



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ERICK AUGUSTO GOMES DE MELO

**ARTHRON: UMA FERRAMENTA PARA
GERENCIAMENTO E TRANSMISSÃO DE MÍDIAS
EM PERFORMANCES ARTÍSTICO-
TECNOLÓGICAS**

JOÃO PESSOA - PB

NOVEMBRO, 2010

ERICK AUGUSTO GOMES DE MELO

**ARTHRON: UMA FERRAMENTA PARA
GERENCIAMENTO E TRANSMISSÃO DE MÍDIAS
EM PERFORMANCES ARTÍSTICO-
TECNOLÓGICAS**

Dissertação de Mestrado apresentada ao Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, como requisito parcial para obtenção do Título de Mestre em Informática (Sistemas de Computação)

Orientador: Prof. Dr. Dênio Mariz Timóteo de Sousa

JOÃO PESSOA - PB

NOVEMBRO, 2010

ERICK AUGUSTO GOMES DE MELO

**ARTHRON: UMA FERRAMENTA PARA
GERENCIAMENTO E TRANSMISSÃO DE MÍDIAS
EM PERFORMANCES ARTÍSTICO-
TECNOLÓGICAS**

Dissertação de Mestrado apresentada ao Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, como requisito parcial para obtenção do Título de Mestre em Informática (Sistemas de Computação)

BANCA EXAMINADORA:

Prof. Dr. Dênio Mariz Timóteo de Sousa

Prof. Dra. Tatiana Aires Tavares

Prof. Dr Aquiles Medeiros Filgueira Burlamaqui

AGRADECIMENTOS

A força maior que me guia, Deus. Sempre presente em todos os momentos.

A minha mãe, Darci Gomes de Melo, que sempre me mostrou o melhor caminho a seguir, o do esforço, abdicção e dedicação em tudo que se faz.

A minha esposa, Nayara Santos M Neiva de Melo que sempre me incentivou através da sua perseverança e vontade de vencer na vida.

A minha filha, Lara Neiva de Melo, que trouxe a tona todo meu amor e ternura, transformando a minha vida e me tornando uma pessoa melhor.

A meu orientador, Dr. Dênio Mariz, por acreditar no meu trabalho e apoio prestado.

Aos meus mentores e amigos, Dr. Guido Lemos e Dra. Tatiana Tavares, que sempre se mostraram presentes e solícitos, ensinando e aconselhando-me no que fosse necessário.

Aos meus companheiros de equipe e amigos, Andre Castelo, Julio César, Alexander de Almeida, Elenilson Vieira, Rennan Toscano e Maria Clara, que sem a entrega e dedicação não seria possível esse trabalho.

Aos meus companheiros e amigos de trabalho árduo do Laboratório de Aplicações de Vídeo Digital – LAViD.

A RNP que proporcionou o auxílio necessário para continuidade e desenvolvimento desse trabalho.

Aos colegas do laboratório NatalNet que contribuíram em vários momentos na concepção deste trabalho.

Por fim, a todos aqueles que direta e indiretamente contribuíram para conclusão deste trabalho o meu muito obrigado.

“Quando as portas da percepção forem abertas,
veremos tudo como realmente é: infinito”

(William Blake)

RESUMO

Para elaboração e execução de um espetáculo midiático compartilhado e distribuído necessita-se de pessoal especializado e de um bom planejamento para que os erros possam ser minimizados. Além do aparato tecnológico necessário, é imprescindível que toda equipe esteja em plena sinergia e os artistas tenham que se preocupar com mais um ponto importante, não no tocante a parte artística, mas que influencia bastante nesse tipo de espetáculo, que são os inevitáveis atrasos causados pela rede durante a transmissão dos fluxos de mídia ao vivo. É nesse contexto de entrelaçamento entre a Arte e a Tecnologia que surgiu a necessidade do desenvolvimento de uma ferramenta que apoiasse de forma sistemática a realização de espetáculos desse tipo. A principal funcionalidade dessa ferramenta em software, chamada Arthron, é oferecer ao usuário uma interface simples para manipulação de diferentes fontes e fluxos de mídia simultâneos pré-gravados ou ao vivo. Dessa forma, o usuário pode, remotamente, adicionar, remover, configurar o formato de apresentação e programar exibição no tempo (quando apresentar?) e no espaço (onde apresentar?) dos fluxos de mídia em um espetáculo artístico-tecnológico.

Palavras-chave: Arte e Tecnologia; Multimídia; Mídias Digitais; Sistemas Distribuídos.

ABSTRACT

Developing and implementing a shared, distributed media spectacle requires some dedicated staff and a good plan so that errors can be minimized. Besides the technological necessary apparatus, it is imperative that everyone in the team be in complete synergy and artists have to worry about one more important point, not regarding to the artistic aspects, but regarding unavoidable technical aspects related to delays caused by network during transmission of live media streams. In this context of entanglement between Art and Technology that it was necessary to develop a tool that is able to support in a systematic way the performance of such shows. This software tool, named Arthron, offer to the user a simple interface for simultaneously manipulate different sources and pre-recorded or live media streams. Thus, the users can remotely add, remove, set the presentation format and adjust exhibition scheduling in time (when to present?) and space (where to present?) of media streams in a technological-artistic performance.

Keywords: Art and Technology, Multimedia, Digital Media, Distributed System

LISTA DE ILUSTRAÇÕES

Figura 1 - Um modelo de classificação para as tecnologias multimídia [15].	19
Figura 2 - Destaques da performance de dança colaborativa [25].	29
Figura 3 - Projeto da peça com definição de estágios, dançarinos e agentes virtuais [26].	30
Figura 4 - Cenas da peça onde dançarinas interagem com as projeções [27].	33
Figura 5 - Fotos do espetáculo Versus.	36
Figura 6 - Especificação do cenário de transmissão do espetáculo (In)TOQue. Observe que os	36
Figura 7 - Dados de entrada suportados pelo INTERACT [32].	38
Figura 8 - Interface gráfica do SLTV	42
Figura 9 - Diagrama de casos de uso UML.	44
Figura 10 - Arquitetura inicial da Arthron.	45
Figura 11 - Redução do fluxo no enlace da rede com a utilização do componente Reflector. A) Envio de três fluxos utilizado o Reflector no Encoder. B) Envio de três fluxos utilizado o Reflector separadamente.	46
Figura 12 - Arquitetura atual da Arthron.	46
Figura 13 - Principais classes e pacotes do pacote commons.	47
Figura 14 - Principais classes do componente Decoder.	50
Figura 15 - Principais classes do componente Encoder.	53
Figura 16 - Principais classes do componente Reflector.	55
Figura 17 - Chat entre o Articulator e um Encoder. a) A cor do Encoder é alterada.	58
Figura 18 - Página gerada automaticamente a partir da escolha de um Decoder.	59
Figura 19 - Principais classes do componente Articulator.	59
Figura 20 - Principais classes do componente MapManager.	62
Figura 21 - Principais classes do componente VideoServer.	64
Figura 22 - Principais classes do componente Monitor.	65
Figura 23 - Principais classes do componente ScenarioMaker.	67
Figura 24 - Pilha de dependência das tecnologias utilizadas pela Arthron.	68
Figura 25 - Mapa esquemático e-Pormundos Afeto.	78
Figura 26 - Cenário de atuação da Arthron.	80
Figura 27 - Tela do Articulator da Arthron controlando o “backstage” do espetáculo e-Pormundos Afeto.	81
Figura 28 - Elementos do espetáculo e-Pormundos Afeto no palco do Teatro do Dragão do Mar.	82
Figura 29 - Cenário de execução do experimento no WRNP 2010.	85
Figura 30 - Análise do número de acesso gerado pelo Google Analytics.	86
Figura 31 - Gráfico com a descrição da conexão dos usuários.	87
Figura 32 - Gráfico com a análise da qualidade de transmissão com o flowplayer.	87
Figura 33 - Gráfico com a análise da qualidade de transmissão com o HTML 5.	88
Figura 34 - Gráfico com a análise da qualidade de transmissão com o plugin do vlc.	89

LISTA DE TABELAS

Tabela 1- Comparativo entre as ferramentas.	42
Tabela 2 – Protocolo do agente robótico Galatéia.	52
Tabela 3 - Configuração cênica ponto A.	79
Tabela 4 - Configuração cênica ponto C.	79
Tabela 5 - Roteiro de câmeras Brasil/Espanha.	84
Tabela 6 – Comparação da Arthron com outras ferramentas.	90

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO	13
1.2	OBJETIVOS	14
1.2.1	<i>Objetivos Gerais</i>	14
1.2.2	<i>Objetivos Específicos</i>	15
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO.....	15
2	FUDAMENTAÇÃO TEÓRICA	16
2.1	ARTE E TECNOLOGIA	16
2.2	TECNOLOGIAS PARA MULTIMÍDIA	18
2.2.1	<i>Representação de Informação Multimídia</i>	19
2.2.2	<i>Sistemas Multimídia</i>	21
2.2.3	<i>Serviços Multimídia</i>	22
2.2.3.1	<i>Vídeo ao vivo</i>	23
2.2.3.2	<i>Protocolos Multimídia</i>	23
2.2.4	<i>Conteúdo Multimídia</i>	24
3	TRABALHOS CORRELATOS.....	26
3.1	NA INSTITUIÇÃO	26
3.2	ESPETÁCULOS	27
3.2.1	<i>Uma Performance de Dança Colaborativa</i>	28
3.2.2	<i>Mesclando Realidades – Um Performance Artística com Colaboração</i>	29
3.2.3	<i>Uma Performance de Dança Interativa Baseada no Movimento</i>	32
3.2.4	<i>Um Estudo de Caso sobre Colaboração em Arte e Tecnologia</i>	33
3.2.5	<i>Experiências Realizadas</i>	35
3.2.5.1	<i>Versus</i>	35
3.2.5.2	<i>(IN)Toque</i>	36
3.3	FERRAMENTAS EXISTENTES	37
4	A ARTHRON	43
4.1	VISÃO GERAL	43
4.2	MODELO CONCEITUAL.....	43
4.3	MODELO ARQUITETURAL	44
4.4	COMPONENTES	47
4.4.1	<i>O Pacote Commons</i>	47
4.4.2	<i>Decoder</i>	49
4.4.3	<i>Encoder</i>	51
4.4.4	<i>Reflector</i>	54

4.4.5	<i>Articulator</i>	56
4.4.6	<i>MapManager</i>	62
4.4.7	<i>VideoServer</i>	63
4.4.8	<i>Monitor</i>	64
4.4.9	<i>ScenarioMaker</i>	66
4.5	DETALHES DE IMPLEMENTAÇÃO	67
4.5.1	<i>Java</i>	68
4.5.2	<i>VLC</i>	68
4.5.3	<i>JVLC</i>	69
4.5.4	<i>Iperf</i>	69
4.5.5	<i>SNMP</i>	70
5	AMBIENTE DE EXECUÇÃO	71
5.1	SOFTWARE	71
5.2	HARDWARE	71
5.3	REDE	73
6	CENÁRIOS DE APLICAÇÃO	74
6.1	IDENTIFICAÇÃO DO PÚBLICO ALVO	74
6.2	CENÁRIOS	75
7	EXPERIÊNCIAS REALIZADAS	76
7.1	E-PORMUDOS AFETO	76
7.1.1	<i>Concepção Artística</i>	76
7.1.2	<i>Concepção Técnica</i>	80
7.1.3	<i>Dificuldades e Problemas</i>	82
7.1.4	<i>Soluções</i>	83
7.1.5	<i>Resultados</i>	83
7.2	TRANSMISSÃO DO WRNP 2010	85
7.2.1	<i>Análise da Transmissão</i>	86
7.2.1.1	<i>Conectividade</i>	86
7.2.1.2	<i>Qualidade da Transmissão</i>	87
7.2.2	<i>Resultados</i>	89
8	CONSIDERAÇÕES FINAIS	90
8.1	TRABALHOS FUTUROS	91
9	REFERÊNCIAS	92

1 INTRODUÇÃO

Esta Dissertação foi desenvolvida com base no projeto do Grupo de Trabalho de Mídias Digitais e Arte (GT MDA¹) do programa de grupos de trabalho da Rede Nacional de Ensino e Pesquisa (RNP).

A RNP é responsável pela promoção do desenvolvimento de redes, incluindo suas aplicações e serviços inovadores, especialmente para o benefício de suas organizações usuárias, que se utilizam da infraestrutura de pesquisa em rede *Internet* avançada (*Internet2*).

A *Internet2* uma iniciativa norte-americana voltada para desenvolvimento de tecnologias e aplicações avançadas de redes *Internet* para as comunidades acadêmicas e de pesquisa, que envolve mais de 200 universidades norte-americanas, além de agências do governo e indústria e visa ao desenvolvimento de novas aplicações como telemedicina, bibliotecas digitais, laboratórios virtuais, entre outras que não são viáveis com a tecnologia *Internet* atual [1].

É no contexto das aplicações da *Internet2* em que se insere a Arthron², uma ferramenta que permite o entrelaçamento de agentes humanos e sintéticos em espaços midiáticos compartilhados e distribuídos, em tempo real, através de redes de altíssima velocidade e com grande volume de informação. Esse entrelaçamento tem a Arte e Tecnologia como protagonistas.

1.1 MOTIVAÇÃO

Hoje passamos por uma transformação na área das Tecnologias da Informação e Comunicação (TICs) caracterizada pela universalização das formas de conectividade (redes de computadores, celulares, TV) e popularização dos dispositivos midiáticos. Novas possibilidades emergem das “super” redes de computadores, as quais lidam com grande volume de dados e alto poder de transmissão. Por exemplo, iniciativas como *Internet2* e o Projeto Giga [2] que abrem novos cenários para atuação das TICs. Um desses cenários é a convergência entre Arte e Tecnologia através de uma infraestrutura

¹ <http://gtmda.lavid.ufpb.br>

² É uma palavra que vem do grego, onde significa "articulação".

de software, hardware e serviços que conecte em tempo real pessoas e mídias digitais com propósito de colaborar em manifestações artísticas.

A ferramenta Arthron se utilizou da experiência e dos resultados obtidos pela equipe do Laboratório de Aplicações de Vídeo Digital (LAVID) nas Fases I e II do Grupo de Trabalho de Vídeo Digital (GT VD) [3, 4] e do Grupo de Trabalho de TV Digital (GT TV) [5, 6] e GigaVR [7] para implementação de um arcabouço de software e hardware especialmente projetado para apoiar a realização de eventos midiáticos distribuídos e em tempo-real.

Pensando nesse contexto o LAVID e a RNP, através de desdobramentos de experimentos artístico-tecnológicos como o “Versus” [8] e “In ToQue” [9], e percebendo a necessidade de um processo sistêmico para o provimento da infraestrutura tecnológica que forneça o arcabouço ferramental imprescindível para apoiar a realização desse tipo de manifestação artístico cultural, criou o GT MDA, destinado ao desenvolvimento de uma ferramenta que tem por principal funcionalidade oferecer ao usuário uma interface simples para manipulação de diferentes fontes e fluxos de mídia simultâneos. Dessa forma o usuário pode remotamente adicionar, remover, configurar o formato de apresentação e programar a exibição no tempo (quando apresentar?) e no espaço (onde apresentar?) dos fluxos de mídia em um evento.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O objetivo desse trabalho é o desenvolvimento de uma arquitetura de software para suporte a criação de eventos distribuídos de cunho artístico cultural e tecnológico, capaz de gerenciar e controlar sistematicamente de forma simplificada as mídias envolvidas. Como prova de conceito da arquitetura foi desenvolvida a ferramenta Arthron.

Além disso, há outros interesses na sistematização dessa infraestrutura. Do ponto de vista tecnológico é importante ter ferramentas para monitoração e medição de dados que possibilitem o estudo do impacto desse tipo de experiência nas redes de alta velocidade. Dessa forma, é possível estudar estratégias e políticas mais eficientes para avaliar qualidade ao atendimento do serviço provido pela rede, especialmente, em

tempo real e com mídias de alta definição (e conseqüentemente grande volume de dados).

1.2.2 Objetivos Específicos

O objetivos deste trabalho são:

- Estudo de ambientes multimídia, aplicações e tecnologias relacionadas;
- Análise de ferramentas semelhantes;
- Definição de uma arquitetura distribuída da ferramenta;
- Especificar, modelar e implementar a ferramenta;
- Integrar com outros sistemas existentes de forma transparente; e
- Realizar um experimento para demonstrar viabilidade e avaliar os resultados alcançados com ferramenta.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está dividida em sete capítulos. Seguindo esta introdução, tem-se no capítulo 2 a fundamentação teórica pertinente a esse trabalho. No capítulo 3 são descritos os trabalhos relacionados e um breve comparativo com outras ferramentas. O capítulo 4 trata da arquitetura da ferramenta, definindo o papel de cada componente, tecnologias utilizadas no desenvolvimento e detalhes de implementação da ferramenta. O capítulo 5 discute a infraestrutura necessária para execução da ferramenta. O capítulo 6 apresenta os possíveis cenários de aplicação da ferramenta. O capítulo 7 apresenta os resultados obtidos com o uso da ferramenta, através da realização de dois experimentos. O capítulo 8 apresenta as considerações finais.

2 FUDAMENTAÇÃO TEÓRICA

Este capítulo nos insere no contexto teórico das principais áreas do conhecimento que fomentam este trabalho: Arte e Tecnologia e Tecnologias para Multimídia.

2.1 ARTE E TECNOLOGIA

Desde a apresentação, em 1966, do espetáculo “9 Evenings: Theatre and Engineering”, organizado pela EAT (do inglês: *Experiments in Art and Technology*) associação fundada logo após a realização do espetáculo, que foi uma série de apresentações de performance artística que uniu artistas e engenheiros, onde durante dez meses foram explorados recursos tecnológicos como vídeo projeção, o microfone sem fio e outros dispositivos eletrônicos [10], que a interação entre Arte e a Tecnologia Contemporânea (computadores, robôs, rede de computadores, etc) não parou mais de acontecer.

