

Figura 16: Objeto Virtual (Guindaste) que compõe o Ambiente Virtual 3D.

3.2.10 Sistema de Telepresença

A ferramenta baseada em Redes de Petri para o controle do ambiente é executada em um servidor, já o ambiente (abstração do mundo em 3D ou o mundo real) pode ser instanciado em vários clientes, que podem estar geograficamente separados, de forma que cada objeto pode ser virtual ou real. Assim, vários colaboradores poderão visualizar e operar o ambiente através da internet. O emprego da Redes de Petri, permite tratar as concorrências de recursos e o paralelismo de eventos comuns em ambientes virtuais e reais.

4 Associação de Redes de Petri com Objetos Virtuais e Reais

Este trabalho tem como base o metamodelo proposto por Palma (2001), já descrito no capítulo 3, onde foi apresentado uma ferramenta para o desenvolvimento de sistemas de telepresença, que permite a modelagem do controle de ambientes virtuais e reais, através de um modelo gráfico/matemático - Redes de Petri (RP) - com execução do controle centralizado. Através da ferramenta apresentada, a complexidade envolvida para o desenvolvimento de ambientes virtuais e telepresença pode ser minimizada.

A ferramenta é um sistema de autoria que contém uma coleção de aplicativos a fim de editar o modelo de controle, configurar o ambiente de telepresença (virtual e real), executar o controle do ambiente (servidor) e gerar os ambientes (abstração do mundo em 3D ou o mundo real) através de instâncias em vários clientes com objetos pré-definidos.

Com a proposta de Palma (2001), é possível gerar ambientes reais e virtuais com uma coleção de máquinas que atuam de forma integrada, como por exemplo um sistema de manufatura. A qualidade da simulação deve estar atrelada ao modelo desenvolvido e sua validação. Enquanto que a qualidade gráfica, a interação e a imersão vão depender dos recursos da área de realidade virtual que foram empregados.

Um dos componentes deste metamodelo é o Módulo Central de Controle (MCC), responsável em colocar a Rede de Petri (RP) na memória e processá-la para gerenciar o Ambiente Virtual (AV). O MCC realiza o processamento das informações, solucionando os problemas com base no processo que intervém entre a entrada e saída, buscando os objetivos subsequentes de um estado inicial ou de uma interação do usuário. Este módulo controla diretamente um AV, através de uma correspondência dos elementos da RP com os elementos do Ambiente Virtual.

O fato de a ferramenta ser baseada em linguagem de especificação formal como as Redes de Petri, requer uma especificação adequada do modo de associação dos modelos de

RP aos elementos básicos do paradigma orientado à objeto, mais especificamente os objetos virtuais que compõem o ambiente virtual.

Em sua proposta inicial, Palma (2001) utilizou uma Rede de Petri Ordinária para modelar o sistema de ambientes reais/virtuais e posteriormente ser utilizada pelo MCC para o controle do sistema de telepresença. No entanto, esta metodologia apresenta algumas limitações que serão apresentadas na seção 4.1.3. Assim, este trabalho visa formalizar a associação proposta por Palma (2001) e superar as limitações verificadas.

Este capítulo, inicialmente apresenta o modo de associação e/ou relação da rede de petri com os objetos virtuais propostos por (PALMA, 2001). Na sequência, as limitações detectadas em sua abordagem. E por fim, a proposta deste trabalho visando o processo de relacionamento ou associação de redes de petri com objetos reais e virtuais para controle de ambientes virtuais imersivos e telepresença.

4.1 Relação entre os Modelos de RP e os Ambientes Virtuais

Para relacionar os modelos de RP com os ambientes virtuais, Palma (2001) faz uma comparação do comportamento dos modelos de RP e ambientes virtuais, destacando os pontos relevantes e traçando uma correspondência entre deles, como segue:

- em sistemas de simulação baseados em RP, os elementos são lugares, transições, arcos e marcas. Enquanto os AV's são compostos por objetos e estes por sua vez são constituídos por atributos e métodos;
- os lugares, as transições, os arcos e seus respectivos pesos são parâmetros fixos, definidos antes de iniciar qualquer análise (ou simulação) dos modelos. Os objetos com os seus respectivos atributos e métodos são definidos antes da simulação do ambiente em Realidade Virtual;
- o estado da RP em um dado instante é dado por uma marcação M . A marcação M corresponde a um conjunto que relata a quantidade de marcas em cada lugar na rede N em um dado momento. O conjunto de valores existentes nos atributos dos objetos que compõem o ambiente em um dado momento determina o estado do ambiente;
- uma transição t (evento) tem um certo número de lugares de entrada e saída que

representam respectivamente, as pré-condições e as pós-condições de t . Quando as pré-condições da transição são verdadeiras então a transição é dita habilitada e se esta for disparada as pós-condições serão atualizadas. Da mesma forma, os métodos são responsáveis por atualizar os estados dos objetos (mudar os valores dos atributos do objeto) e do ambiente. Quando um método é chamado (o objeto recebe uma mensagem), sua execução causa mudanças no estado do objeto, ou seja, atualiza os valores dos seus atributos;

- os modelos em Redes de Petri podem possuir ligações externas com elementos reais ou virtuais, ou tão somente simbolizar ações internas ao modelo. Ou seja, uma marca em um lugar do modelo corresponde a um elemento real em um determinado estado (condição), ou ainda, uma transição (evento) do modelo pode corresponder a um acontecimento no ambiente. O ambiente virtual pode ter pontos de interação com o usuário do sistema, ou seja, pode possuir sensores externos ao ambiente, como por exemplo, uma tecla, um botão, um dispositivo com sensor (HMD, direção, acelerador), uma alavanca, um equipamento etc. A interação efetuada pelo usuário é capturada pelo AV, que atualiza seu estado. O AV pode efetuar atualizações independentes da interação com usuário.

No metamodelo proposto por Palma (2001), foi utilizado uma RP Ordinária para modelar o sistema de ambientes reais/virtuais e posteriormente ser utilizada pelo Módulo Central de Controle (MCC) para o controle do sistema de telepresença. Dessa forma, a execução do sistema tem o MCC como o responsável por todos os eventos no AV. Isto é possível devido a correspondência existente entre os elementos da RP e os Objetos Virtuais (OV), como descrito anteriormente. Para isso, antes de executar o sistema deve ser realizada a integração dos objetos virtuais com os componentes do modelo de RP. Cujas associação RP x OV está centrada nas relações “Lugar” x “Valor do Atributo do Objeto” e “Transições” x “Métodos dos Objetos”.

Em um sistema criado a partir do metamodelo, o estado do AV é atualizado através de métodos acionados por mensagens enviadas pelo módulo de controle, que tem o modelo de RP na memória. Assim, os métodos dos objetos que estão relacionados com o modelo, por meio das transições do mesmo, só poderão ser executados a mando deste módulo. Quando um método é requisitado, o AV responde afetando alguma sensação humana, com eventos visuais, sonoros ou táteis.

4.1.1 Classes de Objetos Virtuais e Reais

Conforme descrito anteriormente, o metamodelo proposto por Palma (2001) possui uma Biblioteca de Classes de Objetos Virtuais e Reais (BCOVR), que é um repositório dos códigos das abstrações de objetos aptos a comporem uma cena virtual e real. Os detalhes sobre esta biblioteca, pode ser visto seção 3.2.1 do capítulo 3.

As classes de objetos virtuais e reais que compõem a biblioteca, devem ser desenvolvidas pensando na ligação com o modelo de RP. Como o AV é composto de objetos virtuais (instâncias das classes) controlados por um modelo de RP, as classes devem possuir propriedades de forma a facilitar a comunicação com o modelo. Logo, as classes dos objetos virtuais e reais devem possuir métodos para:

- alterar os valores de seus atributos - são invocados quando o estado da RP é alterado, ou seja, quando a rede assume um determinado estado M_k ;
- efetuar a dinâmica do objeto - devem efetuar as transformações geométricas, entre outras atividades ligadas ao comportamento dinâmico que seja perceptível no ambiente virtual, ou seja, quando a rede dispara uma transição t_k ;
- captar as interações do usuário sobre o objeto virtual - quando executados devem enviar mensagens ao controle informando o acontecimento, ou seja, quando o AV recebe um evento do usuário, ele simplesmente remete para o sistema de controle. Este verifica as atividades (transições) que podem ser ativadas no modelo, coloca a marca no respectivo lugar p_k e, se alguma atividade for selecionada, requisita a execução do respectivo método ao AV.

A BCOVR deve ser utilizada indiretamente pelo Configurador de Ambientes Virtuais (CAV), que é o responsável por configurar o AV, informando os objetos a serem instanciados, com as suas respectivas propriedades (valores fornecidos aos atributos da classe). Logo o CAV deve conhecer as estruturas dos objetos virtuais (as classes) que vão compor a cena, para poder configurar o objeto. Através da configuração do objeto enviada pelo CAV ao Editor de Redes de Petri (por meio de arquivo), pode-se modelar e relacionar a RP com o AV.

4.1.2 Associação dos Objetos Virtuais e Reais com uma RP Ordinária

Analisando as associações entre a RP e o paradigma orientado à objetos adotado para o desenvolvimento dos ambientes reais e virtuais, foi possível identificar as seguintes relações:

- um lugar (*place*) do modelo de RP deve ter ligação direta com um determinado valor de um atributo do objeto do ambiente de realidade virtual;
- nem todos os lugares precisam ter correspondentes no ambiente virtual, mas todas as interações do usuário no ambiente devem ser representadas no modelo;
- os objetos virtuais podem mudar de estado com a interação do usuário. Exemplo disso são os botões, alavancas, ou operações mais complexas que podem acionar algum método no AV como mover, ligar ou apagar, dentre outros. Os eventos do AV decorridos da interação do usuário deverão possuir relações diretas com a colocação de marcas nos lugares do modelo em Rede de Petri;
- alguns lugares e transições não possuem nenhuma ligação direta com pontos virtuais, mas são necessários para que se faça uma modelagem coerente do sistema que se pretende simular; e o inverso também pode ser válido, objetos virtuais que não possuem ligação com o modelo, mas podem estar no ambiente para melhorar a visualização do mesmo;
- nem todas as transições precisam ter correspondentes no ambiente virtual, mas todas as ações dependentes de análise devem ser representadas por transições;

Os objetos virtuais e reais deverão trabalhar integrados quando for efetuada a associação dos elementos do modelo de RP com o Ambiente Virtual. Logo, as classes dos objetos virtuais e reais devem estar aptas a suportar esta associação para possibilitar a dinâmica do ambiente quando a simulação for executada.

A associação do modelo de RP com os objetos virtuais e reais, pode ser representada por meio de uma tabela de conexão que contém informações relevantes sobre a relação entre os elementos da RP com os objetos virtuais e reais do ambiente virtual e real configurado, como exibido na Figura 17.