Hoje, diferentemente dos anos 60, o artista situa a tecnologia na sua essência, deslocando para o processo em que está inserida. Querendo criar uma obra aberta e absorver as influências externas e imprevisíveis, propondo trocas, alterando o objeto artístico, transformando-o em um organismo vivo, mutável, dinâmico, que permite não um resultado único, mas uma variabilidade de criações e de propostas criadoras e de criadores [10].

O Brasil apresenta uma trajetória de cerca de cinquenta anos de história no campo das poéticas tecnológicas. Nos anos 50, as primeiras experiências com arte cinética foram realizadas por Abraham Palatinik³. A década de 60 foi marcada pelo surgimento da música eletroacústica, por iniciativa inicialmente de Jorge Antunes e pela introdução do computador na arte, por Waldemar Cordeiro. Desde seu surgimento as poéticas tecnológicas se definiram muito rapidamente entre nós com pelo menos duas características mais marcantes: (a) sintonia e sincronia com o que estava sendo produzido fora do Brasil, o que dava aos brasileiros uma condição de atualidade, quando não até mesmo de precocidade em alguns casos específicos; (b) ao mesmo

³ Abraham Palatinik (Natal, 1928) é um artista plástico brasileiro. É um dos pioneiros e a maior referência em arte cinética no Brasil. Suas obras contêm instalações elétricas que criam movimentos e jogos de luzes.

tempo e paradoxalmente uma certa diferença de abordagem, motivada principalmente pelo veio crítico de boa parte dos trabalhos, fruto do enfrentamento de uma trágica realidade social e de uma vida política massacrada por uma ditadura militar, o que tornava as obras brasileiras um tanto distintivas com relação ao que se fazia no exterior.

As gerações seguintes, que enveredaram pelos terrenos da vídeo-arte, *computer art*, *computer music*, artecomunicação, holografia, poesia intersemiótica e intersecção arte ciência (para citar apenas os campos que mais se desenvolveram no Brasil nos anos 1980 e 1990), um pouco mais aliviadas dos constrangimentos, pelo menos no campo político, deram continuidade aos princípios dos pioneiros e fizeram expandir o campo de experiências de modo a abarcar quase todo o universo das poéticas tecnológicas [11].

A popularização dos computadores, a generalização da Internet, a convergência midiática e as inovações tecnológicas de toda sorte já ultrapassaram os limites dos laboratórios científicos e hoje fazem parte do cotidiano de um número cada vez mais significativo da sociedade urbana. Segundo Machado [11] à medida que o mundo natural, tal como conheceram as gerações de outros séculos, vai sendo substituído pela tecnosfera – a natureza criada ou modificada pela ciência –, novas realidades se impõem: de um lado, aumento das expectativas de vida, incremento da produtividade, multiplicação das riquezas materiais e culturais, mudanças profundas nos modos de existir, circular, relacionar-se, perceber e representar o mundo, campo fértil para experiências artísticas inovadoras.

A aproximação entre Arte e Tecnologia é mais que uma tendência da Sociedade Moderna é uma realidade. A arte (qualquer arte, de qualquer tempo) sempre foi produzida com os meios tecnológicos de seu tempo, a apropriação que ela faz do aparato tecnológico que lhe é contemporâneo. Em geral, aparelhos, instrumentos e máquinas semióticas não são projetados para a produção de arte, pelo menos não no sentido secular desse termo, tal como ele se constituiu no mundo moderno a partir mais ou menos do século XV.

No entanto, no cenário de convergência que vivenciamos, atualmente, estamos em meio a uma metamorfose rumo a uma sociedade tecnológica que é culturalmente composta por imagens, sons, textos, vídeos e outras mídias. A cultura formada pelas “novas tecnologias” pode aumentar o nosso conhecimento e consciência do mundo

onde vivemos além de ampliar nossas fantasias e experiências, sobretudo no que diz respeito aos comportamentos corporais, mentais e físicos [12]. Segundo Santana [13] as transformações ocorridas em virtude da aproximação das fronteiras entre Arte, Ciência e Tecnologia culminaram em novos formatos de manifestações artísticas. Paradoxalmente, enquanto o corpo tem sido o objeto de inquietações e investigações nos outros campos, tanto artísticos como científico e tecnológico, a dança, propriedade do corpo fez-se cega, muda e fechada para tais intrusões. No entanto, hoje em dia, vários trabalhos têm sido realizados no sentido de mesclar essas áreas do conhecimento: Arte e Tecnologia.

2.2 TECNOLOGIAS PARA MULTIMÍDIA

O termo multimídia designa a combinação, controlada por computador, de texto, gráficos, imagens, vídeo, áudio, animação, e qualquer outro meio pelo qual a informação possa ser representada, armazenada, transmitida e processada sob a forma digital, em que existe pelo menos um tipo de mídia estática (texto, gráficos, ou imagens) e um tipo de mídia dinâmica (vídeo, áudio, ou animação) [14].

Para perceber o relacionamento entre o número crescente de tecnologias multimídia, é útil classificar as tecnologias de acordo com um modelo de referência rigoroso, que permita caracterizar conjuntos de tecnologias afins, agrupando-as na mesma área tecnológica. Uma das formas possíveis para conceber tal modelo é situar e classificar as áreas tecnológicas em termos da proximidade que as respectivas tecnologias possuem em relação ao usuário. Por exemplo, ao nível mais baixo (mais longe do usuário) situam-se as tecnologias de base que suportam a representação digital da informação multimídia. Por outro lado, ao nível mais elevado (mais próximo do usuário) incluem-se as tecnologias que suportam a criação de conteúdos e de aplicações multimídia, e as próprias aplicações com as quais o usuário interage diretamente [15].

A Figura 1 mostra um modelo de classificação das áreas tecnológicas associadas à multimídia, identificando claramente os níveis a que as áreas tecnológicas se situam em relação ao usuário final.

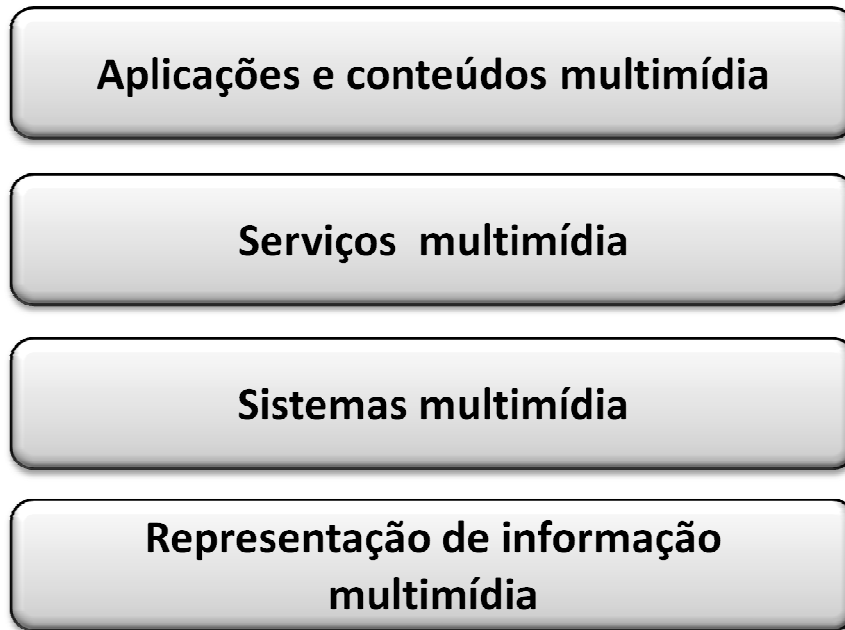


Figura 1 - Um modelo de classificação para as tecnologias multimídia [15].

Nas subseções seguintes detalharemos cada uma das áreas tecnológicas, bem como, as principais ferramentas e técnicas que foram investigadas no escopo desse trabalho.

2.2.1 Representação de Informação Multimídia

As tecnologias de representação de informação multimídia incluem as técnicas de digitalização de informação, os formatos de representação de cada mídia, inserindo as técnicas de compressão que lhes estão associadas, e os aspectos relacionados com o hardware necessário para a obtenção de uma plataforma multimídia, por exemplo, as placas de som, câmeras digitais e placas de vídeo.

As técnicas de compressão de vídeo englobam os padrões de compressão de vídeo, porém podemos classificá-los em duas categorias distintas: padrões de compressão para armazenamento e padrões de compressão para transmissão (*streaming*). Além disso, alguns métodos de compressão atendem a características específicas de um determinado ambiente (exemplo, vídeo para celulares). Alguns dos parâmetros para avaliação de uma determinada técnica são: carga de processamento (para compressão e descompressão), disponibilidade de memória (para armazenamento ou bufferização), resolução do vídeo (para reprodução em dispositivos compatíveis), etc. A harmonização desses parâmetros define a qualidade de uma técnica.

Alguns padrões de compressão/codificação importantes são descritos abaixo:

- MPEG-1 – utilizada nos VCDs (Video CD), qualidade comparada a de um vídeo armazenado em VHS;
- MPEG-2 – utilizada nos DVDs (*Digital Video Disc*) além de na maioria das transmissões de TV digital disponíveis;
- MPEG-4 (Parte 2) – permite o uso de diferentes *profiles*, estabelecendo valores diversos de taxas de compressão. É muito utilizado em transmissões na Internet, por possuir uma ótima qualidade mesmo a baixas taxas de transmissão, permitindo a visualização das imagens do vídeo não importando a capacidade do meio de transmissão (Internet de banda larga ou linha discada, por exemplo);
- THEORA – padrão de codificação de vídeo baseado no VP3 que foi cedido pela empresa *On2 Technologies*, semelhante ao MPEG-4/DivX. Usa o padrão encapsulamento para áudio, vídeo e metadados chamado OGG para transmissão na *Internet*. É completamente livre, aberto e não patentado.
- MPEG-4 (Parte 10) – (*Advanced Video Coding* ou H.264 ou MPEG-4 Part-10) Padrão de codificação de vídeo desenvolvido inicialmente pelo VCEG (sob a sigla H.26L) e, posteriormente, pelo JVT, sendo também conhecido como AVC ou MPEG-4, Parte 10. Juntamente com outros codecs de terceira geração, como o DivX e *Windows Media Video*, constitui o estado da arte em codificação de vídeo no início do século XXI; e
- WMV – (*Windows Media Video*) padrão da Microsoft, adaptável para vídeos desde acessíveis pela internet através de conexão discada até vídeos de alta resolução (HD, *High Definition*). Foi feito baseado no MPEG-4.

Por sua vez, as técnicas de compressão de áudio são utilizadas para atender às exigências intrínsecas à transmissão (largura de banda) ou armazenamento do som.

Alguns padrões de compressão de áudio importantes são descritos abaixo:

- AAC: *Advanced Audio Coding* (Codificação Avançada de Áudio) – Um dos formatos de compressão de áudio usados nos padrões MPEG-2 e MPEG-4;
- VORBIS – é um formato de compressão de áudio muito difundido na *Internet*. É comparável a outros formatos utilizados para armazenar e reproduzir música digital, como MP3, AAC e outros formatos de áudio

digital. O que o difere principalmente dos outros formatos é por ele ser completamente livre, aberto e não patenteado.

- AC-3: *Adaptive Transform Coder 3* (Codificador por Transformadas Adaptativo 3), também conhecido como A/52 – formato de fluxo de dados do padrão *Dolby Digital* de compressão de áudio com perdas, desenvolvido pelos Laboratórios *Dolby*; e
- MP3: MPEG-1, Layer 3 (MPEG-1, Camada 3) – Um dos formatos de compressão de áudio com perdas usado no padrão MPEG-1. Alcançou grande popularidade na codificação de música para distribuição pela Internet e para armazenamento em computadores e dispositivos pessoais de áudio.

Para a composição de um ambiente multimídia se faz necessário um determinado conjunto de hardware para uma plataforma multimídia, que podem ser computadores, placas de áudio, placas de vídeo, câmeras digitais, placas de captura etc.

São descritos alguns desses hardwares que podem ser utilizados:

- High Definition (HD) Camcorders: Canon HV20, Canon HV30, Sony HDRHC9, JVC GZ-HD7, GZ-HD6, Canon HG10, Sony HDR-SR12, Panasonic HDC-HS9, JVC Everio GZ-HD40, etc;
- Placas de vídeo: NVIDIA GeForce 9800GTX, ATI Radeon™ HD 4800 Series, GeForce GTX 295, etc e
- Placas de som: Creative Audigy SE 7.1 24bits, AuzenTech HDA X-Plosion, Encore 7.1 Surround, etc.

2.2.2 Sistemas Multimídia

Outro conceito importante é o conceito de sistemas multimídia que inclui técnicas associadas ao processamento, armazenamento, apresentação e transmissão de informação multimídia. Por exemplo, os dispositivos de armazenamento óptico, tais como o CD e o DVD, constituem os sistemas de armazenamento mais utilizados para aplicações multimídia cuja distribuição é feita de modo *offline*. Por outro lado, as bases de dados multimídia fornecem funcionalidades que permitem armazenar de um modo estruturado os vários tipos de mídia.

Para descrever um sistema multimídia é necessário considerar o Sistema Operacional (SO), as linguagens de programação e os serviços que se deseja oferecer.

Os sistemas operacionais mais usados e que têm suporte multimídia são:

- Windows: Windows 2000, Windows Vista, Windows XP, Windows Server 2008, Windows 7;
- Linux: Debian, Ubuntu, Fedora, etc e
- Mac OS: Mac OS X, 10.5 Leopard, Mac OS X Server, etc.

Hoje em dia, existem muitas linguagens de programação que oferecem suporte ao desenvolvimento de sistemas multimídia, onde destacamos:

- JAVA: Java Media Framework (JMF), Java 2D, Java 3D, Java Advanced Imaging, Java Binding for OpenGL, Java Image I/O, JVLIC, etc;
- C/C++: Bellagio OpenMAX IL Implementation, EiffelMedia, gavl, Gnonlin libavg, libiptcdata, libnemesi, wxMusik, wxWidgets, OpenCV etc e
- NCL: é uma linguagem de aplicação XML com facilidades para a especificação de aspectos de interatividade, sincronismo espaço-temporal entre objetos de mídia, adaptabilidade, suporte a múltiplos dispositivos e suporte à produção ao vivo de programas interativos não-lineares [16].

2.2.3 Serviços Multimídia

Outro fator importante que deve ser considerado são os serviços que se deseja oferecer. Os serviços multimídia tem se tornado cada vez mais diversificados e integrados as plataformas de distribuição de conteúdo existentes. Para tanto, é necessário considerar as técnicas que permitam implementações específicas de um dado sistema multimídia que fornece uma ou mais funções. Por exemplo, as técnicas de sincronização temporal das mídias são essenciais para a criação de aplicações multimídia, e ajustam-se nas funções fornecidas pelos sistemas operacionais sobre os quais as aplicações são instaladas. Outro exemplo é o vídeo sob demanda – este serviço é muito utilizado na TV Interativa (uma aplicação multimídia), e ajustam-se nas funções fornecidas por bases de dados multimídia e servidores de mídia (neste caso, servidores de vídeo).

Para evitar o “nivelamento por baixo”, ou seja, que a presença de clientes com pouco poder de processamento ou pouca banda de acesso acabe forçando a transmissão de vídeo de baixa qualidade, muitas vezes uma mesma transmissão é replicada com diferentes taxas de codificação ou é utilizada a chamada “codificação hierárquica”, um mecanismo que divide o vídeo em camadas, permitindo que uma mesma transmissão atinja pontos com diferentes taxas de transmissão, porém quanto maior a quantidade de informações que chega a um determinado cliente, maior a qualidade do vídeo.

2.2.3.1 Vídeo ao vivo

A transmissão de mídias (*Streaming media*) relaciona-se com o ato de se consumir mídia de alguma fonte. No caso de transmissão de vídeo ao vivo, essa fonte seria uma câmera (webcam ou câmera convencional ligada a uma placa de captura) que capta um evento e repassa para os clientes do serviço.

Formatos utilizados para transmissão ao vivo geralmente são desenvolvidos com base em alguns aspectos inerentes às transmissões. O fato de que um cliente pode acessar uma transmissão em qualquer ponto da mesma (ao contrário de um arquivo, onde o acesso é geralmente feito no início do mesmo) cria a necessidade de decodificação de segmentos. Para viabilizar a sincronização do fluxo (com seus elementos internos, como vídeo e áudio), mecanismos de transporte sinalizado são utilizados, quer seja na forma de protocolos de transmissão ou de sinalização na codificação do vídeo. Outros fatores importantes são relacionados à infraestrutura necessária para transmissão de vídeo ao vivo, o que guia um desenvolvimento que otimize os recursos de rede, processamento e memória, de forma a atingir um maior número de possíveis clientes.

2.2.3.2 Protocolos Multimídia

Os protocolos usados nas aplicações multimídias são responsáveis por empacotar as mídias, controlar as operações durante as sessões, otimizar o processo de transmissão das mídias, possibilitar interação e reservar recursos.

São descritos abaixo alguns desses protocolos:

- RTSP (*Real Time Streaming Protocol*) - é um protocolo de domínio público que permite a interação cliente-servidor entre a fonte do fluxo de mídia a taxa constante e o usuário. Essa interatividade vem da necessidade do usuário ter um maior controle sobre a reprodução da mídia [17];
- RTP (*Real Time Protocol*) - é um protocolo utilizado para o transporte de mídias contínuas de tempo real em uma conexão ponto a ponto, como áudio, vídeo ou dados de uma simulação. Pode ser usado não somente em uma comunicação ponto a ponto, mas também pode ser usada em uma transmissão multicast utilizando um endereço IP da faixa reservada para grupos multicast, porém há uma limitação, pois todos os roteadores envolvidos entre a origem e o destino devem suportar multicast [18];
- RTCP (*Real Time Control Protocol*) - é muito utilizado em paralelo ao RTP contribuindo para que a distribuição dos dados ocorra de maneira escalável ao ponto de permitir grandes transmissões multicast e também provendo um certo controle e identificação dos participantes da comunicação [18] e
- SDP (*Session Description Protocol*) - é um protocolo que descreve sessões multimídias e não possui um mecanismo próprio de transporte (utiliza protocolos como SIP, RTSP, etc), ele descreve algumas informações importantes da mídias, como tipo de mídia, protocolo de transporte usando, formato de mídia, endereço de destino, porta de recepção, etc [19].

2.2.4 Conteúdo Multimídia

Deve-se considerar o conteúdo multimídia que deverá ser transmitido, o que inclui técnicas que permitem produzir, isto é, conceber e desenvolver, aplicações multimídia, incluindo as técnicas para o desenvolvimento de um projeto multimídia, os paradigmas de autoria de aplicações multimídia e as técnicas específicas para a autoria de cada mídia. Este grupo inclui ainda a caracterização de aplicações multimídia de acordo com o seu objetivo e área de aplicação e as técnicas de criação de interfaces interativas [15].

A televisão por meio do protocolo internet ou *Internet Protocol Television* (IPTV) – é uma tecnologia para entrega de serviços de televisão digital e outras mídias por meio de conexão banda larga. A IPTV é uma tecnologia abrangente com amplas

funcionalidades, da aquisição, a codificação e decodificação, ao controle de acesso e gerenciamento de conteúdo de vídeo, entrega de TV digital, filmes sob demanda, assistir programação previamente armazenada e serviços interativos multimídia [20].

Além de beneficiar as empresas de telecomunicações, esta tecnologia permite maior interação dos usuários com a TV, trazendo a escolha de conteúdo para suas mãos, além de poder significar uma redução no valor do pacote fechado, em uma única conta.

O IPTV opera, portanto, de forma diferente dos sistemas tradicionais de televisão (cabo, satélite e terrestre), dado que só os programas selecionados e os conteúdos "*on Demand*" são distribuídos ao consumidor. O IPTV dispõe sempre de duas vias de comunicação, oferecendo uma verdadeira interatividade entre o usuário e o sistema.

Com IPTV é possível, por exemplo, atender a uma chamada telefônica na televisão e ver, em uma janela *Picture-in-Picture*, a imagem da pessoa com quem se está falando. É possível, também, comprar imediatamente um produto que esteja sendo anunciado, ou que seja exibido em uma novela [20].

No IPTV o conteúdo é enviado apenas em streaming, porém com garantia de qualidade na entrega. O receptor é um aparelho set-top box (Terminal de acesso) conectado à televisão (semelhante ao aparelho da televisão a cabo).

O IPTV permite entrega de áudio e vídeo com altas qualidades, e depende de uma conexão Banda Larga (normalmente fornecida junto com o serviço como parte integrante) de no mínimo 4 Mbps. A banda destinada ao IPTV não interfere na banda de internet.

Já a WebTV, ou televisão na Internet, além do conteúdo ser visto principalmente no computador, pode-se montar uma programação para ser enviada por download. Entretanto, se o sistema escolhido for streaming, não há garantia de qualidade, podendo haver pausas ou interrupções no envio do conteúdo (por se tratar da rede pública). O dispositivo receptor usualmente é o computador [21].

Outro fator importante de diferenciação: a rede de distribuição do conteúdo (canais de TV, vídeo sob demanda, jogos, mensagens, etc) do IPTV é fechada, se assemelhando a uma intranet corporativa, contra a WebTV que é transmitida via internet, uma rede de acesso livre.

3 TRABALHOS CORRELATOS

3.1 NA INSTITUIÇÃO

Alguns projetos foram essenciais para fomentação do arcabouço para o desenvolvimento da Arthron.

O GT de Vídeo Digital (GT VD, fases 1 e 2) teve por objetivo implantar uma infraestrutura baseada na RNP que ofereça suporte a aplicações envolvendo manipulação de vídeo digital. Coube ao GT VD incentivar e fornecer condições para criação, armazenamento e transmissão de conteúdo na forma de vídeo digital no país. O trabalho do grupo, em sua primeira fase, baseou-se em duas iniciativas principais: Portal de Vídeo Digital e Rede de Vídeo Digital. A iniciativa Portal de Vídeo Digital visou organizar um site, mantido pelo GT VD, com informações para os interessados em participar das atividades do GT. O site inclui uma agenda com experimentos programados, indicadores para ferramentas e protocolos que permitam aos interessados participar dos experimentos de vídeo digital, documentos sobre o assunto (padrões, artigos, relatórios técnicos etc.) e informações sobre normas para participação, equipe e atividades do GT. O objetivo da iniciativa Rede Vídeo Digital (RVD) foi disponibilizar uma infraestrutura de rede, servidores e equipamentos para dar suporte a experimentos que envolvam a captura, recuperação e transmissão de vídeo sob demanda. Com resultado desta iniciativa foram desenvolvidos e instalados na RNP servidores de vídeo sob demanda, com suporte para transmissão considerando capacidades e restrições dos clientes, servidores e equipamentos de comunicação utilizados. Na sua segunda fase o GT VD utilizou-se da mesma abordagem dada aos serviços de vídeo sob demanda da primeira fase para agregar à Rede Vídeo Digital a funcionalidade de transmissão de vídeo ao vivo. Além disso, o serviço de vídeo sob demanda tornou-se mais flexível, permitindo que o acervo de vídeos fosse armazenado em um serviço de armazenamento distribuído, no caso, o da rede *Internet Backplane Protocol (IBP)* da *University of Tennessee*. Assim como na primeira fase, foram desenvolvidos novos servidores (sob demanda e ao vivo), e instalados na rede RNP. Desta forma, o portal de vídeo digital passou a contar com a programação ao vivo, relativa à transmissão de eventos, tais

como o Simpósio Brasileiro de Redes de Computadores (SBRC) de 2004 em Gramado, além da retransmissão de canais como o Canal Saúde da Fundação Oswaldo Cruz [3, 4].

O GT TV surgiu como desdobramento dos projetos I2TV (Infraestrutura Internet2 para Desenvolvimento e Teste de Programas e Ferramentas para TV Interativa), HiTV (Desenvolvimento de Software e Hardware para Sistemas de Televisão Digital de Alta Definição) e dos GTs de Vídeo Digital da RNP fases 1 e 2. O projeto desenvolveu uma plataforma que facilita o acesso de usuários de características heterogêneas ao conteúdo de canais de TV distribuídos através Internet, utilizando uma infraestrutura IPTV [5, 6].

O GigaVR teve como objetivo o desenvolvimento e demonstrações de uma plataforma, ou seja um acervo de software e hardware, voltada para a criação de aplicações da realidade virtual imersiva sob redes de altíssima velocidade. Nele foi apresentada a teleoperação de um robô associada à transmissão de vários fluxos de vídeos sincronizados com alta qualidade e em tempo real, com a exibição desses vídeos em uma Caverna Digital [7].

O GTMDA tem como foco principal oferecer formas mais avançadas para Interação Humano Computador (IHC), as quais permitam o entrelaçamento de agentes humanos e sintéticos em espaços midiáticos compartilhados e distribuídos, em tempo real, através de redes de computadores de alta velocidade e com grande volume de informação. Essa convergência de Arte e Tecnologia é uma área que emerge da tensão criada pela confluência entre a Arte e as novas mídias digitais, investigando como acontece essa nova forma de pensar e fazer Arte com suporte tecnológico [22, 23, 24]. A idéia central é a construção de uma ferramenta de gerenciamento que concentre ao máximo o controle de todos os dispositivos de hardware e software envolvidos para execução de evento de cunho artístico-tecnológico.

3.2 ESPETÁCULOS

Alguns trabalhos da literatura [25, 26, 27, 28] ilustram bem esse cenário de convergência entre a tecnologia e arte, que juntamente com as experiências realizadas [8, 9], impulsionaram o surgimento de ferramentas como a Arthron e serão apresentados nas subseções a seguir.

3.2.1 Uma Performance de Dança Colaborativa

Nesse trabalho é descrito uma experiência de tecnologia e arte denominada de “*Information Overload*” [25] que mostra a dependência que a tecnologia causa nas pessoas, enfatizando que a tecnologia está atualmente se tornando mais necessária na vida pessoal dos indivíduos. O ápice da peça é quando é mostrada a “tela azul da morte” (tela de erro azul do windows) e os dançarinos que representam as pessoas que são altamente dependentes da tecnologia caem no chão do palco dando a idéia de que já não conseguem mais viver sem os aparatos tecnológicos.

Percebeu-se também que para que fosse possível a realização da peça foi necessária a presença de um grande número de pessoas das mais variadas áreas do conhecimento que incluíam estudantes da Drexel University dos cursos de artes performáticas, mídia digital, engenharia elétrica, engenharia da computação e indústria da música. Também incluíam profissionais para auxiliar no projeto de trajes, no projeto de iluminação e também para assistência técnica no teatro.

Diversas ferramentas foram utilizadas para que fosse obtido o resultado esperado na peça. O fundo do cenário que era composto por nuvens foi obtido na biblioteca de imagens da Adobe (*Adobe Image Library*). O Adobe Flash foi usado na geração da silhueta dos dançarinos. O texto que se transformava na metrópole foi animada e texturizada usando o Autodesk 3D Studio Max. A DICE – *Digital Instrument for Computed Expression* que é um programa criado por John Henry Thompson e que utiliza de 22 diferentes filtros para transformar vídeos em tempo real também foi usado. Na animação final foi utilizado o 3D Studio Max para modelar e animar as peças de fábricas que eram seguradas pelos dançarinos.

A infraestrutura da peça inclui bailarinos e equipamentos. Participaram dessa manifestação artística quatro dançarinos. Quanto aos equipamentos foram utilizados:

- Projetores – um para o fundo (estático) e outro para projetar o vídeo que se encontra em movimento e
- Duas câmeras firewire – uma para obter as informações dos LED’s presentes em um dos dançarinos, e a outra para obter informações sobre o guarda-chuva através de sua cor como pode ser observado na Figura 2.

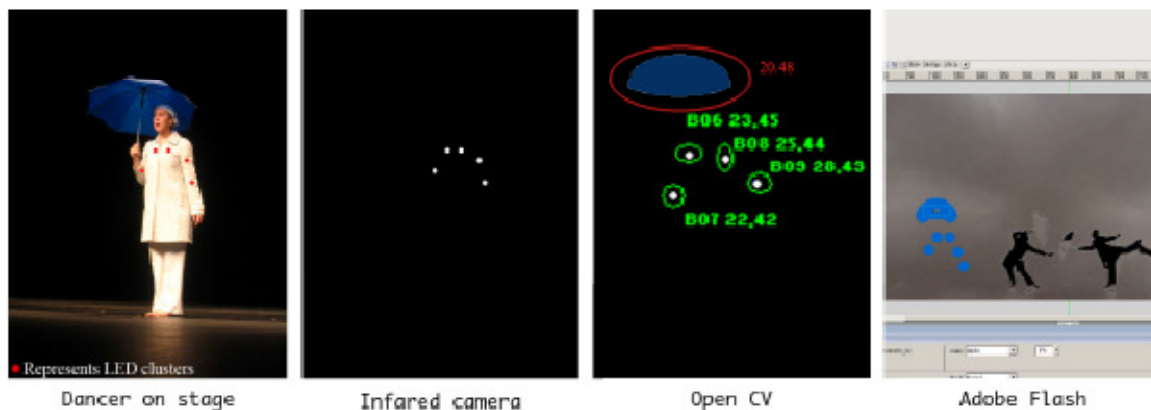


Figura 2 - Destaques da performance de dança colaborativa [25].

3.2.2 Mesclando Realidades – Um Performance Artística com Colaboração

O objetivo principal do trabalho [26] é desenvolver e facilitar o relacionamento entre dançarinos nos diversos estágios da apresentação, bem como criar um som artístico e tecnologicamente sem falhas para a apresentação, ajudar na colaboração entre os diversos membros da equipe, explorar a natureza da tecnologia, entre outros.

Para tanto, foi realizado uma apresentação teatral experimental feita por professores e estudantes da 'Purdue University' com características de captura de movimento ao vivo, computação gráfica em tempo real, e projeção multi-imagem. Essa apresentação enfatizou a interação entre os dançarinos que ocupam ambos os espaços, virtuais e reais.

Para isso foram utilizadas diversas ferramentas. O personagem virtual foi modelado no Discreet's 3dsmax 5.1 e exportado usando o formato de arquivo FBX para o Kayadara's FilmBox Online 3.4 que era responsável pela projeção do vídeo. Para efetuar a captura de movimento ao vivo foi utilizado o sistema Gypsy 3.0 conectado a um notebook Dell Precision M50 que fez a transmissão do fluxo através de uma rede local de 100Mbps. Todos os fluxos de vídeo foram enviados para o equipamento Grass Valley 3000 que foi o responsável pela comutação desses vídeos nos projetores. Três telas foram usadas para projetar o dançarino virtual no palco.

O dançarino virtual encontrou diversas restrições de movimentos causados pelo equipamento de captura de movimento ao vivo, para contornar esse problema foi utilizada a teoria do esforço/forma de Laban, que descreve aspectos qualitativos de movimentos para a comunicação de emoções como briga, apreensão, alegria e

segurança. Era permitido ao dançarino virtual atravessar o cenário e para isso foi feita a coordenação entre os três computadores, o computador, os projetores e o dançarino que estava no sistema.

A peça foi estruturada utilizando-se do encadeamento de estágios como pode ser observado na Figura 3.

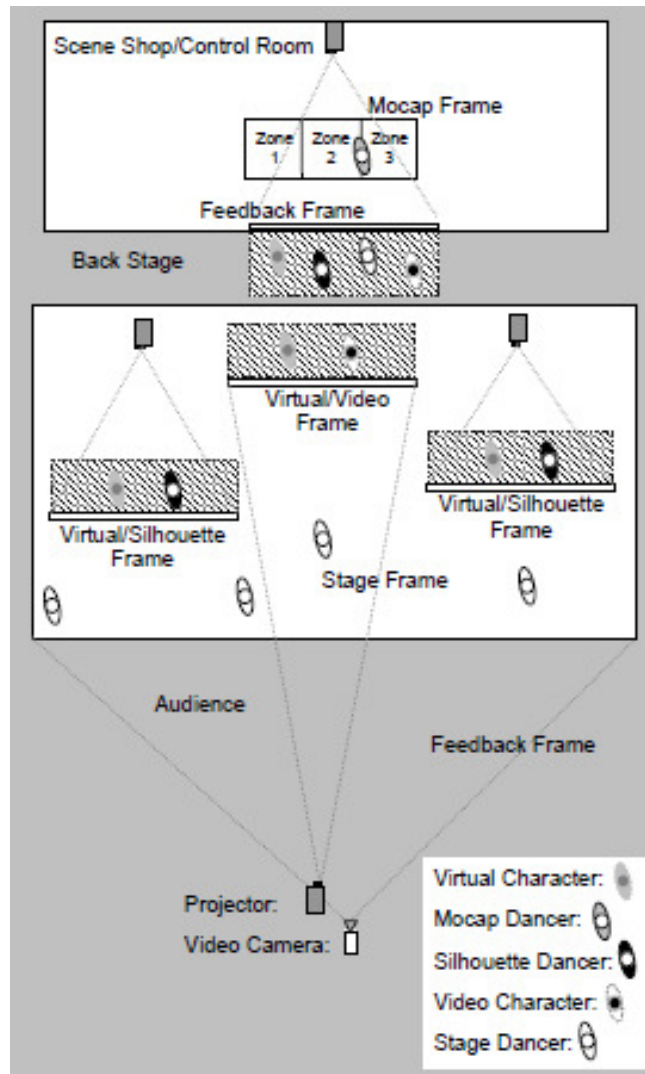


Figura 3 - Projeto da peça com definição de estágios, dançarinos e agentes virtuais [26].

Os estágios foram classificados levando em consideração a disposição entre os dançarinos no palco como se segue:

- O estágio de palco – Era o estágio principal. Continha todos os elementos do palco, incluindo cinco dançarinos ao vivo e três telas de projeção. As telas da esquerda e da direita eram posicionadas no chão e a tela do centro era

posicionada no meio de forma flutuante acima da cabeça dos dançarinos. Os dançarinos interagem um com os outros e também com o conteúdo ao vivo projetado nas telas.

- O estágio de sombra – Em vários pontos da peça um ou mais dançarino se posicionava entre a tela de projeção e o projetor, de forma que a luz do projetor cria uma bem definida camada de sombra em movimento. Nesse estágio os dançarinos que estão no palco podem interagir com o dançarino virtual. O problema nesse estágio é que o dançarino que está atrás da tela não consegue ver os dançarinos que estão na frente, porém ele consegue ver o dançarino virtual se este estiver sendo projetado na mesma tela.
- O estágio de vídeo – Esse estágio ocupa somente a tela central e só mostra vídeos pré-gravados. Pode-se simular a interação com o vídeo fingindo que ele é ao vivo.
- O estágio virtual – É o elemento central da produção, e é ocupado por um dançarino virtual que se movimenta em tempo real, fora do palco, em um sistema de captura de movimento. O dançarino virtual é projetado para ser capaz de atravessar todo o cenário através da mudança de onde o seu vídeo é projetado. Por ser feito em tempo real e por ser possível a comunicação entre os dançarinos reais e virtuais a interação entre eles se torna possível.
- O estágio Mocap – Esse estágio se trata da associação entre o posicionamento do dançarino virtual em relação à tela de apresentação que ele será projetado. O piso do local onde o dançarino virtual faz sua performance é mapeado de forma a coincidir com a sua tela de projeção apropriada.
- O estágio de FeedBack – Tratasse de uma projeção do palco inteiro feita dentro do espaço onde o dançarino virtual está fazendo a coreografia para que ele possa ter a sensação de estar localizado no palco. Para que possa experimentar dessa sensação o dançarino virtual deverá se adaptar ao novo ponto de vista do palco e assim conseguir interagir convincentemente com os dançarinos do estágio do palco.