Figura 17: Associação do Modelo de RP com o Ambiente Virtual.

Esta associação da RP com os objetos virtuais e reais, somente é efetuada quando há dados suficientes do modelo em RP e dos objetos do AV. Os dados relevantes para a associação

do modelo de RP são os lugares e as transições. Enquanto, os dados relevantes dos objetos que compõem o ambiente são os métodos e atributos.

Assim, o processo de associação da RP com os objetos corresponde a ligação dos lugares da RP com os valores dos atributos dos objetos, caracterizados como pontos ativáveis do AV; enquanto as transições da RP são ligadas com os métodos dos objetos do AV, resultando em uma tabela de conexão que armazena as informações referentes a esta associação.

Figura 18: A tabela de conexão especifica a ligação dos elementos da RP com os elementos do objeto botão.

A tabela de conexão, descrita na Figura 18, tem a finalidade de identificar os elementos dos objetos que podem ser modificados durante a execução do sistema. Assim, o Módulo Central de Controle (MCC), não apenas possui a RP em memória para realização do controle do ambiente, como possui as informações necessárias para associar os elementos do modelo de RP que está sendo executado com os objetos virtuais e reais.

4.1.3 Limitações na Abordagem de Palma (2001)

Palma (2001) utiliza a RP de forma a permitir que ela seja associada diretamente com os objetos do AV, ou seja, um lugar ou uma transição pode ser associado a somente um objeto, pois assim é possível identificar para qual objeto do AV o modelo deve enviar a mensagem e vice-versa.

Um modelo de RP, modelado conforme os requisitos do metamodelo de Palma (2001), pode ser muito extenso a medida que a complexidade do sistema for aumentando. Ou seja, sua abordagem se depara com uma explosão combinatória quando tenta modelar ambientes

virtuais complexos (muitos objetos). Modelos extensos dificultam a visualização da lógica e do fluxo do sistema (percurso das marcas) e esta pode dificultar e refletir na interligação da RP com o AV. A melhoria do metamodelo será focada em sintetizar o tamanho do modelo para utilização em ambientes virtuais imersivos na área aplicação deste trabalho, sistemas de manufatura.

A modelagem de um sistema de manufatura para controlar um AV envolve a modelagem de todos os recursos desse sistema. Recurso é a denominação dada a todo elemento que compõe o sistema que, de alguma forma, tem a propriedade de modificar o estado de uma determinada variável. Logo, em um sistema de manufatura os recursos podem ser as máquinas, tornos, robôs, empilhadeiras, etc.

Em sistemas de manufatura complexos existem uma variedade de recursos (máquinas, tornos, robôs, empilhadeiras, etc.) que podem ser utilizados em quantidades diferentes. Ou seja, pode haver vários recursos que são utilizados para a mesma finalidade ou cujos processos de alteração do estado do sistema são semelhantes. Neste caso, quando se utiliza a Rede de Petri Ordinária (lugares com marcas não diferenciadas), têm-se duas opções:

- modelar, individualmente, cada um dos processos que constituem o sistema e modelar a interação existente entre eles, o que consiste, em desdobrar o modelo que representa o comportamento geral ou;
- modelar o comportamento geral sem diferenciar a identidade de cada processo, mas apenas a quantidade;

Optando pela primeira opção, o modelo obtido pode ser pouco prático de se trabalhar, seja pelo tamanho da rede, seja pelo número de interações existentes. Esta foi opção adotada por Palma (2001), onde todos os recursos de um sistema de manufatura são modelados de forma unitária (tanto os estados assim como as operações destes recursos), causando um crescimento exponencial no modelo a medida que são acrescentados mais recursos, como pode ser visto na Figura 19.

A partir da Figura 19 fica fácil perceber que a medida que são acrescentados mais AGV's (*Automatic Guided Vehicle*) ao modelo, o tamanho deste tende a crescer demasiadamente, dificultando não só a visualização do modelo mas também sua análise. A Tabela 1 mostra o número de lugares, transições e arcos para este exemplo conforme vão sendo acrescentados mais AGV's. Podemos observar também que o número de arcos no modelo tende a crescer numa velocidade muito mais rápida do que o número de lugares e transições, dificultando assim a visualização do fluxo das marcas no modelo.

Figura 19: Modelagem individual de 1 e 2 AGV's servindo uma máquina, respectivamente.

No. de AGV's	No. de Lugares	No. de Transições	No. de Arcos
1	3	2	6
2	6	4	16
3	8	6	24
6	14	12	40

Tabela 1: Crescimento do número de elementos da RP.

Optando pela segunda opção, obtém-se uma redução do número de elementos da rede, resultando em uma descrição compacta da mesma, como pode ser visto na Figura 20. Mas, como os lugares da RP Ordinária possuem marcas não diferenciadas, torna-se impossível determinar qual recurso (objeto, máquina) está sendo acionado. Como na metodologia orientada à objetos (OO), o ambiente é composto de objetos que possuem características e funcionalidades próprias, ou seja cada um possui sua própria identidade, modelar o comportamento geral sem diferenciar a identidade de cada processo é inviável para a aplicação no controle de ambientes virtuais e reais que utilizam o paradigma OO.

Figura 20: Modelagem de 2 AGV's servindo uma máquina, sem identificação individual.

No modelo da Figura 20, os AGV's correspondem as marcas, os lugares representam os pontos onde eles podem parar (estacionamento ou máquina) e as transições representam a locomoção desses AGV's. A quantidade de marcas iniciais (antes de começar a execução

do modelo) no lugar “Est” significa a quantidade de AGV's que o sistema possui no estacionamento.

Com a segunda opção, o número de elementos da RP permanece fixo, somente a marcação inicial é alterada conforme ocorra um crescimento no número de AGV's do modelo. No entanto, quando a transição “Ir” é ativada o modelo envia um AGV (uma marca) para a o lugar “Maq”, porém não é possível informar ao controle do ambiente qual é, especificamente, o AGV que irá para a máquina. Dessa forma, adotando esta modelagem, o controle não tem como identificar para qual objeto ele deve enviar as mensagens.

Além das limitações descritas acima, a proposta do metamodelo de Palma (2001) descreve um modelo conceitual da tabela de conexão, como apresentado nas Figuras 17 e 18, porém não estabelece como de fato ela será implementada e interpretada pelo Módulo Central de Controle (MCC) para relacionar os elementos da RP em memória com os objetos virtuais e reais que compõem o sistema.

4.2 Associação dos Objetos Virtuais e Reais com uma RP de Alto Nível

A dinâmica de um ambiente virtual imersivo pode ser gerenciada por um sistema de controle, responsável por controlar tanto os objetos virtuais como a ocorrência de eventos no ambiente. Assim, a modelagem dos sistemas de controle de AV's é essencial, visto que, os modelos são a base para a síntese de controle tanto para a análise como para a simulação da sua conduta, ou seja, o modelo pode:

- ser averiguado se corresponde ao comportamento do problema real;
- prognosticar a eficiência do sistema;
- verificar a consistência;
- utilizar ferramentas de modelagem que permitam a computação automática de um algoritmo de controle, sendo especificada pela mudança do estado do modelo permitindo ou proibindo os estados sequenciais.

Embora as RP Ordinárias sejam uma técnica formal poderosa que permite a modelagem de sistemas concorrentes, pela unicidade do tipo de marca, são necessários lugares distintos para representar recursos semelhantes. Logo, sua aplicação à sistemas de manufatura

mais complexos torna-se inadequada, pois os modelos construídos possuem grandes dimensões dificultando sua compreensão e manipulação. Além disso, caso queira-se fazer um refinamento do modelo, aplicando algum tipo de hierarquização, esse tipo de RP é igualmente inadequada. Assim, um número expressivo de extensões às redes de petri, denominadas de Redes de Petri de Alto Nível (RPAN), tem sido proposto (GENRICH; LAUTENBACH, 1981).

4.2.1 Redes de Petri Coloridas (RPC)

Como descrito na seção 2.1 do capítulo 2, as extensões de RP conhecidas como Redes de Petri de Alto Nível (RPAN), propõem uma estruturação com relação aos dados do sistema. Elas se baseiam no conceito de dobramento (ou compactação) de lugares e transições e das marcas como portadoras de informações, ou seja, já não limitam-se a simples presença ou não de marcas em um determinado lugar. O lugar agora pode armazenar informações mais complexas e não se portar apenas como uma variável booleana.

As Redes de Petri Coloridas (RPC), por exemplo, associam de forma estruturada uma parte dos dados às marcas ou *tokens*, utilizando cores para identificação ou diferenciação de suas marcas. Dessa forma, é possível reduzir o tamanho do modelo representado, permitindo que marcas individualizadas, no caso as cores, representem diferentes recursos em uma mesma subrede. Isto é, elas podem fornecer modelos compactos de sistemas complexos, por representar processos ou recursos semelhantes em um mesmo “lugar” ou *place* da rede, de forma que seja possível identificar cada um dos recursos.

Retomando o exemplo descrito na Figura 20, onde houve uma compactação da rede porém sem diferenciação dos objetos, com o uso de uma RPC seria possível informar ao controle do ambiente qual AGV especificamente deverá ir até a máquina. A representação do mesmo exemplo descrito na Figura 20, porém com o uso de marcas diferenciadas, pode ser visto na Figura 21.

Assim, este trabalho propõe a modelagem do controle de ambientes virtuais imersivos com Redes de Petri de Alto Nível (RPAN), mais especificamente Redes de Petri Coloridas (RPC), além do relacionamento ou a associação dos elementos da RP com os objetos virtuais e reais do ambiente virtual e real que compõem o sistema de telepresença.

A modelagem aqui abordada, visa superar as limitações encontradas na proposta de (PALMA, 2001). Através desta abordagem, pretende-se especificar o uso de RPAN no controle dos ambientes, sem nenhum tipo de alteração no método de construção das classes de objetos virtuais e reais do sistema, conciliando a identificação de cada objeto virtual ou real para o

Figura 21: (a) 2 AGV's servindo uma máquina, sem identificação individual. (b) 3 AGV's servindo uma máquina, com identificação individual.

envio de mensagens e uma descrição compacta do sistema.

4.2.2 Associação dos Objetos Virtuais e Reais com uma RPC

De acordo com o metamodelo de Palma (2001), a Biblioteca de Classes de Objetos Virtuais e Reais (BCOVR) é um repositório dos códigos das abstrações de objetos aptos a comporem uma cena virtual e real. A partir das informações fornecidas por esta biblioteca, o Configurador de Ambientes Virtuais (CAV) permite compor a cena virtual de tal forma que o usuário pode definir a disposição dos objetos no ambiente e o estado inicial dos seus atributos, como pode ser visto na Figura 22. Na figura apresentada, os objetos estão em forma de ícones para criar ou representar as instâncias das classes de objetos virtuais da BCOVR.