Ao final da peça o público teve uma resposta positiva da interação entre o personagem virtual e os dançarinos ao vivo no palco.

3.2.3 Uma Performance de Dança Interativa Baseada no Movimento

Sabe-se que eventos de dança baseados em movimentos interativos tem recentemente atraído um grande interesse dos espetáculos de arte. Assim, embora utilizando tecnologia de captura de movimento, o objetivo deste trabalho foi projetar o mecanismo necessário de análise e observação de movimentos em tempo real, bem como os sistemas de comunicação para a realização de uma performance de dança multimídia interativa baseada em movimentos. Este mecanismo mede a correlação dos movimentos de dança entre três pessoas que vestem um conjunto semelhante de marcadores retro-reflexivos⁴.

O trabalho apresentado em [27] propõe um método de análise de movimentos que se concentra nas relações de tempo e espaço entre dançarinos de um grupo. As informações sobre os movimentos foram gravadas em tempo real e calculadas por um sistema de captura de movimentos constituído por 16 câmeras em conjunção com um mecanismo que analisa os movimentos. Este método permitiu avaliar características de baixo nível dos movimentos e, assim, determinar como os bailarinos estavam se movendo com relação a si e ao espaço do evento.

O sistema também se utiliza de marcadores retro-reflexivos que são acoplados nos bailarinos para capturar movimentos através do espaço e ao longo do tempo. Cada câmera é cercada por um emissor de luz, em anel, que está definido por uma frequência de luz específica (neste experimento utilizou-se uma frequência próxima do infravermelho ou infravermelho). Esta luz é refletida de volta para as câmeras pelos marcadores criando uma imagem 2D das posições dos marcadores. Quando duas ou mais câmeras podem “ver” o mesmo marcador, o sistema é capaz de colocar ou “reconstituir” os marcadores em coordenadas 3D. Outra característica do sistema é que a informação capturada pode ser enviada em tempo real sobre TCP/IP para softwares clientes conectados. Assim, podem-se utilizar os dados para animação, áudio interativo

⁴ O sensor ótico reflexivo, também conhecido como retro-reflexivo, assim como o difuso, tem o transmissor e o receptor montados em uma única unidade. O feixe de luz chega ao receptor após a incidência em um espelho e o acionamento da saída ocorre quando o objeto interrompe o feixe.

e análise de movimentos. O resultado final da peça é uma atmosfera integradora de dançarinos e imagens sintéticas como pode ser observado na Figura 4.

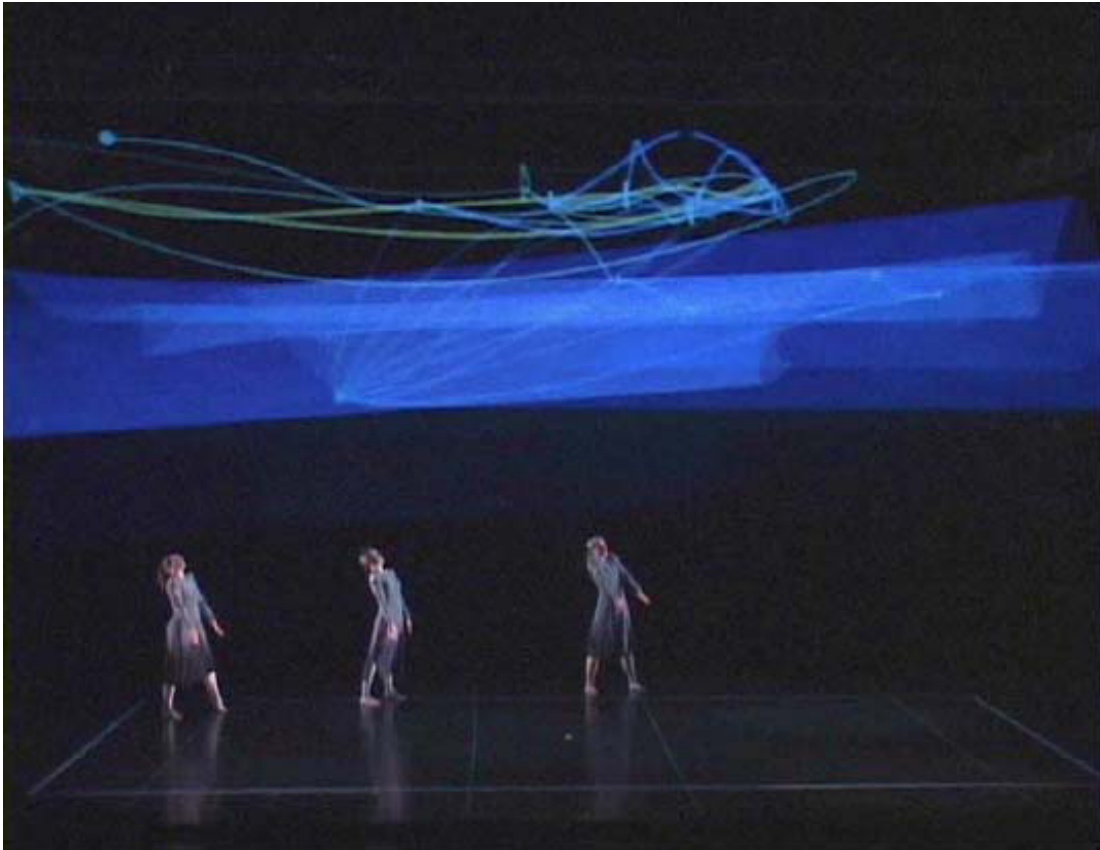


Figura 4 - Cenas da peça onde dançarinas interagem com as projeções [27].

Neste trabalho, desafios foram resolvidos com soluções criativas e de metodologias científicas na criação de uma peça de dança interativa, utilizando tecnologias de captura de movimento. Isso foi possível devido à utilização de um mecanismo de análise de movimentos que acompanhou a proximidade, a tradução e as correlações entre os movimentos corporais em tempo real. Trabalhos desta natureza trazem impactos em outros domínios da investigação, pois ele pode ser usado para entender melhor o comportamento humano e de movimentos sociais, culturais, criativos e de contexto científicos.

3.2.4 Um Estudo de Caso sobre Colaboração em Arte e Tecnologia

Um profundo estudo de caso é um método comumente usado para estudar complexos fenômenos sociais. Poucos estudos têm sido conduzidos sobre o contexto

que envolve a colaboração na área de Arte e Tecnologia. Todos esses estudos têm alguns “insights” sobre a colaboração e têm demonstrado que o estudo de caso em profundidade é um método adequado para estudar a colaboração em aplicações desse tipo em um nível mais detalhado. Porém, a maioria dos estudos está baseada em notas de campo e dados de entrevista que provavelmente dão um quadro incompleto do que acontece de fato na natureza da colaboração. Assim, o trabalho [28] propõe um estudo que visa ampliar os conhecimentos na colaboração em aplicações de Arte e Tecnologia provendo dados de observação mais contínuos que usam instalações de vídeo bem como uma análise esquemática melhorada usando um software de análise de comportamento, o *INTERACT*.

Este trabalho tem como objetivo fazer um estudo de caso detalhado de um processo colaborativo de um projeto criativo que envolve arte e tecnologia. Esse estudo é feito através de um projeto denominado “*GEO Narrative Landscapes*”, utilizando um método moderno para a análise da interação entre os artistas e tecnólogos.

O projeto de GEO foi constituído por um time de pesquisadores multidisciplinar que incluía um artista, um designer gráfico, dois tecnólogos e um observador. Todos os participantes foram co-localizados na mesma cidade. O artista foi o responsável por gerar o artefato digital e dirigir seu desenvolvimento. Os dois tecnólogos estiveram a cargo de diferentes partes das funções técnicas do projeto: um trabalhou com o ambiente de programação MAX/MSP⁵ e o outro usou a linguagem de programação Java. O designer trabalhou principalmente no design de imagens e o observador, acompanhou o processo de colaboração no local.

O modo de colaboração básico para este projeto foi através de reuniões de grupo. A duração das reuniões do grupo foi de cerca de trinta minutos à uma hora baseada na demanda e as reuniões semanais. Além de reuniões, o tempo foi gasto em trabalho individual ou pessoa-a-pessoa da comunicação por e-mail ou telefone. Durante o período do estudo, dezoito reuniões de grupo foram acompanhadas e registradas. O projeto estava na localização *Creativity and Cognition Studios (CCS)*, da *University of*

⁵ É uma linguagem gráfica de programação para música/som. Os programas são criados utilizando objetos gráficos que proporcionam um processo mais intuitivo de desenvolvimento; não recorrendo a instruções (comandos em texto), como usual na programação. Podem-se criar programas para compor, improvisar, tocar ao vivo, etc.

Technology, Sydney, onde um estúdio construído propositadamente para investigação da arte interativa está instalado. O estúdio principal tem várias funções que incluem computação e equipamentos que permitam a interação ambiente de uma equipe para desenvolver e avaliar protótipo de arte interativa.

Por fim, o trabalho aborda um estudo de como artistas e tecnólogos se comunicam entre si durante o processo de desenvolver um artefato digital. Foi apresentada uma aproximação moderna para analisar os padrões de comunicação de trabalho interdisciplinar. A contribuição deste trabalho é que apóia a compreensão da criatividade colaborativa que pode ser aplicada, por exemplo, para facilitar os modos de comunicação que estão muito significativamente em fases diferentes da Arthron.

3.2.5 Experiências Realizadas

Nossas experiências na realização desse tipo de performance são ilustradas por dois espetáculos realizados no Brasil e com a colaboração da RNP: *Versus* e o (In)Toque.

3.2.5.1 Versus

O espetáculo “*Versus*”, Figura 5, consistiu numa experiência midiática distribuída, onde artistas localizados em três pontos distintos, João Pessoa, Salvador e Brasília interagiram através de uma sessão de vídeo em alta definição (High Definition - HD) e em definição padrão (Standard Definition - SD). Cada fluxo era projetado sobre telas semitransparentes localizadas em um palco principal em Brasília durante a 3ª Conferência Nacional de Ciência, Tecnologia e Inovação (CNCTI) [8], no lançamento do anel *Gigabit Ethernet* da RNP, hoje batizado de rede Ipê (RNP). Bailarinos localizados em Salvador e em Brasília interagiam, em tempo real, através de fluxos de vídeo em alta definição, criando a sensação de estarem em um único lugar. Em paralelo, músicos, localizados em João Pessoa, geravam áudio a partir de laptops em quatro esquemas musicais, onde o som e um vídeo em definição padrão eram transmitidos simultaneamente para Salvador e para o palco principal do evento em Brasília. Todos esses fluxos de mídia (áudio e vídeo) eram decodificados e exibidos no palco principal em Brasília auxiliado por um aparato tecnológico formado por câmeras, placas de captura, placas de áudio e vídeo, projetores, computadores, softwares, etc [29].



Figura 5 - Fotos do espetáculo Versus.

3.2.5.2 (IN)Toque

O espetáculo “(In)TOQue” conectou dois bailarinos no Rio de Janeiro, uma bailarina em Salvador, um Disc Jôquei (DJ) em São Paulo e um robô em Natal, seguindo a mesma lógica de configuração do Versus. A imagem da bailarina de Salvador foi transmitida em vídeo de alta definição. Através do telão montado no palco do evento no Rio de Janeiro foi possível observar o entrelaçamento entre os corpos sintéticos e humanos. O diferencial desse trabalho foi à adição de um agente robótico como agente performático da composição cênica. O cenário de transmissão do espetáculo é representado na Figura 6 [30].

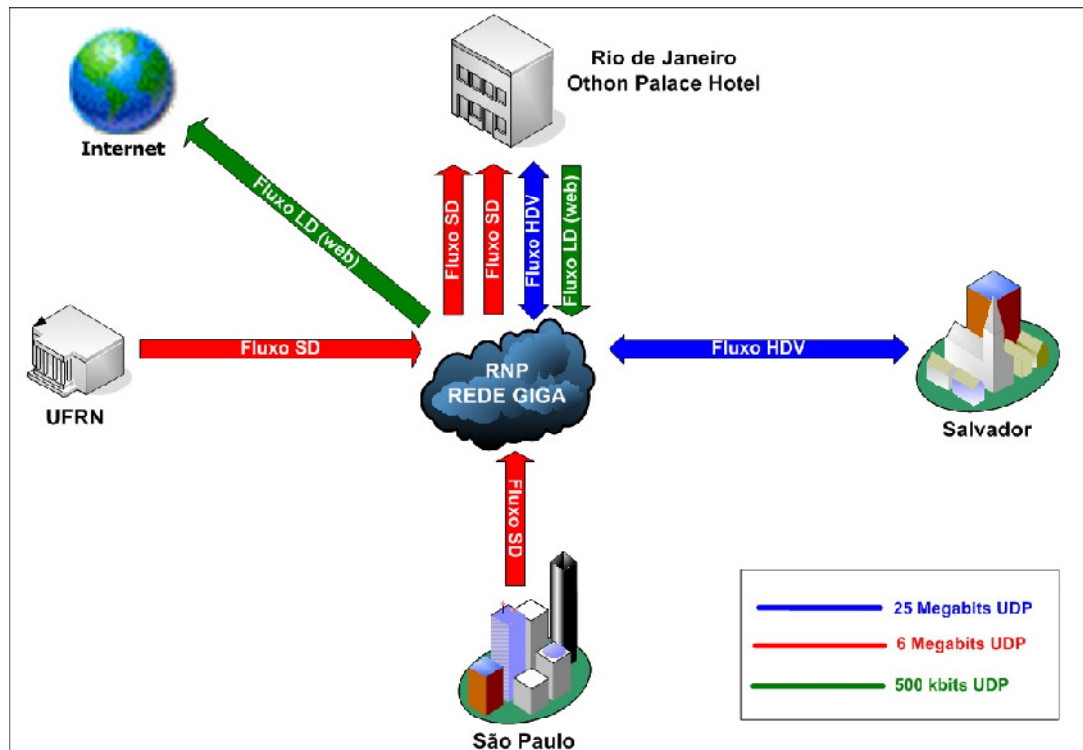


Figura 6 - Especificação do cenário de transmissão do espetáculo (In)TOQue. Observe que os fluxos e os pontos de distribuição são todos especificados nessa figura.

3.3 FERRAMENTAS EXISTENTES

Alguns trabalhos já citados [25, 26, 27, 28] fizeram uso de ferramentas e ambientes multimídia (software e hardware) para composição da parte tecnológica, mesclando-as com a parte artística, formando assim o espetáculo final. Algumas dessas ferramentas e ambientes se aproximam do que hoje é a Arthron e de certa forma estimularam o desenvolvimento de funcionalidades da mesma.

A DICE [31] é uma ferramenta que processa vídeo em tempo real, através de filtros. Esses filtros aplicados aos vídeos nada mais são do que a manipulação de pixels que combinados matematicamente pela ferramenta criam novas imagens a partir de um fluxo de vídeo em tempo real. Essa ferramenta foi desenvolvida para plataforma proprietária Macintosh da Apple e limita-se a inserção de filtros em vídeos em tempo real capturados de câmeras localmente acopladas ao computador, não fazendo uso de algum sistema de transmissão da pilha de protocolos TCP/IP.

O Grass Valley 3000 [32] é um equipamento que tem por principal finalidade comutar várias fontes de vídeo em tempo real ou pré-gravadas. Essas fontes de entrada podem ser via cabo composto (analógica) ou digital (serial ou paralela). Com esse equipamento pode-se ter um total de 64 entradas de fontes de vídeo distintas e comutá-las em qualquer saída através do acionamento de botões e alavancas. Apesar de ser um equipamento altamente profissional, ele limita-se a manipulação de vídeos localmente. Outro entrave é o custo na obtenção ou aluguel de tal aparato tecnológico.

SuperCollider (SC) [33] é um ambiente e linguagem de programação para síntese de áudio e composição algorítmica em tempo real. A partir da versão 3 o ambiente é dividido em servidor, ou *scsynth*, e cliente, ou *sclang*, que se comunicam via *sockets* utilizando a *Open Sound Control* (OSC)⁶ [34]. O servidor SC suporta simples plugins feitos na linguagem de programação C, o que torna fácil o desenvolvimento de algoritmos de manipulação de sons eficientes e todo controle externo no servidor ocorre via OSC. No lado servidor o processo de geração de som se dá através de um

⁶ É um protocolo para comunicação entre computadores, sintetizadores de som e outros dispositivos multimídia que são otimizados para tecnologia das redes modernas. Leva os benefícios da tecnologia das redes para o mundo dos instrumentos musicais eletrônicos.

executável otimizado em linha de comando chamado *scsynth*, que na maioria dos casos é controlado pelo ambiente do SuperCollider, mas também pode ser independente. Neste ambiente observa-se a inserção e manipulação de sons na rede em tempo real usando o protocolo TCP/IP. Também é destacado o caráter da licença ser *GNU Generic Public License* (GNU GPL), ou simplesmente software livre, o que a torna um ambiente propício para ser inserido em outros projetos de software livre, que necessitem a manipulação de áudio em tempo real, como a Arthron.

O INTERACT [35] é um software usado por muitos pesquisadores que necessitam coletar dados em estudos observacionais de ambientes multimídia. Ele permite que o usuário interaja com seu material de áudio/vídeo, analisando-o através do pressionamento de algumas teclas do computador ou usando o mouse. Através dele é possível manipular partes do vídeo, permitindo entrar em detalhes reais da cena, integrar qualquer tipo de dado externo, como informações fisiológicas ou dados do disco, oferecendo uma ampla gama de possibilidades de visualização e análise, tais como estatísticas, análise de confiabilidade (Cohen's Kappa⁷), análise seqüencial num intervalo de tempo, etc. Na Figura 7 observamos as possíveis fontes de entrada para o software INTERACT.