Figura 22: Composição de uma cena através do CAV.

A partir da composição e configuração dos objetos na cena, o CAV deve fornecer um arquivo contendo informações que descrevem os objetos que constituem a cena e o estado inicial dos mesmos. Apesar de atualmente o CAV fornecer um arquivo texto para ser utilizado no Editor de Redes de Petri, propõe-se a utilização de um arquivo XML (*eXtensible Markup*

Language), por ser um formato flexível, independente de plataforma e que oferece a facilidade de compartilhamento de informações.

De acordo com Teague, Palmer e Jackson (2003) XML é um conjunto de regras que permite uma formatação para definir informações estruturadas em um arquivo texto neutro em termos de *software*. Ou seja, ele pode ser usado para armazenar qualquer tipo de informação estruturada, encapsulando os dados e permitindo a troca de informações entre diferentes sistemas de computação.

Logo, os atributos e métodos dos objetos que compõem a cena, assim como outras informações relevantes como a posição dos mesmos, devem ser definidos em um arquivo XML que descreverá a estrutura da classe ao qual o objeto descrito pertence. A Figura 23, apresenta uma versão simplificada do arquivo XML contendo as informações sobre o objeto botão, ilustrado na Figura 22.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ambiente_virtual>
  <objeto id="b1">
    <nome>botao_0</nome>
    <posicao x="500" y="10" z="224"></posicao>
    <atributos>
      <atributo name="botaoAcionado">FALSE</atributo>
    </atributos>
    <metodos>
      <metodo name="acionaBotao()"></metodo>
    </metodos>
  </objeto>
</ambiente_virtual>
```

Figura 23: Arquivo XML que descreve o objeto botão pertencente a cena configurada no CAV.

Este arquivo XML deve ser armazenado em um repositório e será utilizado pelo Editor de Redes de Petri para gerar o modelo de controle do ambiente em RP e associá-lo aos objetos virtuais e reais. Além disso, o arquivo XML pode ser utilizado para configuração de um Ambiente Real correspondente e pelo Simulador de Ambientes Virtuais (SAV).

O Editor de Redes de Petri, além de permitir a construção de um modelo de RP, deve ser responsável também pela associação do modelo de RP com o objeto modelado. Como a modelagem agora envolve o uso de uma Rede de Petri Colorida (RPC) e não mais uma RP Ordinária, como proposto por no metamodelo de Palma (2001), a Figura 24 descreve o novo modelo de associação dos objetos virtuais e reais com uma RPC, utilizando o objeto botão como exemplo.

Figura 24: Associação do Objeto Virtual Botão com a Rede de Petri Colorida.

Com esta nova abordagem, mantém-se o estado dos objetos associado aos lugares da RP, porém a identificação do objeto é dada pelas diferentes marcas (cores) nos lugares. Como a *cor* em uma RPC pode conter algum tipo de informação sobre o objeto, pode-se distinguir cada objeto através de um nome ou uma letra ou então um número inteiro não-negativo.

No exemplo da Figura 24, os estados dos objetos “b1” e “b2” estão associados aos lugares $p1$ e $p2$, porém diferenciados pelas marcas “a” e “b” respectivamente. Tanto o método do objeto 1 como do objeto 2 estão associados a mesma transição $t1$. A identificação de qual método (do objeto 1 ou do objeto2) deve ser acionado, é realizada pela definição da guarda (*guard*) da transição $t1$. A guarda corresponde a uma proposição lógica dentro de colchetes “[...]” que, se verdadeira, habilita o disparo da transição a qual ela se associa, permitindo assim a distinção de qual objeto está sendo invocado.

Conforme descrito na seção 4.1.3, a proposta do metamodelo de Palma (2001) descreve um modelo conceitual da tabela de conexão, como apresentado nas Figuras 17 e 18, porém não estabelece como de fato ela será implementada e interpretada pelo Módulo Central de Controle (MCC) para relacionar os elementos da RP em memória com os objetos virtuais e reais que compõem o sistema.

Neste trabalho, propõe-se que após a modelagem no Editor de Redes de Petri, seja gerado um arquivo XML para representação da associação dos elementos da RPC com os atributos e métodos dos objetos. O arquivo XML gerado pelo Editor de Redes de Petri, deve definir como os diferentes estados dos objetos estão associados com os lugares da RP, e como

os diferentes eventos estão associados as diferentes transições do modelo de RP.

O arquivo XML que representa a associação descrita na Figura 24, é exibido na Figura 25. No arquivo exibido, o lugar *p1* está associado ao atributo *botaoAcionado* com valor *FALSE*, representa o estado “Botão Não-Acionado”. O lugar *p2* está associado ao atributo *botaoAcionado* porém com valor *TRUE*, representando o estado “Botão Acionado”. E por fim, a transição *t1* está associada ao método *acionarBotao()*, representando o evento “Acionar Botão”. Dessa forma, fica definido como o MCC pode interpretar a conexão entre a RP em memória e os objetos do ambiente.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<associacao>
  <lugar name="p1">
    <atributo name="b1.botaoAcionado">FALSE</atributo>
  </lugar>
  <lugar name="p2">
    <atributo name="b1.botaoAcionado">TRUE</atributo>
  </lugar>
  <transicao name="t1">
    <metodo name="b1.acionaBotao()"></metodo>
  </transicao>
</associacao>
```

Figura 25: Associação da RPC com o objeto virtual botão, representada em XML.

Com essa proposta, é possível processar qualquer RP e tornar o aplicativo genérico para qualquer modelo construído. Além disso, a redução no tamanho e complexidade do modelo de RP simplifica ainda mais a tarefa de criação de Ambientes Virtuais. Ainda com esta abordagem, a RP Colorida pode ser transformada numa RP Ordinária equivalente sem alterar a concepção do metamodelo de (PALMA, 2001). O processo de conversão consiste no desdobramento da rede, onde cada lugar da RPC deverá ser desdobrado em vários lugares da RP Ordinária, cada um representando o estado de apenas um recurso. O mesmo processo vale para as transições e arcos.

Como o MCC trabalha em nível elevado de abstração dos objetos, ou seja ele não precisa distinguir se o cliente é um ambiente que instanciou objetos virtuais ou reais da entidade, quando o MCC iniciar com a RP em memória e o arquivo XML que corresponde a associação da mesma com o objeto virtual, ele poderá de fato controlar tanto o ambiente virtual como o real, de forma que as ações realizadas no ambiente virtual aconteçam da mesma forma no real e vice-versa.

4.3 Estudo de Caso

O capítulo 3, que refere-se ao Sistema de Autoria de Telepresença com Controle em Redes de Petri, descreve que para composição do ambiente real, foram desenvolvidos protótipos com sucata criando-se os seguintes equipamentos: um botão, uma esteira e um guindaste. E para composição do ambiente virtual, foram desenvolvidos protótipos virtuais que representam as funcionalidades e as características físicas do real, ou seja, obedecem a critérios de escala de tamanho, espaço, velocidade e tempo para que os resultados sejam os mais exatos possíveis. Isto é, para que a execução ocorrida no ambiente com os protótipos virtuais produza uma situação idêntica a que ocorrerá no ambiente real e vice-versa.

Dessa forma, a fim de exemplificar o modo de associação dos objetos virtuais e reais com uma Rede de Petri Colorida (RPC), como proposto na seção anterior, foram utilizados os protótipos disponíveis pelo sistema, até o presente momento, como os objetos virtuais e reais a serem associados ao modelo. Portanto, este estudo de caso consiste na construção de modelos de RP Colorida que representam os objetos botão, esteira e guindaste - disponíveis na biblioteca de classes de objetos virtuais e reais - e posteriormente, a associação desses modelos aos respectivos objetos.

Para a construção dos modelos de RPC que representam os protótipos virtuais e reais, foram adotados os passos seguintes:

- **identificação dos estados dos objetos virtuais e reais:** para cada classe existente na Biblioteca de Classes de Objetos Virtuais e Reais, foram identificados os possíveis estados pelos quais cada objeto pode passar. O produto desta atividade resultou em uma lista inicial de estados para cada classe de objetos. O critério para identificar os estados depende das informações obtidas a partir do arquivo de configuração emitido pelo Configurador de Ambientes Virtuais (CAV) ao Editor de Redes de Petri. A princípio, qualquer modificação de valor dos atributos de um objeto representa uma mudança de estado. O analista deve selecionar, através de um processo de abstração, aquelas julgadas relevantes à aplicação.
- **identificação dos eventos dos objetos virtuais e reais:** os objetos que compõem o ambiente virtual e real, possuem um conjunto de eventos que ao serem acionados, provocam alterações em seus estados. Assim como a alteração dos estados pode ser relacionada com alteração dos atributos dos objetos, os eventos estão relacionadas com os métodos disponíveis pelas classes de objetos para cada objeto virtual e real. Esta informação também é obtida através do arquivo emitido pelo CAV ao Editor de Redes

de Petri.

- **construção dos modelos de RPC dos objetos virtuais e reais:** conhecendo a lista das transações dos objetos e os estados que os objetos do ambiente podem assumir, foi iniciada a construção da RPC para modelar os aspectos dinâmicos de cada objeto. Onde, os estados dos objetos foram representados por lugares na RPC e os eventos por transições. É importante lembrar que, nem todos os lugares precisam ter correspondentes no ambiente virtual e real, mas todos os estados assumidos no ambiente devem ser representados no modelo; e que, nem todas as transições precisam ter correspondentes no ambiente virtual e real, mas todas as ações dependentes de análise devem ser representadas por transições;

4.3.1 Associação de uma RPC com os Protótipos Botão/Esteira

Nessa seção, será descrito um estudo de caso utilizando os protótipos “botão” e “esteira” como protótipos reais que compõem o Ambiente Real (AR) e protótipos virtuais (réplicas 3D dos protótipos reais) que compõem o Ambiente Virtual (AV). A configuração do AR é feita manualmente pelo usuário do sistema. O usuário define um espaço físico que representará o seu mini-mundo e dispõe os objetos reais de acordo com o que se deseja que seja simulado no AV.

A cena do AV foi editada através do Configurador de Ambientes Virtuais (CAV), onde foi gerado um arquivo XML contendo informações que descrevem os objetos que constituem a cena e o estado inicial dos mesmos. Uma amostra desse arquivo pode ser visto na Figura 26. Este arquivo XML deve ser armazenado em um repositório e será utilizado pelo Editor de Redes de Petri para gerar o modelo de controle do ambiente em RPC e associá-lo aos objetos virtuais e reais. Além disso, o arquivo XML pode ser utilizado para configuração de um AR correspondente e pelo Simulador de Ambientes Virtuais (SAV).

Através das informações contidas no arquivo XML gerado pelo CAV, foi modelado o comportamento de 2 botões sendo acionados para ligar e desligar duas esteiras, como descrito na Figura 27. O modelo descreve uma situação onde cada botão é responsável pela acionamento de sua respectiva esteira.

Note que há um evento externo (acionar o botão) e dois eventos que ocorrem dependendo do estado em que se encontram as esteiras: quando uma esteira está desligada, ao acionar o seu respectivo botão fará disparar a transição t_2 (ligar esteira) e quando a esteira estiver ligada, ao acionar seu respectivo botão fará disparar a transição t_3 (desligar esteira).

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<ambiente_virtual>
  <objeto id="b1">
    <nome>botao_0</nome>
    <posicao x="500" y="10" z="224"></posicao>
    <atributos>
      <atributo name="botaoAcionado">FALSE</atributo>
    </atributos>
    <metodos>
      <metodo name="acionaBotao()"></metodo>
    </metodos>
  </objeto>
  <objeto id="e1">
    <nome>esteira_0</nome>
    <posicao x="276" y="0" z="224"></posicao>
    <atributos>
      <atributo name="esteiraLigada">FALSE</atributo>
    </atributos>
    <metodos>
      <metodo name="ligaEsteira()"></metodo>
      <metodo name="desligaEsteira()"></metodo>
    </metodos>
  </objeto>
</ambiente_virtual>

```

Figura 26: Arquivo XML que descreve os objetos (botão e esteira) que compõem a cena do ambiente virtual.

Como descrito na Tabela 2, as marcas correspondentes aos botões e as esteiras, são do tipo inteiro, ou seja, elas assumem valores numéricos para distinguir o estado (atributo) de cada objeto, bem como seus respectivos métodos. O que permite ao MCC definir o método de qual objeto deve ser invocado no disparo da transição.

A prioridade e a condição para que cada botão ligue e desligue sua respectiva esteira, é definida pelas guardas (*guard*) das transições t_2 e t_3 . A guarda corresponde a uma proposição lógica dentro de colchetes “[...]” que, se verdadeira, habilita o disparo da transição a qual ela se associa. Assim, sempre que $b = e$, ou seja o valor que corresponde ao botão b é igual ao valor que corresponde à esteira e , apenas a marca referente a b ou e é removida do lugar de saída e colocada no lugar de entrada.

A descrição das transições bem como das variáveis utilizadas nas expressões dos arcos, podem ser vistas nas Tabelas 3 e 4, respectivamente.

Caso houvesse um terceiro botão e uma terceira esteira, não seria necessário alterar o modelo, bastando apenas inserir suas respectivas marcas. O que seria inviável em uma RP Ordinária, pois seria necessário a descrição de todos os seus estados novamente, aumentando exponencialmente as dimensões do modelo.

Figura 27: Modelo em RPC dos objetos botão e esteira.

Após a modelagem da RPC, foi gerado um arquivo XML para representação da associação dos elementos da RPC com os atributos e métodos dos objetos. O arquivo XML define como os diferentes estados dos objetos estão associados com os lugares da RPC, e como os diferentes eventos estão associados as diferentes transições do modelo de RPC.

Lugar	Domínio de Cores	Descrição
p1	Inteiro	Representa o estado <i>Botão Acionado</i>
p2	Inteiro	Representa o estado <i>Esteira Desligada</i>
p3	Inteiro	Representa o estado <i>Esteira Ligada</i>

Tabela 2: Descrição dos lugares apresentados na RPC da Figura 27.

Transição	Descrição
t1	Representa o método <i>acionaBotao()</i>
t2	Representa o método <i>ligaEsteira()</i>
t3	Representa o método <i>desligaEsteira()</i>

Tabela 3: Descrição das transições apresentados na RPC da Figura 27.

O arquivo XML que representa a associação do modelo de RPC com os objetos virtuais, é exibido na Figura 28. No arquivo exibido, o lugar *p1* está associado ao estado ou atributo *botaoAcionado* com o valor booleano *TRUE*, o lugar *p2* ao atributo *esteiraLigada* com

Variável	Tipo	Descrição
b	Inteiro	Identificação do objeto botão
e	Inteiro	Identificação do objeto esteira

Tabela 4: Descrição das variáveis apresentada na RPC da Figura 27.

o valor booleano *FALSE* e *p3* ao atributo *esteiraLigada* com o valor booleano *TRUE*. A transição *t1* está associada ao evento ou método *acionarBotao()*, a transição *t2* ao método *ligaEsteira()* e a transição *t3* ao método *desligaEsteira()*. Dessa forma, fica definido como o MCC pode interpretar a conexão entre a RPC em memória e os objetos do ambiente.

<pre><?xml version="1.0" encoding="ISO-8859-1"?> <associacao> <lugar name="p1"> <marca tipo="Inteiro" valor="1"> <atributo name="b1.botaoAcionado">TRUE</atributo> </marca> <marca tipo="Inteiro" valor="2"> <atributo name="b2.botaoAcionado">TRUE</atributo> </marca> </lugar> <lugar name="p2"> <marca tipo="Inteiro" valor="1"> <atributo name="e1.esteiraLigada">FALSE</atributo> </marca> <marca tipo="Inteiro" valor="2"> <atributo name="e2.esteiraLigada">FALSE</atributo> </marca> </lugar> <lugar name="p3"> <marca tipo="Inteiro" valor="1"> <atributo name="e1.esteiraLigada">TRUE</atributo> </marca> <marca tipo="Inteiro" valor="2"> <atributo name="e2.esteiraLigada">TRUE</atributo> </marca> </lugar> </associacao></pre>	<pre><transicao name="t1"> <marca tipo="Inteiro" valor="1"> <metodo name="b1.acionaBotao()"></metodo> </marca> <marca tipo="Inteiro" valor="2"> <metodo name="b2.acionaBotao()"></metodo> </marca> </transicao> <transicao name="t2"> <marca tipo="Inteiro" valor="1"> <metodo name="e1.ligaEsteira()"></metodo> </marca> <marca tipo="Inteiro" valor="2"> <metodo name="e2.ligaEsteira()"></metodo> </marca> </transicao> <transicao name="t3"> <marca tipo="Inteiro" valor="1"> <metodo name="e1.desligaEsteira()"></metodo> </marca> <marca tipo="Inteiro" valor="2"> <metodo name="e2.desligaEsteira()"></metodo> </marca> </transicao> </associacao></pre>
--	---

Figura 28: Associação dos objetos (botão e esteira) com a RPC.

Como o MCC trabalha em um nível elevado de abstração dos objetos, ou seja ele não precisa distinguir se o cliente é um ambiente que instanciou objetos virtuais ou reais da entidade, quando o MCC iniciar com a RPC em memória e o arquivo XML que corresponde a associação da mesma com o objeto virtual, ele poderá de fato controlar tanto o ambiente virtual como o real, de forma que as ações realizadas no ambiente virtual aconteçam da mesma forma no real e vice-versa.

4.3.2 Associação de uma RPC com o Protótipo Guindaste

Nessa seção, será descrito um estudo de caso utilizando o protótipo “guindaste” como protótipo real que compõem o Ambiente Real (AR) e protótipo virtual (réplica 3D do protótipo real) que compõem o Ambiente Virtual (AV). A configuração do AR é feita manualmente pelo usuário do sistema. O usuário define um espaço físico que representará o seu mini-mundo e dispõe os objetos reais de acordo com o que se deseja que seja simulado no AV.

Assim, como no exemplo dado com os protótipos “botão” e “esteira”, a cena do AV

foi editada através do Configurador de Ambientes Virtuais (CAV), onde foi gerado um arquivo XML contendo informações que descrevem os objetos que constituem a cena e o estado inicial dos mesmos. Uma amostra desse arquivo pode ser visto na Figura 29. Este arquivo XML deve ser armazenado em um repositório e será utilizado pelo Editor de Redes de Petri para gerar o modelo de controle do ambiente em RPC e associá-lo aos objetos virtuais e reais. Além disso, o arquivo XML pode ser utilizado para configuração de um AR correspondente e pelo Simulador de Ambientes Virtuais (SAV).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ambiente_virtual>
  <objeto id="g1">
    <nome>guindaste_0</nome>
    <posicao x="400" y="25" z="154"></posicao>
    <atributos>
      <atributo name="imaLigado">FALSE</atributo>
    </atributos>
    <metodos>
      <metodo name="ligaIma()"></metodo>
      <metodo name="desligaIma()"></metodo>
      <metodo name="sobeIma()"></metodo>
      <metodo name="desceIma()"></metodo>
      <metodo name="giraDireita()"></metodo>
      <metodo name="giraEsquerda()"></metodo>
    </metodos>
  </objeto>
</ambiente_virtual>
```

Figura 29: Arquivo XML que descreve o objeto guindaste que compõe a cena do ambiente virtual.

Através das informações contidas no arquivo XML gerado pelo CAV, foi modelado o comportamento do protótipo guindaste, ao transportar produtos de um ponto A (origem) para um ponto B (destino) através de um eletroímã conectado à ponta do cabo. Os produtos que estão na origem, encontram-se sempre à *esquerda* do guindaste, enquanto o destino está sempre à *direita* do mesmo. A RPC que representa esta dinâmica é apresentada na Figura 30.

Como descrito na Tabela 5, as marcas correspondentes aos guindastes e aos produtos, são do tipo inteiro, ou seja, elas assumem valores numéricos para distinguir o estado (atributo) de cada objeto, bem como seus respectivos métodos. O que permite ao MCC definir o método de qual objeto deve ser invocado no disparo da transição.

De acordo com a RPC descrita na Figura 30, o objeto guindaste só pode pegar produtos ao encontrar-se pronto para pegá-los e ainda, caso existam produtos na origem. As marcas no lugar p_2 indicam instâncias de diferentes guindastes. Para cada guindaste instanciado, deverá existir uma marca que o represente neste mencionado lugar. Além disso, um guindaste $g = 1$, só pode pegar produtos com a mesma identificação $p = 1$. Dessa forma,

Figura 30: Modelo em RPC do objeto guindaste.

pode-se controlar o comportamento de vários objetos guindastes trabalhando no transporte de seus respectivos produtos, através da identificação de suas respectivas marcas.