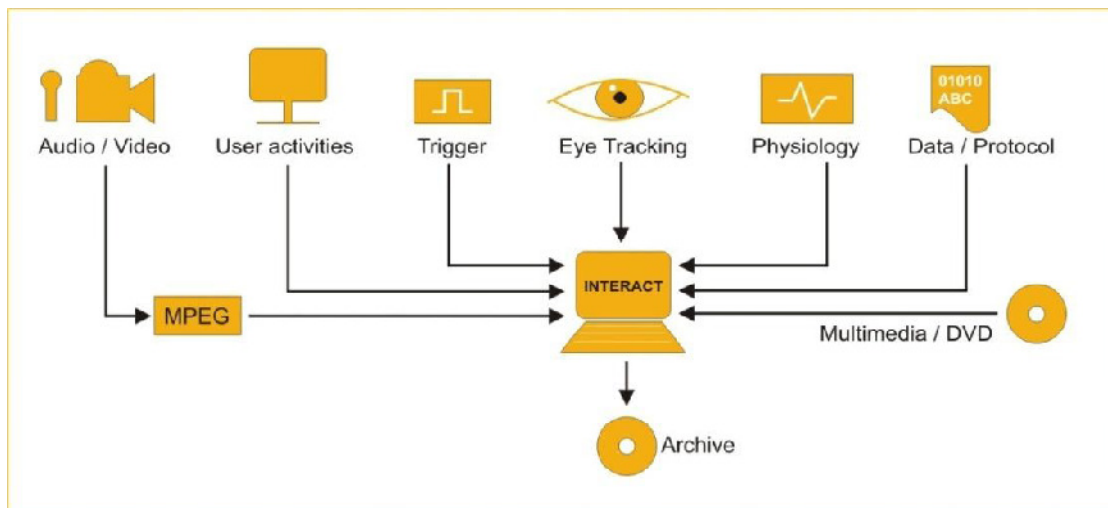


Figura 7 - Dados de entrada suportados pelo INTERACT [32].

⁷ Coeficiente Cohen's Kappa é um coeficiente de concordância que corrige o erro devido ao acaso e é usado para determinação da variação intra e inter-observador, sendo utilizado quando dois observadores classificam separadamente uma amostra de objetos empregando a mesma escala de categoria.

O INTERACT é um software avançado para coleta e geração de estatísticas para uma gama de cenários, tais como espetáculos artístico-tecnológicos como vimos no trabalho [28], porém ele é um software proprietário e como tal impõe custo financeiro para o seu uso e não pode ser integrado com outras plataformas de software que não sejam a do fabricante. O interessante deste software é que ele leva em consideração várias informações externas para gerar dados estatísticos. Assim ele aproxima-se bem de como seria um cenário real para a elaboração e execução de um determinado projeto, sendo este artístico, tecnológico ou meramente experimental, nas diversas áreas do saber. Através dele foi percebida a necessidade de formas de monitoramento e medição dentro da Arthron, para que fossem geradas estatísticas de uma determinada performance artístico-tecnológica, absorvendo assim a experiência a partir das informações coletadas para elaboração e execução de eventos futuros.

O SLTV [36] é uma ferramenta que captura conteúdo de áudio e vídeo e envia estas mídias para servidores *Icecast*⁸. Ele possui uma interface gráfica que apresenta o vídeo transmitido, permite a aplicação de efeitos dinâmicos e a configuração dos parâmetros da transmissão. No uso da TV Software Livre, foi identificada a necessidade de um software que reúna as tarefas de captura de áudio e vídeo, transcodificação de formatos e transmissão de vídeo (stream) para servidores *Icecast 2*. Há diferentes métodos de transmissão, utilizando e combinando programas que realizam cada uma das tarefas supracitadas. Este programa *publisher* a ser desenvolvido é uma ferramenta que se soma ao método básico de transmissão multimídia através da internet na plataforma Linux. Assim, neste cenário, está se desenvolvendo o SLTV como o programa principal para o setup de transmissão da TVSL (TV Software Livre), permitindo a captura e codificação em tempo real de diferentes entradas de vídeo e áudio, de dispositivos externos de captura como câmeras firewire, webcams USB e placas PCR e DVB (*Digital Video Broadcasting*).

⁸ Icecast é um software de transmissão de dados via rede em tempo real (streaming). Executa a função de servidor por ser o responsável por receber dados de uma fonte e distribuir para outros receptores (broadcast).

O SLTV é desenvolvido em *python*, usando as tecnologias Glade, GTK e Gstreamer⁹. A sua arquitetura segue uma estrutura modular que facilita o desenvolvimento de extensões, pois os diferentes módulos independentes poderão ser desenvolvidos de forma independente. Essa arquitetura facilita a colaboração e a utilização de apenas módulos específicos de acordo com a necessidade.

- *Capture* - Módulo responsável por identificar os dispositivos de captura e criar as entradas de fluxo de dados multimídia.
- *Encoder* - Módulo com a funcionalidade de transcodificação de mídias entre diferentes formatos.
- *Control* - Interface gráfica do programa, utilizada para controle dos processos de captura, codificação, visualização e transmissão de vídeos. Incluindo a monitoração da transmissão.
- *Stream* - Módulo que realiza a transmissão de vídeo Ogg Theora para um servidor *Icecast 2* da internet. Neste módulo é realizada a análise do fluxo de transmissão, apresentando um relatório da qualidade da conexão e recomendação de configurações de vídeo (*bitrate*, resolução, etc.) que podem otimizar a transmissão.
- *Upload* - Módulo responsável por realizar o envio de arquivos de vídeo para servidores da internet.
- Interface GUI - Módulo que o usuário utiliza diretamente e se comunica com o core do sistema para configuração das funcionalidades. Funciona desacoplado do core, permitindo que sejam desenvolvidas interfaces com diferentes níveis de complexidade, seja uma tela "wizard" ou um editor complexo com suporte a múltiplos *pipelines*.

O fluxo de áudio e vídeo originado no módulo *Capture*, ou proveniente de um arquivo do disco é direcionado para processamento no *Encoder*. Depois de codificado em Ogg theora/vorbis ele é opcionalmente apresentado na tela de *preview* do módulo Controle. Através do módulo de *Streamer*, o fluxo de vídeo e áudio é direcionado para um servidor de web TV ou rádio da internet. O *Streamer* deve suportar os protocolos de

⁹ GStreamer é um framework para criação de aplicações que lidam com streaming de mídia (áudio e vídeo).

stream de vídeo *Icecast 2* (http) e *Red5* (rtmp e rtmpt). O módulo *Uploader* se comunica com serviços remotos de armazenamento de vídeos através de protocolos de transmissão de arquivos ou *Webservices*.

Para o projeto SLTV foi escolhido o framework multimídia *Gstreamer* por possuir as seguintes funcionalidades fundamentais:

- Multiplataforma: Linux, Windows e Mac OS.
- Captura de dados brutos DV (*Digital Video*) de câmeras firewire.
- Captura de vídeo de webcams USB.
- Captura de áudio da placa de som.
- Codificador de diferentes formatos de vídeo e áudio.
- Decodificador de diferentes formatos, com saída de vídeo e áudio em interface gráfica.
- Envio de fluxo de vídeo e áudio para servidores *Icecast2*.

A interface gráfica (GUI) do SLTV é independente dos módulos de processamento de mídia, permitindo a implementação de diferentes modelos de interface. Inicialmente foi criada uma interface simplificada, que permita a configuração dos parâmetros da captura e codificação do fluxo de áudio/vídeo, além de visualização (*preview*) do fluxo de vídeo processado. Atualmente buscou-se um modelo de GUI que unifique as melhores idéias dos softwares livres *flumotion* e *dvswitch*. A Figura 8 ilustra a interface gráfica desta ferramenta.



Figura 8 - Interface gráfica do SLTV

Na Tabela 1 temos um comparativo entre as ferramentas descritas acima em relação ao suporte a pilha TCP/IP, *streaming* de áudio, *streaming* de vídeo, geração de dados estatísticos, sistema distribuído e suporte a uma grande variedade de *codecs* de áudio e vídeo.

Tabela 1- Comparativo entre as ferramentas.

	Suporte TCP/IP	Streaming Áudio	Streaming Vídeo	Gerador de Estatísticas	Sistema Distribuído	Codecs
Grass Valley	Não	Sim (apenas local)	Sim (apenas local)	Não	Não	Não
SuperCollider	Sim (apenas áudio)	Sim	Não	Não	Sim	Não
DICE	Não	Não	Sim (apenas local)	Não	Não	Não
INTERACT	Não	Sim (apenas local)	Sim (apenas local)	Sim	Não	Não
SLTV	Sim	Sim	Sim	Não	Sim	Não

4 A ARTHRON

Neste capítulo é apresentada a ferramenta proposta neste trabalho, onde são descritas uma visão geral, modelo conceitual, o modelo arquitetural, os componentes da ferramenta, descrevendo e definindo o papel de cada um e detalhes da implementação da mesma.

4.1 VISÃO GERAL

A Arthron é uma ferramenta desenvolvida com intuito de gerenciar e articular remotamente fontes distribuídas de mídia com diferentes formatos de codificação em eventos de cunho artístico-tecnológicos. Ela é formada basicamente por cinco componentes que serão descritos nas subseções a seguir, *Decoder*, *Encoder*, *Reflector*, *Articulator* e *VideoServer*, cada um tem um papel específico na preparação de um evento e podem estar localmente ou geograficamente distribuídos na rede.

Outros componentes foram desenvolvidos no decorrer do projeto, *MapManager*, *Monitor* e *ScenarioMaker*, tais componentes não são necessários para o funcionamento da Arthron, mas a auxilia na conectividade, monitoração e criação de cenários artísticos. Eles também serão apresentados nas subseções a seguir.

4.2 MODELO CONCEITUAL

A ferramenta utiliza qualquer rede que dê o mínimo de suporte para transmissão dos fluxos gerados em espetáculos como Versus e InTOque, ou seja, fluxos em alta definição (1440x1080) a taxas de 25Mbps e definição padrão (720x480) a taxas de 8Mbps ou 4Mbps dependendo da técnica de compressão utilizada. Esses fluxos podem ser gerados localmente ou em localizações geográficas diferentes e devem ser controlados e gerenciados pelo *Articulator*. A Figura 9 apresenta o diagrama de casos de uso *Unified Modeling Language* (UML) da ferramenta. Observe que apresentamos três perfis de usuários distintos: (a) telespectador, (b) artista e (c) gerente técnico.

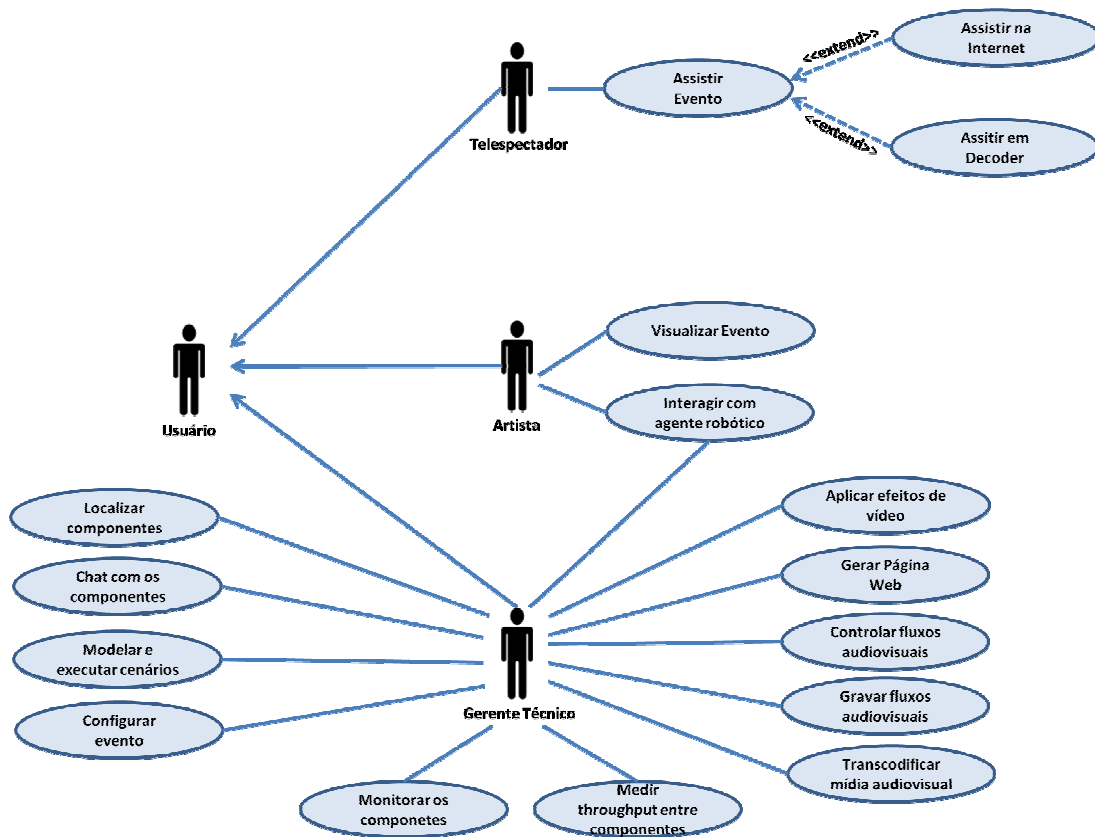


Figura 9 - Diagrama de casos de uso UML.

4.3 MODELO ARQUITETURAL

Desde o início do desenvolvimento do projeto GTMDA a ferramenta Arthron passou por diversas transformações no tocante a sua arquitetura. Será mostrada a evolução da sua arquitetura, pois as suas características poderão ser úteis ao desenvolvimento de outras ferramentas similares.

A primeira arquitetura implementada centralizava a passagem dos fluxos de mídia no componente *Articulator* como podemos ver na Figura 10, fazendo com que ele acumulasse os papéis de gerenciador do sistema, redistribuidor, replicador de fluxos e decodificador de vídeo em alta, padrão e baixa definição. Porém percebeu-se que essa arquitetura não permitia a escalabilidade do sistema uma vez que a concentração desses fluxos em um único ponto causava uma grande sobrecarga de processamento principalmente por causa da decodificação dos vídeos na definição alta e padrão, e da rede à medida que o evento necessitasse de mais fluxos. Outro problema que surgiu dessa arquitetura foi o aumento da quantidade de falha, pois caso o *Articulator* torna-se

inacessível, por falha em rede, ou interrompido, por outra razão, o sistema como um todo entraria em colapso.

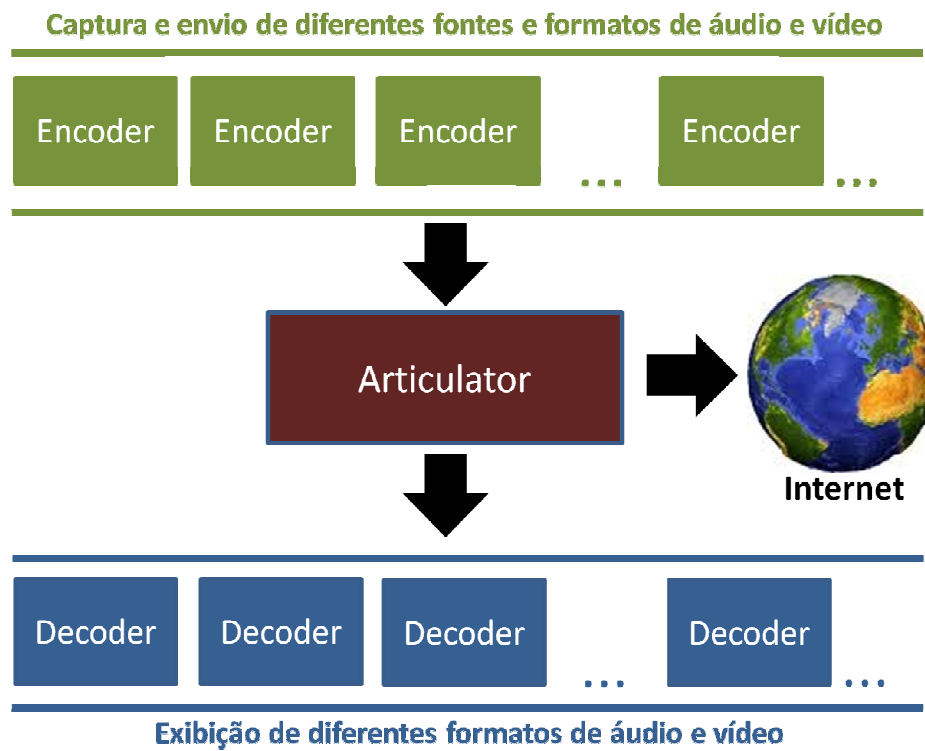


Figura 10 - Arquitetura inicial da Arthron.

Para resolver esses problemas foi proposta uma nova arquitetura que atribuía ao componente *Articulator* apenas a função de gerenciador e acumulava nos *Encoders* a função de capturar, codificar (quando necessário), distribuir e replicar o fluxo que ele mesmo gerava para os destinos finais. Dois problemas foram encontrados nessa nova arquitetura. Um deles é que se verificou um aumento no consumo de processamento dos módulos *Encoders* forçando a utilização de máquinas mais robustas a fim de evitar travamentos no vídeo causados por essa sobrecarga, também se verificou um desperdício na utilização da rede como podemos ver na Figura 11.