Lugar	Domínio de Cores	Descrição
p1	Inteiro	Representa o estado <i>Produto na origem</i>
p2	Inteiro	Representa o estado <i>Guindaste pronto para pegar produto</i>
p3	Inteiro	Representa o estado <i>Eletróimã ligado</i>
p4	Inteiro	Representa o estado <i>Eletróimã pronto para subir</i>
p5	Inteiro	Representa o estado <i>Eletróimã em cima</i>
p6	Inteiro	Representa o estado <i>Guindaste pronto para girar para direita</i>
p7	Inteiro	Representa o estado <i>Eletróimã pronto para descer</i>
p8	Inteiro	Representa o estado <i>Eletróimã em baixo</i>
p9	Inteiro	Representa o estado <i>Guindaste pronto para soltar produto</i>
p10	Inteiro	Representa o estado <i>Eletróimã desligado</i>
p11	Inteiro	Representa o estado <i>Produto no destino</i>
p12	Inteiro	Representa o estado <i>Guindaste pronto para girar para esquerda</i>
p13	Inteiro	Representa o estado <i>Guindaste pronto para reiniciar</i>

Tabela 5: Descrição dos lugares apresentados na RPC da Figura 30.

A prioridade e a condição para que cada guindaste pegue seu respectivo produto, é definida pelas guardas (*guard*) das transições $t1$ e $t5$. A guarda corresponde a uma proposição

lógica dentro de colchetes “[...]” que, se verdadeira, habilita o disparo da transição a qual ela se associa. Assim, sempre que $p = g$, ou seja o valor que corresponde ao guindaste g é igual ao valor que corresponde ao produto p , apenas a marca referente a g ou p é removida do lugar de saída e colocada no lugar de entrada.

Transição	Descrição
t1	Representa o método <i>ligalma()</i>
t2	Representa o método <i>sobelma()</i>
t3	Representa o método <i>giraDireita()</i>
t4	Representa o método <i>descelma()</i>
t5	Representa o método <i>desligalma()</i>
t6	Representa o método <i>giraEsquerda()</i>
t7	Representa o método <i>relnicia()</i>

Tabela 6: Descrição das transições apresentados na RPC da Figura 30.

A descrição das transições bem como das variáveis utilizadas nas expressões dos arcos, podem ser vistas nas Tabelas 6 e 7, respectivamente.

Variável	Tipo	Descrição
g	Inteiro	Identificação do guindaste
p	Inteiro	Identificação do produto

Tabela 7: Descrição das variáveis apresentada na RPC da Figura 30.

À medida que mais guindaste vão sendo inseridos no ambiente virtual, não há necessidade de alterar o modelo, bastando apenas inserir suas respectivas marcas ao modelo já existente. O que seria inviável em uma RP Ordinária, pois seria necessário a descrição de todos os estados de cada guindaste novamente, aumentando exponencialmente as dimensões do modelo.

Após a modelagem da RPC, foi gerado um arquivo XML para representação da associação dos elementos da RPC com os atributos e métodos dos objetos. O arquivo XML define como os diferentes estados dos objetos estão associados com os lugares da RPC, e como os diferentes eventos estão associados as diferentes transições do modelo de RPC.

É importante lembrar que, nem todos os lugares precisam ter correspondentes no ambiente virtual e real, mas todos os estados assumidos no ambiente devem ser representadas no modelo; e que, nem todas as transições precisam ter correspondentes no ambiente virtual e real, mas todas as ações dependentes de análise devem ser representadas por transições.

O arquivo XML que representa a associação do modelo de RPC com o objeto guindaste, é exibido na Figura 31. No arquivo exibido, o lugar $p3$ está associado ao estado ou atributo *imaLigado* com o valor booleano *TRUE* e o lugar $p10$ ao atributo *imaLigado* com

o valor booleano *FALSE*. A transição *t1* está associada ao evento ou método *ligalma()*, *t2* ao método *sobelma()*, *t3* ao método *giraDireita()*, *t4* ao método *sobelma()*, *t5* ao método *desligalma()* e a transição *t6* ao método *giraEsquerda()*. Dessa forma, fica definido como o MCC pode interpretar a conexão entre a RPC em memória e os objetos do ambiente.

<pre><?xml version="1.0" encoding="ISO-8859-1"?> <associacao> <lugar name="p3"> <marca tipo="Inteiro" valor="1"> <atributo name="g1.imaligado">TRUE</atributo> </marca> </lugar> <lugar name="p10"> <marca tipo="Inteiro" valor="1"> <atributo name="g1.imaligado">FALSE</atributo> </marca> </lugar> <transicao name="t1"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.ligaIma()"></metodo> </marca> </transicao> <transicao name="t2"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.sobeIma()"></metodo> </marca> </transicao> <transicao name="t3"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.giraDireita()"></metodo> </marca> </transicao> <transicao name="t4"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.desceIma()"></metodo> </marca> </transicao> <transicao name="t5"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.desligaIma()"></metodo> </marca> </transicao> <transicao name="t6"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.giraEsquerda()"></metodo> </marca> </transicao> </associacao></pre>	<pre><transicao name="t3"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.giraDireita()"></metodo> </marca> </transicao> <transicao name="t4"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.desceIma()"></metodo> </marca> </transicao> <transicao name="t5"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.desligaIma()"></metodo> </marca> </transicao> <transicao name="t6"> <marca tipo="Inteiro" valor="1"> <metodo name="g1.giraEsquerda()"></metodo> </marca> </transicao> </associacao></pre>
--	---

Figura 31: Associação do objeto guindaste com a Rede de Petri Colorida.

Como o MCC trabalha em um nível elevado de abstração dos objetos, ou seja ele não precisa distinguir se o cliente é um ambiente que instanciou objetos virtuais ou reais da entidade, quando o MCC iniciar com a RPC em memória e o arquivo XML que corresponde a associação da mesma com o objeto virtual, ele poderá de fato controlar tanto o ambiente virtual como o real, de forma que as ações realizadas no ambiente virtual aconteçam da mesma forma no real e vice-versa.

4.4 Resultados e Discussão

Um dos módulos da ferramenta de autoria, a Biblioteca de Classes de Objetos Virtuais e Reais (BCOVR), compreende a representação das classes dos objetos virtuais e reais sob o paradigma de orientação à objetos. Isso, resulta em clareza na concepção do controle de cada objeto, simplificando a comunicação entre o projetista e o sistema de autoria, além da atualização do sistema através da facilidade de acréscimo de novos objetos.

Como o controle dos métodos dos objetos virtuais e reais a serem invocados é feito pelo Módulo Central de Controle (MCC) a partir de uma Rede de Petri (RP), a associação da RP com os objetos virtuais e reais é essencial para gerenciamento dos ambientes que compõem o sistema de telepresença.

Com a utilização de uma RP Ordinária, a construção do modelo lógico do sistema pode, freqüentemente, tornar-se difícil quando refere-se à sistemas de manufatura complexos.

Nestes casos, o modelo pode se tornar muito extenso, dificultando a legibilidade, consumindo tempo e ocasionando erros no processo de modelagem. Visando diminuir a complexidade dessa tarefa, a utilização de extensões de RP que permitam a redução dos elementos do modelo é altamente desejável.

A utilização de elementos codificados por cor nas Redes de Petri Coloridas (RPC), ofereceu maior capacidade para o controle distinguir entre os diferentes tipos de equipamentos, informações e fluxo de recursos. Essa capacidade é fundamental para o controle de sistemas virtuais e reais complexos como os sistemas de manufatura.

Ao associar os modelos de RPC aos elementos básicos do paradigma Orientado à Objeto (OO), mais especificamente aos objetos virtuais e reais de manufatura, para projetar o controle de ambientes virtuais imersivos e telepresença, obteve-se uma redução na complexidade dos modelos resultantes. Para o engenheiro de simulação, a redução do tamanho do modelo pode significar não apenas um ganho visual importante, mas também uma redução no tempo de modelagem e a uma possível diminuição de erros típicos de sistemas muito extensos.

Apesar da representação mais compacta, o método proposto manteve o mesmo poder de representação das Redes de Petri Ordinárias. Esta equivalência é vantajosa pelo fato de que os métodos de análise proporcionados pelas redes ordinárias continuam válidas no método proposto. Ou seja, além das vantagens de redução de tempo de partida e manutenção do sistema, pela obtenção de um sistema mais consiso, pode-se usufruir da análise matemática e simulação desses modelos para efeitos de verificação e validação do mesmo.

Os resultados obtidos com esta pesquisa demonstraram a capacidade de distinção entre os objetos a serem invocados pelo MCC durante o gerenciamento do sistema de realidade virtual e telepresença. Além disso, os modelos de RPC podem ser utilizados para avaliar a dinâmica do sistema antes de seu funcionamento de fato, e assim, efetuar as devidas modificações caso seja necessário. A simulação dos modelos pode ser utilizada para aperfeiçoar os processos existentes ou desenvolver novos processos. Assim, é possível obter-se a visualização prévia de movimentos e permitir a detecção de erros ou visualização de possibilidades de melhorias antes mesmo que se faça uma implantação em um ambiente definitivo, onde não espera-se a ocorrência de falhas.

O uso do XML facilitou a especificação formal da tabela de conexão que representa a associação dos elementos da RPC com os elementos dos objetos virtuais e reais, uma vez que esta notação tem sido bastante difundida e utilizada em diversos sistemas de computação, como meio de integrar e possibilitar a comunicação entre sistemas diferentes, como protocolo

de comunicação entre processos, etc. A utilização do XML, portanto, constituiu um fator positivo para facilitar a escrita da especificação formal por programadores que não são especialistas em formalizar sistemas através de métodos formais como as Redes de Petri.

Com relação ao controle do ambiente, a associação da RPC com os objetos virtuais e reais apresentada, permitiu um controle dos métodos a serem invocados com a identificação do estado atual de cada objeto presente nos ambientes. Como o MCC trabalha em nível elevado de abstração dos objetos, ou seja ele não precisa distinguir se o cliente é um ambiente que instanciou objetos virtuais ou reais da entidade, quando o MCC iniciar com a RPC em memória e o arquivo XML que corresponde a associação da mesma com o objeto virtual, ele poderá de fato controlar tanto o ambiente virtual como o real, de forma que as ações realizadas no ambiente virtual aconteçam da mesma forma no real e vice-versa.

5 Conclusão

A complexidade na construção e controle de ambiente virtuais inibe a aplicação da tecnologia de RV e da telepresença. O metamodelo proposto por Palma (2001), por sua vez, facilita o emprego de protótipos reais e virtuais em aplicações envolvendo ambientes virtuais imersivos e telepresença. Com a proposta apresentada é possível gerar ambientes com uma coleção de máquinas que atuam de forma integrada, como por exemplo um sistema de manufatura, onde a qualidade da simulação deve estar atrelada ao modelo desenvolvido e sua validação, enquanto que a qualidade gráfica, a interação e a imersão vão depender dos recursos da área de realidade virtual que foram empregados.

O metamodelo descrito no capítulo 3 envolve a modelagem do controle de ambientes virtuais e reais, através de um modelo gráfico/matemático (Redes de Petri) com execução do controle centralizado. Com o desenvolvimento de diversos trabalhos relacionados ao metamodelo, foi possível obter um sistema de autoria que contém uma coleção de aplicativos a fim de editar o modelo de controle, configurar o ambiente de telepresença (virtual e real), executar o controle do ambiente (servidor) e gerar os ambientes (abstração do mundo em 3D ou o mundo real) através de instâncias em vários clientes com objetos pré-definidos.