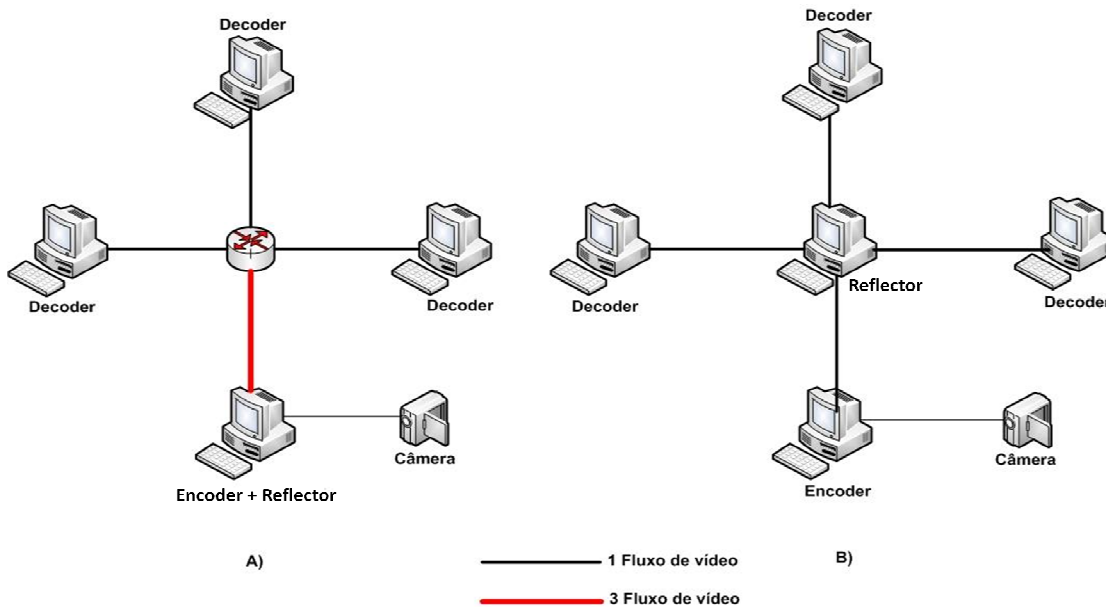


Figura 11 - Redução do fluxo no enlace da rede com a utilização do componente Reflector. A) Envio de três fluxos utilizado o Reflector no Encoder. B) Envio de três fluxos utilizado o Reflector separadamente.

A última arquitetura implementada pode ser vista na Figura 12. Ela visou à diminuição do processamento nos componentes *Encoders* utilizando-se para isso o componente *Reflector* que é responsável pela replicação e redistribuição das mídias. A utilização desse componente também provocou um melhor aproveitamento do uso da rede, pois permitiu inserir um intermediário na retransmissão do fluxo melhorando assim a utilização da rede.

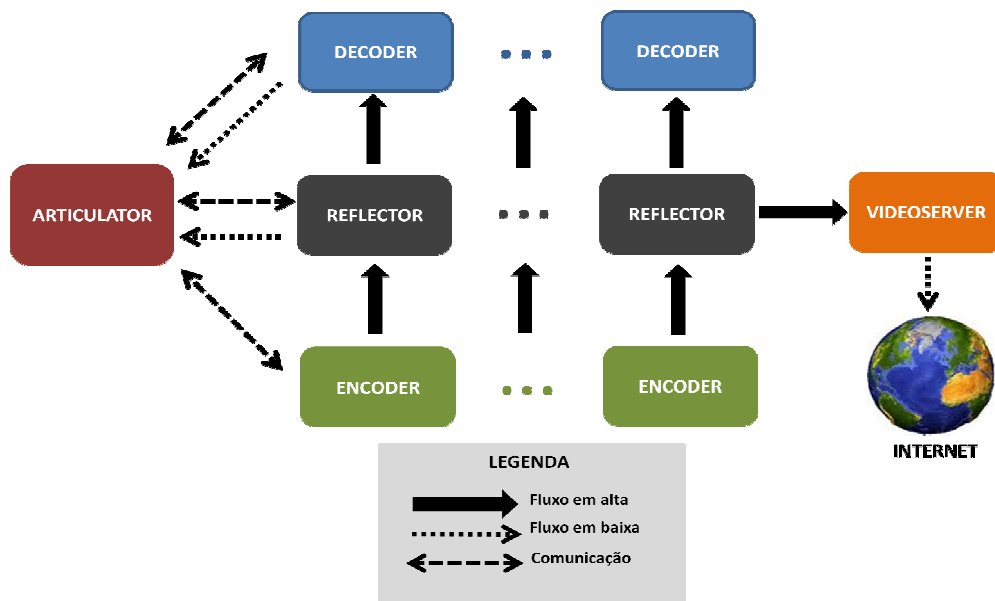


Figura 12 – Arquitetura atual da Arthron.

4.4 COMPONENTES

Nesta seção serão descritas as principais classes que compõem cada componente da ferramenta bem como o pacote *commons* que possui classes que são comuns a todos os componentes.

4.4.1 O Pacote Commons

Esse pacote contém os códigos que são compartilhados com todos os componentes da ferramenta e por isso suas principais classes serão detalhadas antes. Na Figura 13 são ilustradas as principais classes e pacotes que compõem o pacote *commons*.

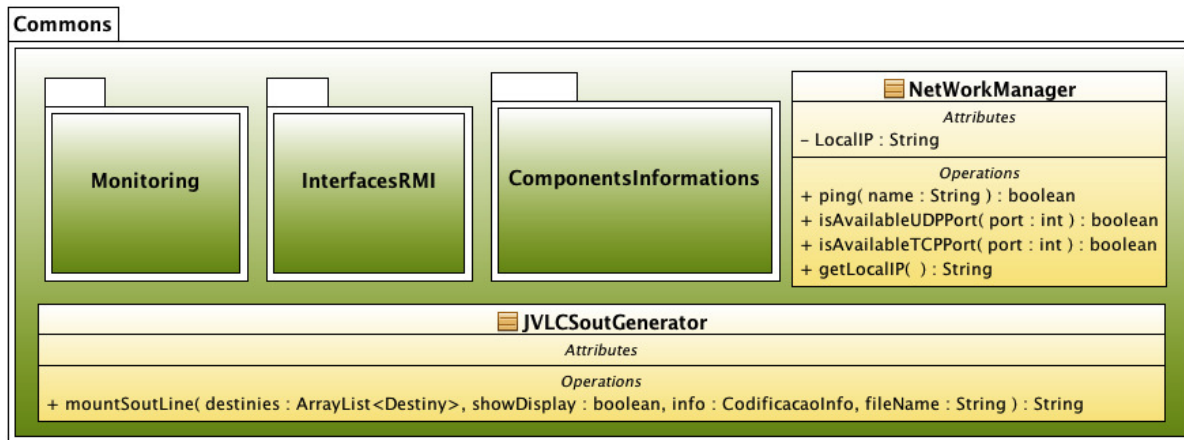


Figura 13 - Principais classes e pacotes do pacote commons.

- **Pacote Monitoring** – Contém as classes que são responsáveis pelo monitoramento via *Simple Network Management Protocol* (SNMP) e medição via Iperf¹⁰. Diversas classes responsáveis pela medição compõem esse pacote.
- **Pacote InterfacesRMI** – Contém as interfaces *Remote Method Invocation* (RMI) de todos os componentes. É necessário que todos os componentes tenham essa informação para saber quais os métodos estão disponíveis em cada componente. As interfaces presentes são: *RMIDecoderInterface*, *RMIEncoderInterface*, *RMIProxyInterface* e *RMIManagerInterface*.

¹⁰ Iperf é um software de análise de performance de banda e cálculo de perda de datagramas na rede que é mantido pela Universidade de Illinois sob licença GPL.

- **Pacote ComponentsInformations** – Contém as classes que representam as informações dos componentes e devem estar sempre sincronizadas com o estado real do componente em questão, essas informações são passadas ao *Articulator* no momento da inicialização para que ele possa aceitar ou não a conexão, e também sempre que houver mudanças de estado. Esse pacote é formado pelas classes *DecoderInformation*, *EncoderInformation* e *ProxyInformation*. Não existe nesse pacote uma classe para representar as informações do *Articulator*, pois essa informação não é passada aos demais componentes que são controlados.
- **Classe NetworkManager** – Essa classe é responsável pela obtenção das informações relativas a rede. Os principais métodos dela são:
 - *ping(name : String) : boolean* – tenta resolver um *hostname*¹¹ e obter uma resposta em até 8 segundos retornando verdadeiro caso tenha sucesso;
 - *isAvaliableTCPPort(port : int) : boolean* – informa se uma dada porta TCP está disponível para uso;
 - *isAvaliableUDPPort(port : int) : boolean* – informa se uma dada porta UDP está disponível para uso e
 - *getLocalIP() : String* – esse método tenta obter o endereço de rede da máquina procurando primeiro IPs registrados em placas de rede cabeadas e depois em placas de rede sem fio, caso não consiga, o endereço de *loopback*¹² é assumido.
- **Classe JVLCSoutGenerator** – Para acionar o player VLC é necessário o envio de um comando que informa o que se deve ser feito. Esse comando é passado em forma de uma *string* com um padrão próprio. Para facilitar a utilização do VLC foi criada uma classe que forma a *string* de saída do fluxo

¹¹ É um rótulo que é atribuído a um dispositivo que esteja conectado a uma rede de computadores. Ele é usado para identificar o dispositivo em diversas formas de comunicação eletrônica, tais como a *world wide web*, *e-mail* ou *usenet*.

¹² É um endereço que permite a comunicação com a máquina atual como se estivesse se comunicando com uma máquina remota é útil para a finalidade de testes, assim como para usar recursos localizados na máquina atual, mas que se esperariam serem remotos. Normalmente é representado pelo IP 127.0.0.1.

certa a partir dos parâmetros que lhe são passados. O único método que a compõe é o seguinte:

- *mountSoutLine(destinies : ArrayList<Destiny>, showDisplay : boolean, info : CodificacaoInfo, fileName : String) : String* – onde a variável *destinies* informa todos os destinos que o *stream* deve ser enviado, *showDisplay* informa se a mídia deve ser reproduzida localmente, *info* informa a codificação de áudio e de vídeo que deve ser aplicada à mídia e *fileName* indica o nome do arquivo em que a mídia deve ser gravada, caso não se queira efetuar a gravação uma referência nula deverá ser passada.

4.4.2 Decoder

O *Decoder* tem como funcionalidade principal a exibição de um único fluxo de mídia, ou seja, decodificar e exibir na saída de algum dispositivo (monitor, projetor, etc) um fluxo específico. Esse componente se utiliza de uma interface simples, não necessitando do usuário muito conhecimento para utilizá-lo. A captura do fluxo é feita através do protocolo *User Datagram Protocol* (UDP) em uma porta que é previamente e automaticamente combinada com o *Articulator*. Atualmente esse componente é utilizado apenas no sistema operacional (SO) Linux e possui um componente *Reflector* interno que por motivos de desempenho só funciona nesse SO. Mas detalhes sobre o não funcionamento desse componente no SO Windows serão explicados na subseção que descreve o *Reflector*.

No *Decoder* é possível utilizar diversas personalizações de exibição do vídeo como, por exemplo, tela cheia, desentrelaçamento, aspect ratio, entre outros que promovem uma melhor adequação do vídeo para o usuário final, além disso, é possível gravação da mídia em exibição em uma unidade de armazenamento como um *Hard Disk* (HD) interno, onde todos esses parâmetros são configurados remotamente pelo componente *Articulator*. Outra funcionalidade introduzida no *Decoder* é a utilização de animações, que são pré-programadas no *Articulator*, sendo exibidas sempre em tela cheia, sobrepondo o fluxo de vídeo que esteja sendo exibido. A utilização dessa animação com uma taxa de quadro pequeno funciona como *Slide Show*.

A configuração desse componente é feita basicamente em dois passos, no primeiro deve-se cadastrar as informações locais que serão avaliadas pelo *Articulator* no momento da conexão, os dados solicitados são, identificação, nome, email, telefone, *Internet Protocol* (IP) local, porta RMI local e um campo adicional para observações que sejam pertinentes. A segunda parte compreende no cadastro dos IPs dos *Articulators* que poderão ser contatados em ordem de prioridade, ou seja, o primeiro cadastrado na lista de *Articulators* e que esteja ativo será estabelecida a conexão.

Na Figura 14 estão listadas as principais classes que compõem esse componente.

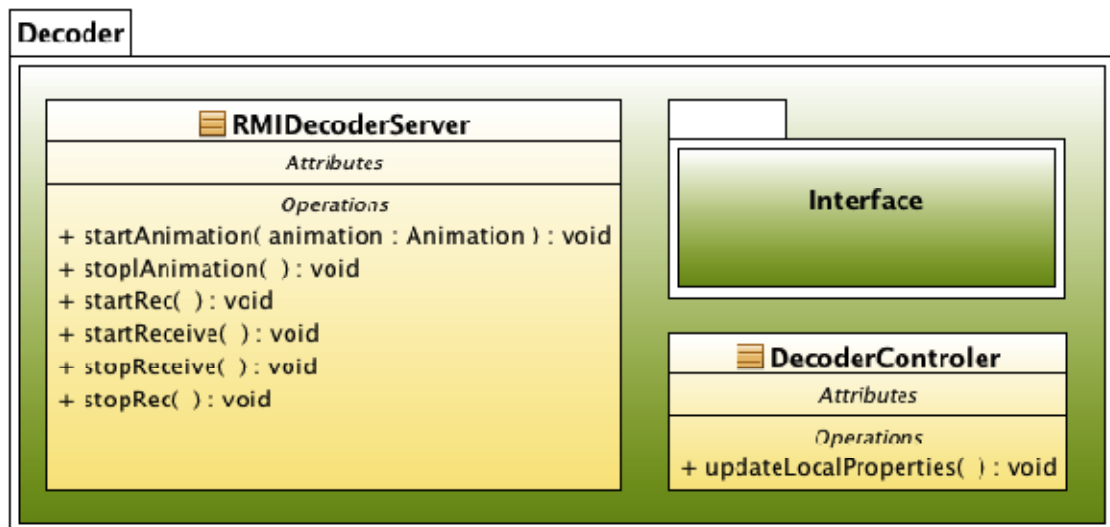


Figura 14 – Principais classes do componente Decoder.

As principais classes são detalhadas a seguir:

- **RMIDecoderServer:** é a implementação da interface RMI do lado do *Decoder* com sua respectiva implementação a fim de permitir a sua utilização no lado do *Articulator*. Os principais métodos que a compõem são:
 - *startAnimation(animation : Animation) : void* – executa imediatamente a animação que for passada como parâmetro, sobrepondo a mídia que esteja sendo exibida no momento;
 - *stopAnimation() : void* – interrompe qualquer animação que esteja sendo exibida, caso exista uma mídia sendo decodificada em segundo plano ela voltará a ser exibida na tela principal;

- *startRec()* : *void* – inicia a gravação da mídia localmente. O nome da mídia assuma o padrão DD_MM_YYYY_hh_mm_ss.ts, onde DD é o dia, MM é o mês, YYYY é o ano, hh é a hora, mm é os minutos e ss os segundos em que a gravação se iniciou;
- *stopRec()* : *void* – interrompe a gravação da mídia;
- *startReceive()* : *void* – inicia o recebimento de um fluxo UDP na porta que foi estabelecida com o *Articulator* e
- *stopReceive()* : *void* – interrompe o recebimento do fluxo UDP.
- **DecoderControler** – Essa classe faz a mediação da comunicação entre a interface gráfica e as demais classes. O seu objetivo é separar a interface da implementação e centralizar as principais funções do componente.
 - *updateLocalProperties()* : *void* – atualiza as propriedades local do *Decoder* e envia uma mensagem para o *Articulator* que esteja conectado a ele informando as mudanças ocorridas.

4.4.3 Encoder

É responsável por fazer a captura, codificação (quando necessária) da fonte de mídia, podendo esta ser tanto de uma fonte externa (filmadora, DVD, vídeo cassete, etc) como por meio de um arquivo local e fazer seu envio para um *Reflector* que irá fazer a distribuição para os destinos configurados no *Articulator*, esse *Reflector* pode ser interno ao *Encoder*, porém esta opção só irá funcionar no SO Linux. O *Encoder* também se utiliza de uma interface simples voltada a usuários principiantes. As codificações de vídeo em software permitidas atualmente são MPEG-4, MPEG-2, DIV3, WMV, MJPG com a taxa variando de 8 a 8192 Kbps as de áudio são MPGA, MP2A, MP4A e MP3 com a taxa variando de 8 a 512 Kbps, porém é permitido enviar taxas e codificações diferentes bastando que essa seja codificada na fonte, preferencialmente por um hardware específico como a hauppauge PVR 350¹³ (MPEG-2) ou HD PVR 1212¹⁴ (H.264). Essas codificações foram escolhidas por serem as mais

¹³ http://www.hauppauge.com/html/wintvpvr350_datasheet.htm

¹⁴ http://www.hauppauge.com/site/products/data_hdvpvr.html

comumente suportadas via software, e as taxas foram impostas pelo VLC [37] que tem essa limitação.

A maioria das funcionalidades do *Encoder* pode ser controlada localmente ou pelo *Articulator*, o sistema se encarrega de manter as informações sincronizadas, porém a indicação de utilização de um *Reflector* externo e a indicação da disponibilidade de um agente robótico deve ser feita durante a inicialização pelo utilizador do *Encoder*. Caso haja um agente robótico disponível deve-se informar o IP e a porta de comunicação, toda a comunicação é feita obedecendo ao protocolo da Galatéia¹⁵ que é baseado em um byte como descrito na Tabela 2.

Tabela 2 – Protocolo do agente robótico Galatéia.

Protocolo	Resultado
00000000	Parar
00001010	Cima
00000110	Esquerda
00000101	Baixo
00001001	Direita

Esse protocolo ainda permite uma grande variedade de outras combinações permitindo acomodar comandos adicionais.