Um dos módulos da ferramenta de autoria, a Biblioteca de Classes de Objetos Virtuais e Reais (BCOVR), compreende a representação das classes dos objetos virtuais e reais sob o paradigma de orientação à objetos. Isso, resulta em clareza na concepção do controle de cada objeto, simplificando a comunicação entre o projetista e o sistema de autoria, além da atualização do sistema através da facilidade de acréscimo de novos objetos.

Como o controle dos métodos dos objetos virtuais e reais a serem invocados é feito pelo Módulo Central de Controle (MCC) a partir de uma Rede de Petri (RP), a associação da RP com os objetos virtuais e reais é essencial para gerenciamento dos ambientes que compõem o sistema de telepresença.

Com a utilização de uma RP Ordinária, a construção do modelo lógico do sistema pode, freqüentemente, tornar-se difícil quando refere-se à sistemas de manufatura complexos.

Nestes casos, o modelo pode se tornar muito extenso, dificultando a legibilidade, consumindo tempo e ocasionando erros no processo de modelagem. Visando diminuir a complexidade dessa tarefa, a utilização de extensões de RP que permitam a redução dos elementos do modelo é altamente desejável.

Para superar esta limitação e alcançar os objetivos mencionados no capítulo introdutório, foi utilizado uma Rede de Petri de Alto Nível, mais especificamente uma Rede de Petri Colorida (RPC). A utilização de elementos codificados por cor nas RPC's, ofereceu maior capacidade para o controle distinguir entre os diferentes tipos de equipamentos, informações e fluxo de recursos. Essa capacidade é fundamental para o controle de sistemas virtuais e reais complexos como os sistemas de manufatura.

Ao associar os modelos de RPC aos elementos básicos do paradigma Orientado à Objeto (OO), mais especificamente aos objetos virtuais e reais de manufatura, para projetar o controle de ambientes virtuais imersivos e telepresença, obteve-se uma redução na complexidade dos modelos resultantes. Para o engenheiro de simulação, a redução do tamanho do modelo pode significar não apenas um ganho visual importante, mas também uma redução no tempo de modelagem e a uma possível diminuição de erros típicos de sistemas muito extensos.

Apesar da representação mais compacta, o método proposto manteve o mesmo poder de representação das Redes de Petri Ordinárias. Esta equivalência é vantajosa pelo fato de que os métodos de análise proporcionados pelas redes ordinárias continuam válidas no método proposto. Ou seja, além das vantagens de redução de tempo de partida e manutenção do sistema, pela obtenção de um sistema mais consiso, pode-se usufruir da análise matemática e simulação desses modelos para efeitos de verificação e validação do mesmo.

Os resultados obtidos com esta pesquisa demonstraram a capacidade de distinção entre os objetos a serem invocados pelo MCC durante o gerenciamento do sistema de realidade virtual e telepresença. Além disso, os modelos de RPC podem ser utilizados para avaliar a dinâmica do sistema antes de seu funcionamento de fato, e assim, efetuar as devidas modificações caso seja necessário. A simulação dos modelos pode ser utilizada para aperfeiçoar os processos existentes ou desenvolver novos processos. Assim, é possível obter-se a visualização prévia de movimentos e permitir a detecção de erros ou visualização de possibilidades de melhorias antes mesmo que se faça uma implantação em um ambiente definitivo, onde não espera-se a ocorrência de falhas.

Foi estabelecido uma estrutura de arquivo em XML para reger a escrita de uma especificação que possibilitasse ao MCC identificar o estado e as requisições de cada objeto

participante de uma cena virtual e real, e assim, ser capaz de impedir a chamada de métodos de objetos errados ou em ordem incorreta.

O uso do XML facilitou a especificação formal da tabela de conexão que representa a associação dos elementos da RPC com os elementos dos objetos virtuais e reais, uma vez que esta notação tem sido bastante difundida e utilizada em diversos sistemas de computação, como meio de integrar e possibilitar a comunicação entre sistemas diferentes, como protocolo de comunicação entre processos, etc. A utilização do XML, portanto, constitui um fator positivo para facilitar a escrita da especificação formal por programadores que não são especialistas em formalizar sistemas através de métodos formais como as Redes de Petri.

Com relação ao controle do ambiente, a associação da RPC com os objetos virtuais e reais apresentada, permitiu um controle dos métodos a serem invocados com a identificação do estado atual de cada objeto presente nos ambientes. Como o MCC trabalha em nível elevado de abstração dos objetos, ou seja ele não precisa distinguir se o cliente é um ambiente que instanciou objetos virtuais ou reais da entidade, quando o MCC iniciar com a RPC em memória e o arquivo XML que corresponde a associação da mesma com o objeto virtual, ele poderá de fato controlar tanto o ambiente virtual como o real, de forma que as ações realizadas no ambiente virtual aconteçam da mesma forma no real e vice-versa.

5.1 Contribuições

Através da especificação do modo de associação da RP com o ambiente virtual, foi possível complementar e enriquecer o trabalho iniciado por Palma (2001), consolidando as funcionalidades da ferramenta resultante do metamodelo proposto, com a inserção de melhorias ao mesmo.

A associação das RP com os objetos do ambiente, permite que o modelo que descreve os ambientes virtuais e reais não seja utilizado apenas para simulação e análise do sistema, mas para o próprio controle do mesmo através da definição de prioridades de invocação de métodos de objetos, durante a execução das aplicações de RV imersivas e de telepresença.

Além disso, com o estudo realizado e apresentado neste texto, este trabalho fornece subsídios para a continuidade no desenvolvimento de aplicações de realidade virtual em ambientes imersivos e de telepresença, contribuindo para o crescimento do *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos* que oferece uma infraestrutura de visualização 3D imersiva (uma CAVE) disponível para o desenvolvimento de pesquisas, principalmente no contexto de desenvolvimento da Manufatura Virtual.

5.2 Trabalhos Futuros

Como perspectivas imediatas de trabalho tem-se dois aspectos que cabem aqui ressaltar. Ambos dizem respeito à aplicação efetiva da proposta apresentada, e não do conceito envolvido na associação das RP com os objetos virtuais e reais.

O primeiro refere-se a adequação do sistema de autoria às melhorias aqui propostas. Logo, o Configurador de Ambientes Virtuais (CAV) deve gerar um arquivo XML que descreve os objetos virtuais a serem modelados, conforme apresentado no capítulo anterior. O Editor de Redes de Petri, por sua vez, deve gerar o XML que descreve associação das Redes de Petri Coloridas (RPC) com os objetos virtuais e reais que foram modelados, para posterior utilização através do Módulo Central de Controle (MCC) ao executar a RP em memória. E por fim, o MCC deve ser adequado à utilização da RPC ao invés da RP Ordinária.

O segundo refere-se a aplicação do sistema de autoria de telepresença com controle em RP para controle e operação de objetos reais remotos através de objetos virtuais projetados na CAVE do *Laboratório Multiusuário de Visualização 3D Imersiva de São Carlos*. Ou seja, através da modelagem dos objetos e do processo de interação utilizando os aspectos aqui abordados, será possível manipular um objeto virtual 3D em um ambiente virtual imersivo e ainda operar objetos reais remotos por telepresença que se comportam como os objetos virtuais apresentados ao ambiente imediato ao usuário.

A utilização do paradigma orientado à objetos na implementação da Biblioteca de Classes dos Objetos Virtuais e Reais (BCOVR) e do módulo Central de Controle (MCC) proporciona a capacidade de definir-se outras extensões de RP que integram a formalização matemática com os conceitos de abstração, herança e polimorfismo.

A integração das RP com o paradigma orientado à objetos, pode fornecer suporte aos aspectos de modularidade. Ou seja, permite a construção de sistemas complexos que envolve um vasto número de elementos, através da combinação de pequenas redes que descrevem módulos específicos do sistema.

Apesar da redução significativa do modelo proporcionada pelo uso de RP Coloridas, quando tratamos de sistemas de dimensões macroscópicas, é interessante combiná-la com as Redes Hierárquicas que, apesar de não acrescentar nenhum poder de modelagem, é útil por questões de organização e conveniência gráfica.

Assim, como perspectivas não-imediatas tem-se a possibilidade de introduzir o conceito de hierarquia aos modelos de RPC, utilizando assim uma extensão chamada Redes de

Petri Coloridas Hierárquicas (RPCH). As RPCH são capazes de representar modelos com alto nível de abstração, melhor representação gráfica, maior compactação, e dados estruturados que permitem a construção modular da rede de forma similar às usadas em linguagens de programação estruturada.

A capacidade de uma RPCH modelar um sistema hierarquicamente resulta do fato de que uma rede inteira pode ser substituída por um simples lugar ou transição, em um nível mais abstrato e compacto, denominados, respectivamente, de lugar de fusão e transição de substituição. Assim, lugares e transições podem ser substituídos por sub-redes para prover um modelo mais detalhado.

Por fim, futuramente pode-se ainda preparar o Servidor de Ambientes Virtuais (SAV) para identificar dispositivos *plug and play* ou configurá-lo através do manual do dispositivo. Logo, em cada uma das instâncias do ambiente virtual pode-se ter grau de imersão diferenciado, uma vez que os dispositivos serão empregados apenas para interação com os objetos do ambiente, sem fazerem parte do mesmo. Os dispositivos são apenas vestimentas para proporcionar a sensação de presença e não interfere diretamente na lógica e comportamento do ambiente. Apenas as ações proporcionadas pela interação destes dispositivos, são enviados para o servidor tal como uma ação do mouse, do teclado ou diretamente do dispositivo no protótipo real.

Referências

- ADAMS, L. *Visualização e Realidade Virtual*. São Paulo, SP: Makron Books, 1994.
- ARAUJO, R. B. *Especificação e Análise de um Sistema Distribuído de Realidade Virtual*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 1996.
- AZUMA, R. Tracking requirements for augmented reality. *Communications of the ACM*, v. 36, n. 7, p. 50–51, 1993.
- BEGAULT, D. R. *3-D Sound for Virtual Reality and Multimedia*. San Diego, CA, USA: Academic Press Professional, Inc., 1994.
- BEIER, K.-P. A virtual reality check for manufacturers - a short introduction to virtual reality. *ASTD - American Society for Training and Development*, v. 9, n. 1, p. 14, February 1998.
- BEIER, K.-P. Web-based virtual reality in design and manufacturing applications. *Proceedings of COMPIT'2000 - 1st International EuroConference on Computer Applications and Information Technology in the Maritime Industry*, Potsdam, Germany, p. 45–55, March 2000.
- BENTON, S. A. *Selected Papers on Three-Dimensional Displays*. Bellingham, WA - USA: SPIE Optical Engineering Press, 2001.
- BIERBAUM, A. et al. Vr juggler: A virtual platform for virtual reality application development. *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*, p. 89, 2001.
- BILLINGHURST, M.; KATO, H.; POUPYREV, I. The magicbook - moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, v. 21, n. 3, p. 6–8, 2001.
- BOMAN, D. K. International survey: Virtual-environment research. *IEEE Computer*, v. 28, n. 6, p. 57–65, 1995.
- BOUGUILA, L.; ISHII, M.; SATO, M. Realizing a new step-in-place locomotion interface for virtual environment with large display system. *EGVE '02: Proceedings of the Workshop on Virtual environments 2002*, p. 197–207, 2002.
- BOURAS, C. et al. X3d multi-user virtual environment platform for collaborative spatial design. *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, p. 40, 2007.
- BROWN, K.; PETERSEN, D. *Ready-to-Run Java 3D*. New York, NY: John Wiley & Sons, 1999.
- BURDEA, G. C. *Force and Touch Feedback for Virtual Reality*. New York, NY, USA: John Wiley & Sons, Inc., 1996.

- CAI, Y.; ISHII, M.; SATO, M. A human interface device for cave: Size virtual workspace. *IEEE International Conference on Systems, Man, and Cybernetics*, v. 3, p. 2084–2089, 1996.
- CAPIN, T. K. et al. *Avatars in Networked Virtual Environments*. New York, NY, USA: John Wiley & Sons, 1999.
- CAREY, R. The virtual reality modeling language explained. *IEEE MultiMedia*, v. 5, n. 3, p. 84–93, 1998.
- CHAPIN, W.; LACEY, T.; LEIFER, L. Designspace: A manual interaction environment for computer aided design. *Proceedings of CHI '94*, p. 24–28, 1994.
- CHEN, S. E. Quicktime vr: An image-based approach to virtual environment navigation. *Siggraph '95: Proceedings of The 22nd Annual Conference on Computer Graphics and Interactive Techniques*, p. 29–38, 1995.
- CHURCHILL, E. F.; SNOWDON, D. Collaborative virtual environments: An introductory review of issues and systems. *Virtual Reality: Research, Development and Applications*, v. 3, n. 1, p. 3–15, 1998.
- CORSEUIL, E. T. L. et al. Buscando o uso operacional de realidade virtual em grandes modelos de engenharia. *VI Symposium on Virtual Reality - SVR*, p. 187–198, 2003.
- CREAGH, H. Cave automatic virtual environment. *Proceedings of Electrical Insulation Conference and Electrical Manufacturing & Coil Winding Technology Conference*, p. 499–504, 2003.
- CRUZ-NEIRA, C. *Virtual Reality Based on Multiple Projection Screens: The Cave and its Applications to Computational Science and Engineering*. Tese (Doutorado) — University of Illinois, Chicago, IL, USA, 1995.
- CRUZ-NEIRA, C.; SANDIN, D. J.; DEFANTI, T. A. Surround-screen projection-based virtual reality: The design and implementation of the cave. *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, p. 135–142, 1993.
- CRUZ-NEIRA, C. et al. The cave: Audio visual experience automatic virtual environment. *Communications of the ACM*, v. 35, n. 6, p. 64–72, 1992.
- CZERNUSZENKO, M. et al. The immersadesk and infinity wall projection-based virtual reality displays. *ACM SIGGRAPH Computer Graphics*, v. 31, n. 2, p. 46–49, 1997.
- DALY, L.; BRUTZMAN, D. X3d: Extensible 3d graphics standard [standards in a nutshell]. *IEEE Signal Processing Magazine*, v. 24, n. 6, p. 130–135, 2007.
- DODGSON, N. A. Autostereoscopic 3d displays. *Computer (IEEE)*, v. 38, n. 8, p. 31–36, 2005.
- DOERR, K.-U. et al. Evaluation of a low-cost 3d sound system for immersive virtual reality training systems. *IEEE Transactions on Visualization and Computer Graphics*, v. 13, n. 2, p. 204–212, 2007.
- DRAPER, J. V.; KABER, D. B.; USHER, J. M. Speculations on the value of telepresence. *CyberPsychology & Behavior*, v. 2, n. 4, p. 349–362, 1999.

- EARNSHAW, R. A.; VINCE, J. A.; JONES, H. *Virtual Reality Applications*. London, UK, UK: Academic Press Ltd., 1995.
- ELLIS, S. R. What are virtual environments? *IEEE Computer Graphics and Applications*, v. 14, n. 1, p. 17–22, 1994.
- FARIA, R. R. A. *Auralização em Ambientes Audiovisuais Imersivos*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, São Paulo, SP, 2005.
- FELEZ, J.; VERA, C.; MARTINEZ, M. L. Virtual reality applied to traffic accident analysis. *Computer Networks*, v. 30, n. 20, p. 1907–1914, 1999.
- FERNANDES, K. J.; RAJA, V.; EYRE, J. Cybersphere: The fully immersive spherical projection system. *Communications of the ACM*, v. 46, n. 9, p. 141–146, 2003.
- FISHER, S. S. Virtual environments, personal simulation & telepresence. *Virtual Reality: Theory, Practice and Promise*, p. 101–110, 1991.
- FISHER, S. S.; TAZELAAR, J. M. Living in a virtual world. *Byte*, v. 15, n. 7, p. 215–221, 1990.
- GENRICH, H. J.; LAUTENBACH, K. System modelling with high level petri nets. *Theoretical Computer Science*, v. 13, p. 109–133, 1981.
- GIBSON, W. *Neuromancer*. New York, USA: ACE Books, 1984.
- GILLET, A. et al. Augmented reality with tangible auto-fabricated models for molecular biology applications. *Proceedings of The 15th IEEE Visualization 2004 (VIS'04)*, p. 235–242, 2004.
- GOLDBERG, K. *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*. Cambridge, MA, USA: The MIT Press, 2000.
- GRADECKI, J. D. *The Virtual Reality Construction Kit*. New York, NY - USA: John Wiley & Sons Inc, 1995.
- GROSS, M. et al. blue-c: A spatially immersive display and 3d video portal for telepresence. *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, San Diego, CA, USA, p. 819–827, 2003.
- HAIQING, L.; GUOFU, Y.; JIE, F. Research on the collaborative virtual products development based on web and x3d. *16th International Conference on Artificial Reality and Telexistence—Workshops (ICAT'06)*, p. 141–144, 2006.
- HANÁK, D. et al. Building a panoramic recording and presentation system for telepresence. *(VRIC'2007) 9th Virtual Reality International Conference*, Laval, France, p. 185–190, April 2007.
- HARRISON, S.; DOURISH, P. Re-place-ing space: The roles of place and space in collaborative systems. *CSCW '96: Proceedings of the 1996 ACM conference on Computer Supported Cooperative Work*, New York, NY, USA, p. 67–76, 1996.
- HELD, R. M.; DURLACH, N. I. Telepresence. *Presence: Teleoperators and Virtual Environments*, v. 1, n. 1, p. 109–112, 1992.

- HOFFMANN, H.; STEFANI, O.; PATEL, H. Extending the desktop workplace by a portable virtual reality system. *International Journal of Human-Computer Studies*, v. 64, n. 3, p. 170–181, 2006.
- HORVATH, L.; RUDAS, I. J. Application of part manufacturing process model in virtual manufacturing. *Proceedings of ISIE'99 (IEEE)*, v. 3, p. 1367–1372, 1999.
- HUANG, T. et al. A virtual prototyping system for simulating construction processes. *Automation in Construction*, v. 16, n. 5, p. 576–585, 2006.
- HUGHES, C. E. et al. Mixed reality in education, entertainment, and training. *IEEE Computer Graphics and Applications*, v. 25, n. 6, p. 24–30, 2005.
- IMMERSION, C. 3d interaction technologies. <http://www.immersion.com/>, 2008.
- ISHII, H. et al. A study on constructing a machine-maintenance training system based on virtual reality technology. *Proceedings of The 5th International Topical Meeting on Nuclear Thermal Hydraulics, Operations and Safety*, p. 1–6, 1997.
- IWATA, H. et al. Circulafloor. *SIGGRAPH '04: ACM SIGGRAPH 2004 Emerging Technologies*, 2004.
- IWATA, H.; YANO, H.; TOMIYOSHI, M. String walker. *ACM SIGGRAPH 2007 Emerging Technologies*, 2007.
- JACOBSON, J. Configuring multiscreen displays with existing computer equipment. *Proceedings of Conference on Human Factors HFES'2002*, p. 760–765, 2002.
- JACOBSON, J.; LEWIS, M. Game engine virtual reality with caveut. *Computer (IEEE)*, v. 38, n. 4, p. 79–82, 2005.
- JACOBSON, L. *Realidade Virtual em Casa*. Rio de Janeiro, RJ: Berkeley, 1994.
- JENSEN, K. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. New York, NY, USA: Springer-Verlag New York, Inc., 1997.
- JEONG, S.; HASHIMOTO, N.; MAKOTO, S. A novel interaction system with force feedback between real - and virtual human: An entertainment system: "virtual catch ball". *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, p. 61–66, 2004.
- JUAN, M. C. et al. Using augmented reality to treat phobias. *IEEE Computer Graphics and Applications*, v. 25, n. 6, p. 31–37, 2005.
- KALAWSKY, R. *The Science of Virtual Reality and Virtual Environments*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1993.
- KAUFMANN, C.; LIU, A.; BURRIS, D. Dti autostereographic display: Initial evaluation. In: WESTWOOD, J. D. et al. (Ed.). *Medicine Meets Virtual Reality 2000*. Amsterdam, Netherlands: IOS Press, 2000. p. 156–158.
- KIM, S.; KOIKE, Y.; SATO, M. Tension based 7 dofs force feedback device: Spidar-g. *Transactions on Control, Automation and Systems Engineering*, v. 4, n. 1, p. 9–16, 2002.