A configuração do componente *Encoder* é simples e está dividida basicamente em quatro passos, o primeiro consiste em informar se há necessidade de alocação de um *Reflector* externo, essa opção deve ser escolhida sempre que a máquina que faz a captura da mídia tiver um desempenho inferior ou houver necessidade de otimizar o caminho do fluxo pela rede. O segundo passo é indicar as informações locais que serão avaliadas pelo *Articulator* na fase de conexão. Os dados solicitados são: identificação, nome, e-mail, telefone, IP local, porta RMI local, existência ou não de agente robótico disponível e caso haja as informações que permitam o contato, e um campo adicional para observações que sejam pertinentes. O terceiro passo compreende o cadastro dos *Articulators* que poderão ser contatados em ordem de prioridade. O quarto passo é a definição da codificação em que a mídia será transmitida. A Figura 15 ilustra as principais classes que compõem esse componente.

¹⁵ É um robô Pioneer AT utilizado pelo laboratório NatalNet na UFRN.

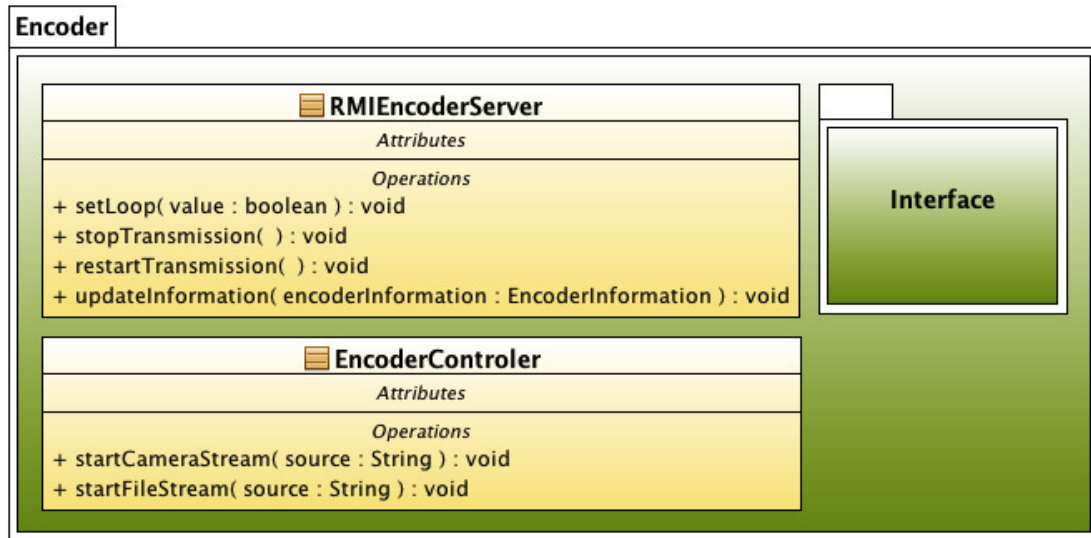


Figura 15 – Principais classes do componente Encoder.

- **RMIEncoderServer** – é a implementação da interface RMI do lado do Encoder com sua respectiva implementação a fim de permitir a sua utilização no lado *Articulator*. Os principais métodos que o compõe são:
 - *setLoop(value : boolean) : void* – informa se deseja-se reiniciar a exibição da mídia após seu término. Funciona apenas para mídias que forem carregadas através de arquivos.
 - *stopTransmission() : void* – interrompe a transmissão do fluxo de mídia que esteja sendo enviado.
 - *restartTransmission() : void* – reinicia a transmissão do fluxo de mídia que estava sendo enviado, mantendo sua origem, câmera ou arquivo.
 - *updateInformation (encoderInformation : EncoderInformation) : void* – utilizada para atualizar as informações relativas ao *Encoder* que foram modificadas no *Articulator*.
- **EncoderControler** – Essa classe faz a mediação da comunicação entre a interface gráfica e as demais classes. O seu objetivo é separar a interface da implementação e centralizar as principais funções do componente.
 - *startCameraStream(source : String) : void* - Iniciar um fluxo de mídia a partir de uma câmera que esteja conectada ao *Encoder*, a *String source* indica a localização e os parâmetros adicionais. O

padrão do *source* é o definido pelo VLC, atualmente estamos usando apenas dois desses padrões:

- No linux: *pvr://* que indica a captura através de um dispositivo (placas como hauppauge, visiotec, dentro outras).
 - No windows: *dshow://* que indica a captura de um dispositivo via entrada IEEE1394 (firewire).
- *startFileStream(source : String) : void* – iniciar um fluxo de mídia a partir de um arquivo, a *String source* indica a localização absoluta¹⁶ do arquivo.

4.4.4 Reflector

Em redes de computadores, um *Proxy* é um servidor (um sistema de computador ou um programa de aplicação) que atua como um intermediário para solicitações de clientes que procuram recursos de outros servidores. Um cliente conecta-se ao servidor *Proxy*, solicitando algum serviço, como um arquivo, conexão, página web, conteúdos multimídia (áudio e vídeo) ou outro recurso disponível a partir de um servidor diferente. O servidor *Proxy* avalia o pedido de acordo com suas regras de filtragem. Por exemplo, pode filtrar o tráfego por endereço IP ou protocolo. Se o pedido for validado pelo filtro, o *Proxy* fornece o recurso de conexão com o servidor em questão e solicita o serviço em nome do cliente. Um servidor *Proxy* pode, opcionalmente, alterar a solicitação do cliente ou a resposta do servidor, e às vezes pode atender à solicitação sem contato com o servidor especificado, em um esquema de *cache*.

Na Arthron, a função de *Proxy* é destinada ao *Reflector*, que é o componente intermediário cujas funções são a replicação, redistribuição e transcodificação (apenas para o *Articulator*) de um determinado fluxo de mídia (áudio e vídeo) sobre a rede. Ele faz isso de duas maneiras. Uma é o envio direto, ou seja, replicado e redistribuindo sem realizar qualquer alteração no fluxo original para um ou vários *Decoders* especificados pelo *Articulator* e a outra é transcodificando um fluxo em uma taxa mais baixa para monitoramento pelo *Articulator*.

¹⁶ É o caminho completo, exemplo: /home/gtmda/videos/espetaculo.ts

O *Reflector* pode ser inserido internamente a um dos componentes da Arthron, porém a fim de otimizar o envio de um fluxo é recomendável que ele seja colocado em pontos estratégicos da rede como pode ser visto no comparativo da Figura 11. Outra vantagem do seu uso distribuído é a redução do processamento em um dado componente que não tem na distribuição a sua função principal.

O desenvolvimento desse componente em sistemas Windows foi descontinuado, pois a distribuição ficava limitada a uma taxa considerada muito baixa por algum detalhe de arquitetura do sistema operacional, enquanto esse problema não é resolvido só está liberado o seu uso em sistemas Linux.

Na Figura 16 estão ilustradas as principais classes que compõem esse componente que são detalhadas a seguir.

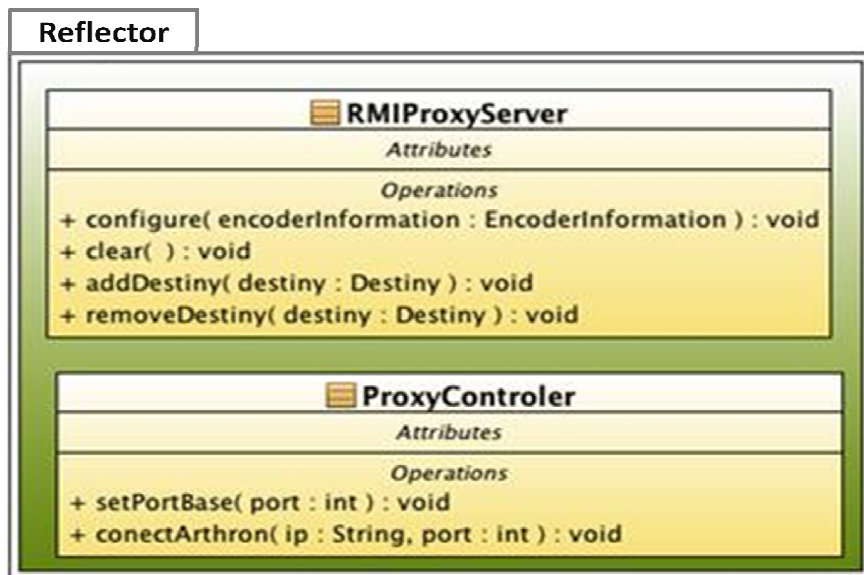


Figura 16 - Principais classes do componente Reflector.

- **RMIProxyServer:** é a implementação da interface RMI do lado do *Reflector* com sua respectiva implementação a fim de permitir a sua utilização no lado do *Articulator*.
 - `configure(encoderInformation : EncoderInformation) : void` – efetua a configuração do *Reflector* de modo que ele possa trabalhar corretamente com o *Encoder* que lhe for associado.

- *clear()* : *void* – remove a associação do *Reflector* com qualquer *Encoder* que esteja associado a ele, também interrompe todas as distribuições de fluxos.
- *addDestiny(destiny : Destiny)* : *void* – adiciona um novo destino de distribuição de fluxo.
- *removeDestiny(destiny : Destiny)* : *void* – remove um destino de distribuição de fluxo.
- **ProxyControler:** essa classe faz a mediação da comunicação entre a interface gráfica e as demais classes. O seu objetivo é separar a interface da implementação e centralizar as principais funções do componente.
 - *setPortBase(port : int)* : *void* – altera a porta base do *Reflector* para a que lhe for passada por parâmetro.
 - *conectArthron(ip : String, port : int)* : *void* – efetua a conexão com o *Articulator* representado pelo IP e Porta que for passada por parâmetro.

4.4.5 Articulator

É o principal e mais complexo componente da Arthron, ele é responsável pelo gerenciamento remoto dos demais componentes concentrando grande parte das funcionalidades. Por motivos de desempenho a sua utilização é restrita ao SO Linux.

Abaixo são descritas algumas das funcionalidades:

- **Programação de fluxos manual** – permite enviar fluxos de mídia de *Encoders* para *Decoders* com apenas um arrastar do mouse, sendo este de forma imediata ou a partir de uma programação temporal, podendo para isso se utilizar de um *Reflector* intermediário. É considerada a funcionalidade principal do sistema por se tratar de um requisito primordial para execução de espetáculos artístico-tecnológicos.
- **Gerenciamento automático dos fluxos** – essa foi uma necessidade que surgiu justamente da execução da troca dos fluxos entre as diversas telas do espetáculo “e-Pormundos Afeto” [38]. Anteriormente as trocas eram executadas manualmente ou programadas no tempo manualmente, o que

causava certo desconforto e estresse durante a execução por parte de quem comandava o componente *Articulator*. Em um espetáculo não há como determinar o tempo exato para passagem entre as cenas, pois isso é determinado pela coreografia do artista. Detectou-se a necessidade da criação de um arquivo de configuração de trocas de fluxos (passagem de cenas) automático, ou seja, pré-configurado pelo usuário e executado pelo sistema e outro automatizado, onde o usuário programa uma sequência e apenas apertar uma tecla para próxima troca de vídeo e assim sucessivamente, ou seja, temos eventos com passagens síncronas e assíncronas.

- **Chat** – permite que o usuário do *Articulator* se comunique com os usuários de outros componentes (apenas *Encoders* e *Decoders*) conectados através de janelas de bate-papos, Figura 17. Esse requisito surgiu da necessidade de comunicação da equipe (Brasil e Espanha) que manipulava os componentes da Arthron durante o espetáculo “e-Pormundos Afeto”.
- **Geração automática de página web** – Essa funcionalidade surgiu da necessidade de publicação dos vídeos capturados durante o espetáculo em uma página padrão com vários tipos de *players*. Permite a geração de uma página (.html) padrão para publicação dos vídeos na Internet através da interface do componente *Articulator*. O usuário não precisa de conhecimentos específicos de programação web para gerá-la, ou seja, ele só precisa escolher o fluxo que deseja transmitir para a Internet, o formato que vai ser à saída desse vídeo e o tipo do *player* que irá tocar tal vídeo. O objetivo dessa funcionalidade é permitir com que se atinja um número maior de espectadores, pois são oferecidas várias opções de visualização dos vídeos em diferentes taxas e formatos, Figura 18.
- **Programação de animação** – permite programar animações (imagens jpeg, gifs, etc) nos *Decoders*. A montagem da animação é feita no *Articulator* ou carregada de arquivo e enviada pela rede para os *Decoders* que irão exibi-la.
- **Monitoramento e medição** – é possível efetuar o monitoramento dos componentes que estejam conectados ao *Articulator*, porém eles deverão ter

previamente instalado o pacote SNMP. A medição da rede é feita utilizando o Iperf e cada componente envolvido com a conexão também deve ter ele instalado.

- **Modificação de configurações** - É possível alterar as configurações (nome, e-mail, telefone, etc) dos demais componentes remotamente pelo *Articulator*, para isso basta que cada componente implemente a função correspondente em sua Interface RMI.
- **Bloqueio e liberação** – no momento da conexão são exibidos para o *Articulator* os dados do solicitante, somente no caso do seu IP não tiver sido bloqueado ou liberado. Caso esteja bloqueado a solicitação será imediatamente negada e caso seja liberado a solicitação será imediatamente aceita.
- **Modificação dos parâmetros dos fluxos** – permite acessar qualquer *Encoder* ou *Decoder* conectado ao *Articulator* e alterar quando necessário os parâmetros de codificação, *aspect ratio*, desentrelaçamento, dentro outros dos fluxos de mídia. Outro requisito imprescindível na realização de espetáculos artístico-tecnológicos.

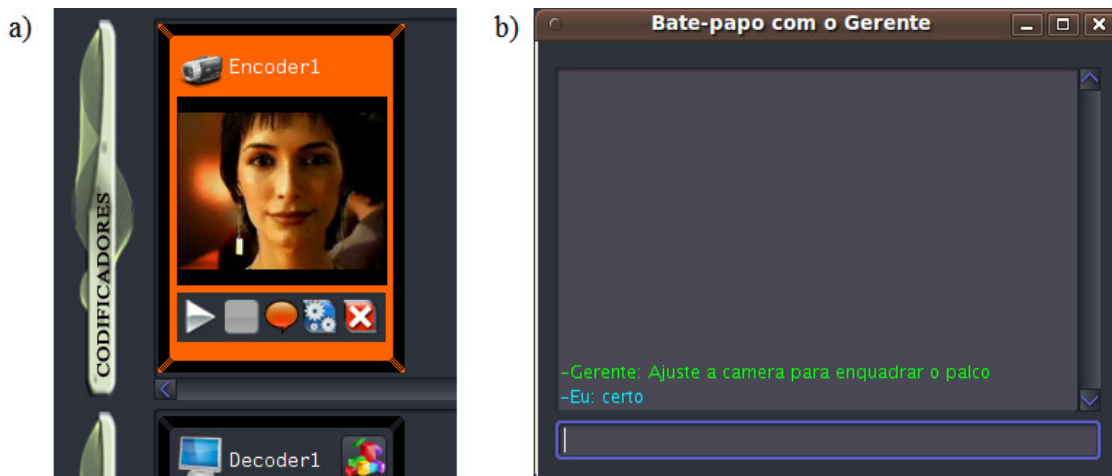


Figura 17 - Chat entre o Articulator e um Encoder. a) A cor do Encoder é alterada quando se comunicar com o Articulator. b) Encoder se comunicando.



Figura 18 - Página gerada automaticamente a partir da escolha de um Decoder.

É recomendável fazer a medição da rede (com Iperf) e a correta configuração dos componentes antes do início de uma apresentação.

Na Figura 19 estão ilustradas as principais classes que compõem esse componente que são detalhadas a seguir.

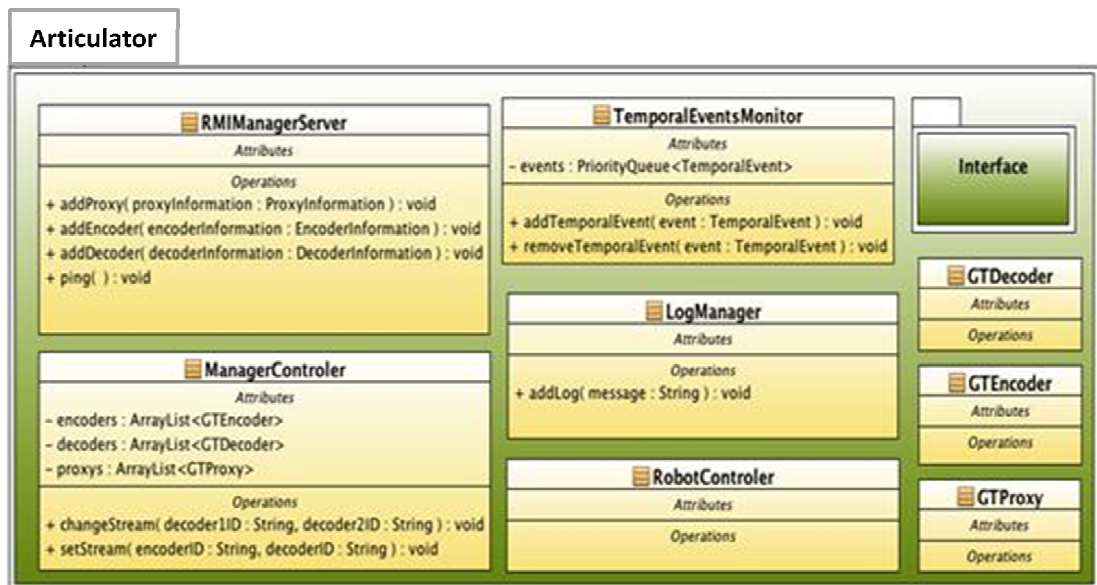


Figura 19 - Principais classes do componente Articulator.