- KIMURA, K.; IBARAKI, J. Visualization of satellite orbital motions by virtual reality. *Proceedings of the Symposium on Celestial Mechanics*, v. 31, p. 413–419, 1999.
- KIRNER, C.; KIRNER, T. G. Virtual reality and augmented reality applied to simulation visualization. In: *Simulation and Modeling: Current Technologies and Applications*. Hershey, NY, USA: IGI Publishing, 2007. p. 391–419.
- KIRNER, C.; TORI, R. *Realidade Virtual: Conceitos e Tendências*. São Paulo, SP: Editora Mania de Livro, 2004.
- KRUEGER, M. W.; GIONFRIDDO, T.; HINRICHSEN, K. Videoplace - an artificial reality. *CHI '85: Proceedings of The Sigchi Conference On Human Factors In Computing Systems*, p. 35–40, 1985.
- KUNZLER, U.; RUNDE, C. Kinesthetic haptics integration into large-scale virtual environments. *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, p. 551–556, 2005.
- LATTA, J. N.; OBERG, D. J. A conceptual virtual reality model. *IEEE Computer Graphics and Applications*, v. 14, n. 1, p. 23–29, 1994.
- LESTON, J. Virtual reality: The it perspective. *The Computer Bulletin*, v. 38, n. 3, p. 12–13, June 1996.
- LIPTON, L. *Foundations of the Stereoscopic Cinema*. New York, NY - USA: Van Nostrand Reinhold, 1982.
- LOKKI, T. et al. Creating interactive virtual auditory environments. *IEEE Computer Graphics and Applications*, v. 22, n. 4, p. 49–57, 2002.
- MACHADO, L. S. *A Realidade Virtual no Modelamento e Simulação de Procedimentos Invasivos em Oncologia Pediátrica: Um Estudo De Caso no Transplante de Medula Óssea*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, 2003.
- MAIR, G. M. Telepresence - the technology and its economic and social implications. *Proceedings of the IEEE International Symposium on Technology and Society*, Glasgow, UK, p. 118–124, June 1997.
- MASCARENHAS, R. et al. Modeling and analysis of a virtual reality system with time petri nets. *ICSE '98: Proceedings of the 20th International Conference on Software Engineering*, Kyoto, Japan, p. 33–42, April 1998.
- MCDERMOTT, S. An exploration of virtual 3d sound. *Acoustical Society of America*, New Orleans, Louisiana, 2007.
- MILGRAM, P. et al. Augmented reality: A class of displays on the reality-virtuality continuum. *SPIE - Telemanipulator and Telepresence Technologies*, v. 2351, p. 282–292, 1994.
- MINSKY, M. Telepresence. *Omni*, p. 45–51, June 1980.
- MONTEIRO, E. F.; PAIVA, A. C. de. Um ambiente para a visualização de informações baseado em realidade virtual não-imersiva. *Proceedings of V Synposium on Virtual Reality*, p. 395–397, 2002.

- MORIE, J. F. Inspiring the future: Merging mass communication, art, entertainment and virtual environments. *ACM SIGGRAPH Computer Graphics*, v. 28, n. 2, p. 135–138, 1994.
- MURATA, T. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, April 1989.
- NETTO, A. V.; MACHADO, L. S.; OLIVEIRA, M. C. F. *Realidade Virtual: Fundamentos e Aplicações*. Florianópolis, SC: Visual Books, 2002.
- NOVINT, T. I. Novint falcon. <http://home.novint.com/>, 2008.
- OBERHEID, H. A coloured petri net model of cooperative arrival planning in air traffic control. *Proceedings of the Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, p. 177–196, October 2006.
- OLIVEIRA, C. et al. A real object control system for use in manufacturing telepresence environments. *COBEM 2007- Proceedings of the 19th International Congress of Mechanical Engineering*, 2007.
- OLIVEIRA, C. et al. Cavadapt: Configurador de ambientes virtuais adaptável aplicado na criação e edição de ambientes virtuais de manufatura. *CIBIM 8 - Memorial 8 Congresso Iberoamericano de Ingeniería Mecánica*, Cusco, Peru, 2007.
- PALMA, J. G. *Metamodelo para a Modelagem e Simulação de Sistemas a Eventos discretos, baseado em Redes de Petri e Realidade Virtual. Uma Aplicação em Sistema de Manufatura*. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2001.
- PARK, C.-H.; JANG, G.; CHAI, Y.-H. Development of a virtual reality training system for live-line workers. *International Journal of Human-Computer Interaction*, v. 20, n. 3, p. 285–303, 2006.
- PARK, S. H. et al. Virtual knee joint replacement surgery system. *Geometric Modeling and Imaging*, v. 4, n. 6, p. 79–84, 2007.
- PETRI, C. A. *Kommunikation mit Automaten*. Tese (Doutorado) — Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.
- PIMENTEL, K.; TEIXEIRA, K. *Virtual Reality: Through the New Looking Glass*. New York, NY, USA: McGraw-Hill, Inc., 1994.
- PORTO, A. J. V. et al. Manufatura virtual: Conceituação e desafios. *Gestão & Produção*, v. 9, n. 3, p. 297–312, 2002.
- RAPOSO, A. B. *Coordenação em Ambientes Colaborativos Usando Redes de Petri*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação, Univerdade Estadual de Campinas, Campinas, 2000.
- RASKAR, R. et al. The office of the future: A unified approach to image-based modeling and spatially immersive displays. *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, p. 179–188, 1998.

- RHEE, S.-M.; PARK, J. Low-cost telepresence for collaborative virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, Piscataway, NJ, USA, v. 13, n. 1, p. 156–166, 2007.
- RHEINGOLD, H. *Virtual Reality*. New York: Summit Books, 1991.
- ROBERTSON, G. G.; CARD, S. K.; MACKINLAY, J. D. Three views of virtual reality: Nonimmersive virtual reality. *Computer (IEEE)*, v. 26, n. 2, p. 81–83, February 1993.
- ROSENBERG, L. Virtual haptic overlays enhance performance in telepresence tasks. *Proceedings of SPIE Telemanipulator and Telepresence Technologies*, v. 2351, p. 99–108, 1995.
- RYU, J. et al. A game engine based architectural simulator on multi-projector displays. *Proceedings of The 11th Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA)*, v. 4, p. 613–615, 2006.
- SCHENK, M.; STRASSBURGER, S.; KISSNER, H. Combining virtual reality and assembly simulation for production planning and worker qualification. *Proc. of The International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, p. 411–414, 2005.
- SCHWAIGER, M. C.; THÜMMEL, T.; ULBRICH, H. Cyberwalk: Implementation of a ball bearing platform for humans. *Lecture Notes in Computer Science*, v. 4551, p. 926–935, 2007.
- SCHWEBELA, D. C.; GAINESA, J.; SEVERSON, J. Validation of virtual reality as a tool to understand and prevent child pedestrian injury. *Accident Analysis & Prevention*, 2008.
- SENSABLE, T. I. Haptic devices. <http://www.sensable.com/>, 2008.
- SHERIDAN, T. B. Musings on telepresence and virtual presence. *Presence: Teleoperators and Virtual Environments*, v. 1, n. 1, p. 120–126, 1992.
- SHERMAN, W. R.; CRAIG, A. *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- SHRIDHAR, J. M.; RAVI, S. Virtual manufacturing: An important aspect of collaborative product commerce. *Journal of Advanced Manufacturing Systems*, Great Britain, v. 1, p. 113–119, 2002.
- SMITH, S.; DUKE, D. Virtual environments as hybrid systems. *Proceedings of Eurographics UK 17th Annual Conference: EG-UK99*, 1999.
- SOMMERVILLE, I.; SAWYER, P. *Requirements Engineering: A Good Practice Guide*. New York, NY, USA: John Wiley & Sons, Inc., 1997.
- SORID, D.; MOORE, S. K. The virtual surgeon. *IEEE Spectrum*, v. 37, n. 7, p. 26–31, 2000.
- STEUER, J. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, v. 42, n. 4, p. 73–93, 1992.
- STUART, R. *The Design of Virtual Environments*. New York: McGraw-Hill, Inc., 1996.

- SUSNIK, R. et al. Spatial sound generation using hrtf created by the use of recursive filters. *Proceedings of EUROCON 2003. Computer as a Tool. The IEEE Region 8*, v. 1, p. 449–453, 2003.
- TAKACS, B.; SIMON, L. A clinical virtual reality rehabilitation system for phobia treatment. *Proceedings of the 11th International Conference Information Visualization*, p. 798–806, 2007.
- TAN, P. S.; HASHIMOTO, N.; SATO, M. Game-engine based virtual environments for immersive projection display systems. *Eizo Joho Media Gakkai Gijutsu Hokoku*, v. 29, n. 17, p. 29–32, 2005.
- TEAGUE, T.; PALMER, M.; JACKSON, R. Xml for capital facilities. *Leadership and Management in Engineering*, v. 3, n. 2, p. 82–85, 2003.
- TIFFIN, J.; TERASHIMA, N. *HyperReality: Paradigm for the Third Millennium*. London, UK: Routledge, 2001.
- VALLIUS, T.; RÖNING, J. A telepresence robot system realized by embedded object concept. *Proceedings of SPIE, The International Society for Optical Engineering*, Boston, MA, USA, v. 6384, p. 638403.1–638403.13, October 2006.
- VINCE, J. *Introduction to Virtual Reality*. London: Springer-Verlag, 2004.
- VORAA, J. et al. Using virtual reality technology for aircraft visual inspection training: Presence and comparison studies. *Applied Ergonomics*, v. 33, n. 6, p. 559–570, 2002.
- WANG, L.; SHEN, W.; LANG, S. Wise-shopfloor: A web-based and sensor-driven e-shop floor. *Journal of Computing and Information Science in Engineering*, v. 4, p. 56–60, March 2004.
- WILLIAMS, J. S.; MICHAEL, D. H.; SMITH, S. P. Implementing virtual environment object behaviour from a specification. *Proceedings of the Workshop on Guiding Users through Interactive Experiences - Usability Centred Design and Evaluation of Virtual 3D Environment*, 2001.
- WILSON, S. *Information Arts: Intersections of Art, Science, and Technology*. Cambridge, MA, USA: The MIT Press, 2003.
- YANG, D. et al. The teleoperation system of space robot based on virtual technology. *Mechatronics and Automation, 2005 IEEE International Conference*, Niagara Falls, Canada, v. 4, p. 2156–2159, August 2005.
- YING, J.; GRACANIN, D. Petri net model for subjective views in collaborative virtual environments. *Proceedings of the 4th International Symposium on Smart Graphics*, p. 128–134, 2004.
- YU, C.; WU, H.; WU, M. A complete model for collaborative virtual environment integrating extended fuzzy-timing petri nets with role and agent technology. *IEEE International Conference On Systems, Man and Cybernetics*, 2004.

- ZHOU, Y.; MURATA, T.; DEFANTI, T. Modeling and analysis of collaborative virtual environments by using extended fuzzy-timing petri nets. *Workshop on Application of Petri Nets to Intelligent System Development, 20th International Conference on Applications and Theory of Petri Nets*, 1999.
- ZHOU, Z. et al. Interactive entertainment systems using tangible cubes. *Proceedings of Australian Workshop on Interactive Entertainment*, p. 19–24, 2004.
- ZIMMERMAN, T. G. et al. A hand gesture interface device. *Chi '87: Proceedings Of The SIGCHI/GI - Conference on Human Factors in Computing Systems and Graphics Interface*, p. 189–192, 1987.