- **RMIManagerServer** – é a implementação da interface RMI do lado do *Articulator* com sua respectiva implementação a fim de permitir a sua utilização pelos demais componentes.

- *addProxy(proxyInformation : ProxyInformation) : void* – adiciona um *Reflector* permitindo que ele seja gerenciado pelo *Articulator*.
- *addEncoder(encoderInformation : EncoderInformation) : void* – efetua uma solicitação de adição de um *Encoder*, todas as suas informações são exibidas para que o *Articulator* possa decidir aceitá-lo ou não.
- *addDecoder(decoderInformation : DecoderInformation) : void* – efetua uma solicitação de adição de um *Decoder*, todas as suas informações são exibidas para que o *Articulator* possa decidir aceitá-lo ou não.
- *ping() : void* – é um comando que usa o protocolo *Internet Control Message Protocol* (ICMP¹⁷) para testar a conectividade entre o componente solicitante e o *Articulator*. Seu funcionamento consiste no envio de pacotes para o *Articulator* e aguardo de uma resposta.
- **ManagerControler** – essa classe faz a mediação da comunicação entre a interface gráfica e as demais classes. O seu objetivo é separar a interface da implementação e centralizar as principais funções do componente.
 - *encoders : ArrayList<GTEncoder>* - é um atributo que representa todos os *Encoders* que estejam conectados.
 - *decoders : ArrayList<GTDecoder>* - é um atributo que representa todos os *Decoders* que estejam conectados.
 - *proxies : ArrayList<GTProxy>* - é um atributo que representa todos os *Reflectors* que estejam conectados.
 - *changeStream(decoderIID : String, decoder2ID : String) : void* – efetua uma troca de fluxos. O fluxo que estiver sendo recebido pelo *Decoder01* será enviado para o *Decoder02* e vice-versa.

¹⁷ É um protocolo integrante do Protocolo IP, definido pelo RFC 792, e utilizado para fornecer relatórios de erros à fonte original. Qualquer computador que utilize IP precisa aceitar as mensagens ICMP e alterar o seu comportamento de acordo com o erro relatado. Os *gateways* devem estar programados para enviar mensagens ICMP quando receberem datagramas que provoquem algum erro.

- *setStream (encoderID : String, decoderID : String) : void* – atribui um fluxo de um *Encoder* para um *Decoder* cancelando qualquer fluxo que esse *Decoder* estiver recebendo previamente.
- **TemporalEventsMonitor** – essa classe é responsável por gerenciar os eventos temporais que estiverem cadastrados. Esses eventos atualmente podem ser de atribuição de fluxo, troca de fluxo e atribuição de animação.
 - *events : PriorityQueue<TemporalEvent>* - é um atributo que contém todos os eventos temporais que estiverem cadastrados ordenados pelo seu tempo de início de execução. A cada segundo é verificado se o primeiro elemento da fila está pronto para ser executado.
 - *addTemporalEvent(event : TemporalEvent) : void* – adiciona um evento temporal à fila de prioridade¹⁸.
 - *removeTemporalEvent(event : TemporalEvent) : void* – remove um evento temporal da fila de prioridade.
- **LogManager** – essa classe é responsável por armazenar um *Log* dos principais eventos ocorridos no *Articulator*.
 - *addLog(message : String) : void* – adiciona uma mensagem indicando algum procedimento realizado no *Articulator*.
- **RobotControler** – essa classe é responsável pelo intermédio da comunicação com o robô.
- **GTDecoder** – essa classe representa um componente *Decoder* remoto que esteja conectado ao *Articulator*. As suas informações devem estar sempre sincronizadas.
- **GTEncoder** – essa classe representa um componente *Encoder* remoto que esteja conectado ao *Articulator*. As suas informações devem estar sempre sincronizadas.
- **GTProxy** – essa classe representa um componente *Reflector* remoto que esteja conectado ao *Articulator*. As suas informações devem estar sempre sincronizadas.

¹⁸ São filas que tem seus elementos ordenados por algum atributo.

4.4.6 MapManager

Componente desenvolvido com intuito de permitir a visualização em um mapa-múndi da localização (Latitude e Longitude) de alguns componentes que fazem parte da Arthron. Basicamente, o *MapManager* é um servidor que guarda a informação que cada componente (*Encoder*, *Decoder* e *Articulator*) o passa, como nome, latitude, longitude, tipo, etc. No momento que um determinado componente se conecta a ele, ele retorna para o respectivo componente uma lista contendo todos os componentes que estão conectados ao servidor. Assim, é possível inserir os marcadores que irão representar cada componente no mapa. Outra função importante é a interatividade que o mapa permite aos componentes *Encoder* e *Decoder*, ou seja, o usuário de um determinado componente tem a possibilidade de se conectar ao *Articulator* com um simples clicar do *mouse* no mapa, assim, o usuário não tem necessidade de saber qual o endereço IP de um determinado *Articulator* ativo.

Na Figura 20 estão ilustradas as principais classes que compõem esse componente que são detalhadas a seguir.

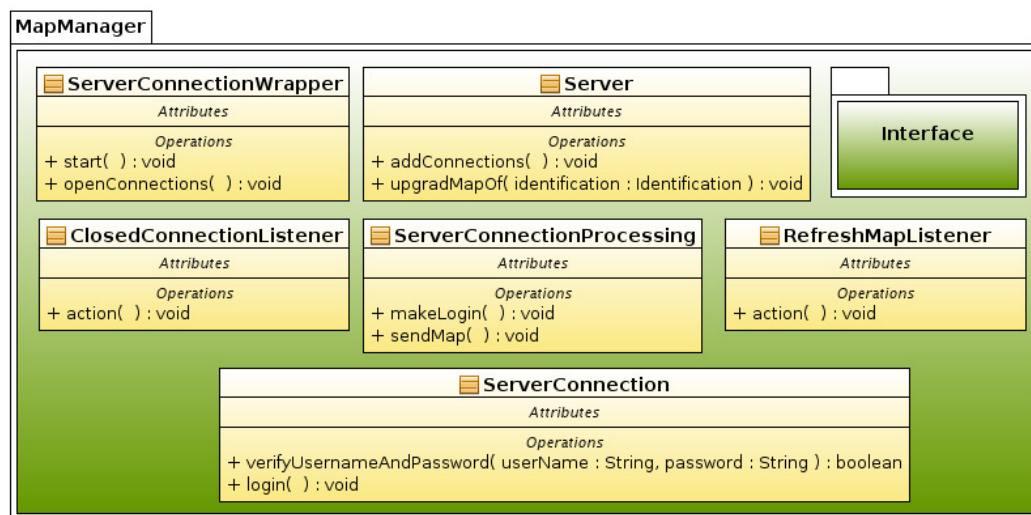


Figura 20 - Principais classes do componente MapManager.

- **ServerConnectionProcessing** – essa classe contém métodos para o processamento de uma nova conexão como uma nova *thread*.
- **ServerConnectionWrapper** – essa classe tem a finalidade de carregar as configurações de conexão ou receber essas configurações através de seus métodos e realizar a conexão do servidor.

- **Server** – essa classe representa um servidor esperando conexões por sockets¹⁹.
- **RefreshMapListener** – essa classe representa um *listener* que é chamado quando o mapa do servidor for alterado.
- **ServerConnection** – essa classe abstrata representa um servidor em uma comunicação via sockets com clientes.
- **ClosedConnectionListener** – essa classe representa um *listener* de fechamento de conexões.

4.4.7 VideoServer

Esse componente surgiu da necessidade de contemplar a transmissão de vídeos em baixa definição para web em vários formatos, especificamente, nos formatos mais utilizados hoje em dia, tais como FLV²⁰, OGG e H.264, possibilitando assim uma gama maior de opções para usuários que visualizam os vídeos transmitidos na *Internet*. Porém, para realizar tais transmissões em diferentes formatos é necessária uma máquina robusta que possa realizar a transcodificação do conteúdo audiovisual original para tais formatos especificados. O *VideoServer* é um componente que está associado diretamente a funcionalidade já citada do componente *Articulator* chamada “Geração automática de página web”, ou seja, através da interface do *Articulator* o usuário escolhe um *Decoder* específico para publicação do conteúdo na *Internet* e o *VideoServer* que irá receber tal conteúdo para realizar a transcodificação no formato desejado. As informações do formato escolhido, tais como, taxa de transmissão de áudio e vídeo e resolução do vídeo são especificadas também pelo usuário na interface do *Articulator*. O *VideoServer* também funciona como um servidor de conteúdo audiovisual, pois ele fica aguardando os pedidos dos usuários na web a partir da página gerada pelo *Articulator*, onde a transmissão do conteúdo se dá via HTTP.

Na Figura 21 está ilustrada a principal classe que compõe esse componente.

¹⁹ Socket é a interface entre a camada de aplicação e a de transporte dentro de uma máquina. Ele é a “porta” na qual os processos enviam e recebem informações (dados).

²⁰ FLV ou Flash Video é um método de encapsulamento muito utilizado para transmissão de vídeo na Internet usando as versões do Adobe Flash Player 6 a 10.

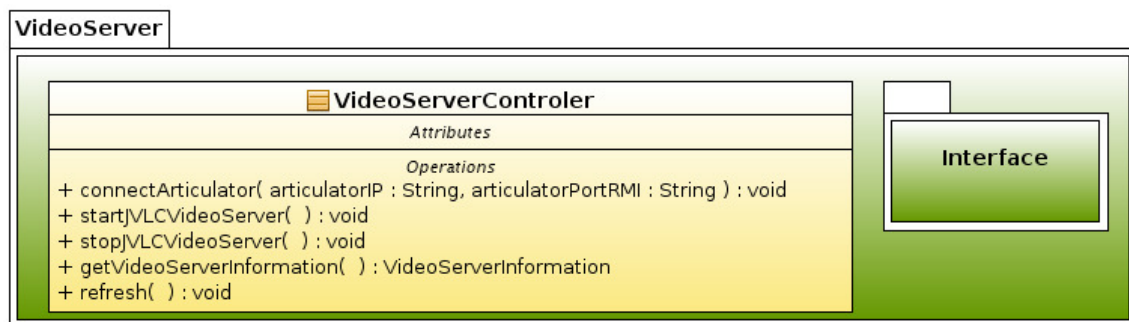


Figura 21 - Principais classes do componente VideoServer.

4.4.8 Monitor

A funcionalidade já citada do *Articulator* de “Monitoramento e medição” foi criada para que o usuário da interface deste componente pudesse visualizar, avaliar, medir e verificar o comportamento de outros componentes em tempo real de forma fácil, tais como uso de CPU, memória RAM, bytes enviados e recebidos, espaço em disco, etc. Porém, além disso, verificou-se a necessidade da criação de um componente a parte que ficasse exclusivamente com essa função de monitorar e gravar tais informações em um arquivo para posterior análise, com o intuito de analisar o impacto causado com a execução de um espetáculo midiático como o “e-Pormundos Afeto”.

O componente *Monitor* utiliza o protocolo SNMP para realizar o monitoramento remoto de máquinas onde estão sendo executados os componentes da Arthron. Bastando, apenas para isso, passar o endereço IP do componente que se deseja monitorar. Assim, é criada uma pasta denominada monitoramento e, dentro desta, uma pasta para cada componente cadastrado e dentro destas o arquivo contendo as informações salvas com data e hora para cada informação adquirida. Essas informações são: bytes enviados na entrada e na saída, memória RAM, espaço no HD dentre outras.

Na Figura 22 estão ilustradas as principais classes que compõem esse componente que são detalhadas a seguir.

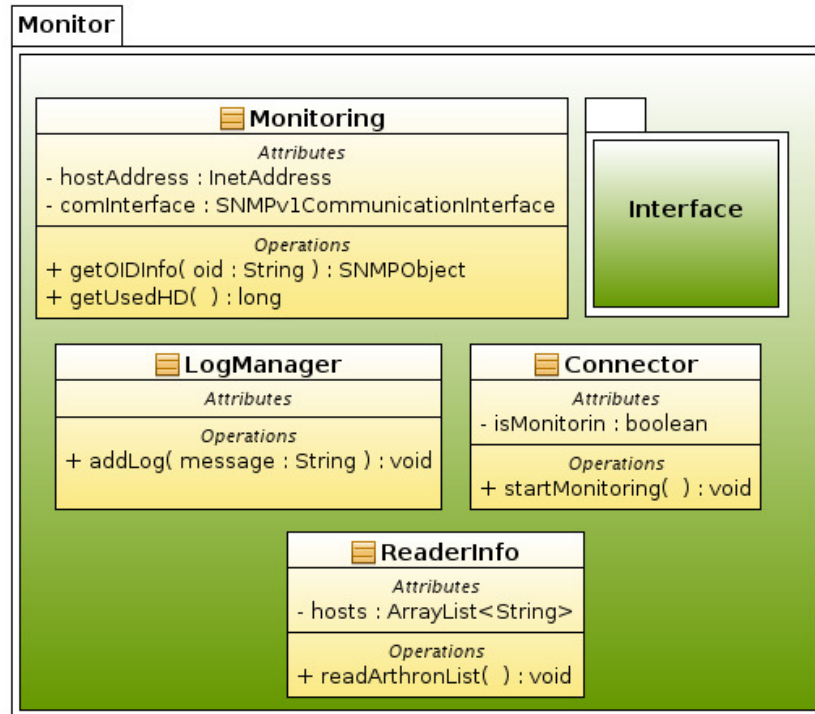


Figura 22 - Principais classes do componente Monitor.

- **Monitoring** – essa classe é responsável por fazer a abstração da utilização do *parser* do SNMP para Java. Seus principais métodos são:
 - *getOIDInfo(oid : String)* – método pelo qual é passado o *object ID* (OID) que se quer saber a informação desejada, como memória, processamento, etc.
 - *getUsedHD* – neste método é passado o OID referente ao HD (1.3.6.1.4.1.2021.9.1.8.1) e com isso, é retornado um valor do tipo long referente ao valor do percentual de uso do HD. A classe apresenta inúmeros outros métodos semelhantes, onde apenas o OID é alterado para o que se deseja saber sobre um host remoto.
- **LogManager** – classe responsável por armazenar o log de atividades do monitoramento. Assim, para cada host, é salvo um arquivo contendo todas as informações que foram cadastradas na classe *Monitoring*.
 - *addLog(message : String) : void* – método responsável por armazenar o log no arquivo.

- **Connector** – classe responsável por iniciar uma *thread* que ficará verificando em loop a lista de host que foram registrados na interface gráfica.
 - *startMonitoring()* : *void* – método responsável por instanciar uma *thread* que irá verificar toda a lista de hosts cadastrados, fazer as solicitações das informações via SNMP e salvar em um arquivo tais informações.
- **ReaderInfo** – classe responsável por ler a lista de *Articulators* num arquivo texto e salvar numa lista.

4.4.9 ScenarioMaker

No evento artístico “e-Pormundos Afeto” onde a ferramenta foi utilizada, percebeu-se que as cenas do espetáculo já eram pré-definidas como na Tabela 5, e que o chaveamento dos fluxos no Articulador eram sempre os mesmos, surgiu então necessidade de automatizar esse processo.

Assim, pensando nisso, foi desenvolvido um componente denominado *ScenarioMaker* com o qual o usuário especifica de forma prévia para qual *Decoder* vai o fluxo de um determinado *Encoder* e em que momento.

Na Figura 23 estão ilustradas as principais classes que compõem esse componente que são detalhadas a seguir.

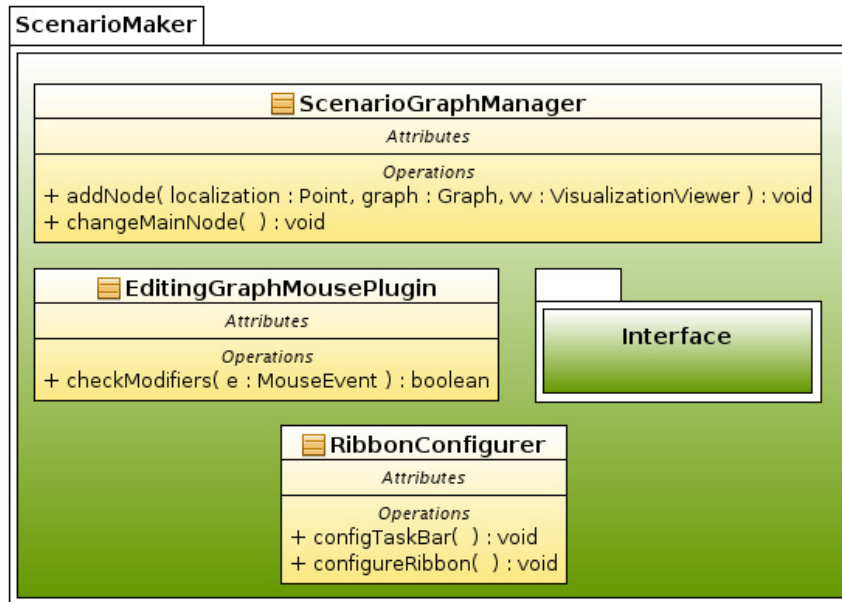


Figura 23 - Principais classes do componente ScenarioMaker.

- **ScenarioGraphManager** – esta classe é responsável por fazer o gerenciamento de todo o cenário no qual os nós e arestas são inseridos.
 - *addNode(localization : Pont, graph : Graph, vv : VisualizationViewer)* – método responsável por adicionar ao cenário um nó de um grafo, em uma certa localização.
- **EditingGraphMousePlugin** – classe responsável por permitir a criação de vértices, nós e arestas usando o mouse.
- **RibbonConfigurer** – classe responsável por configurar uma interface gráfica com componentes diferente dos *swing* convencionais.

4.5 DETALHES DE IMPLEMENTAÇÃO

Na Figura 24 estão ilustradas as tecnologias que foram utilizadas no desenvolvimento da Arthron. Elas serão detalhadas nas subseções a seguir.

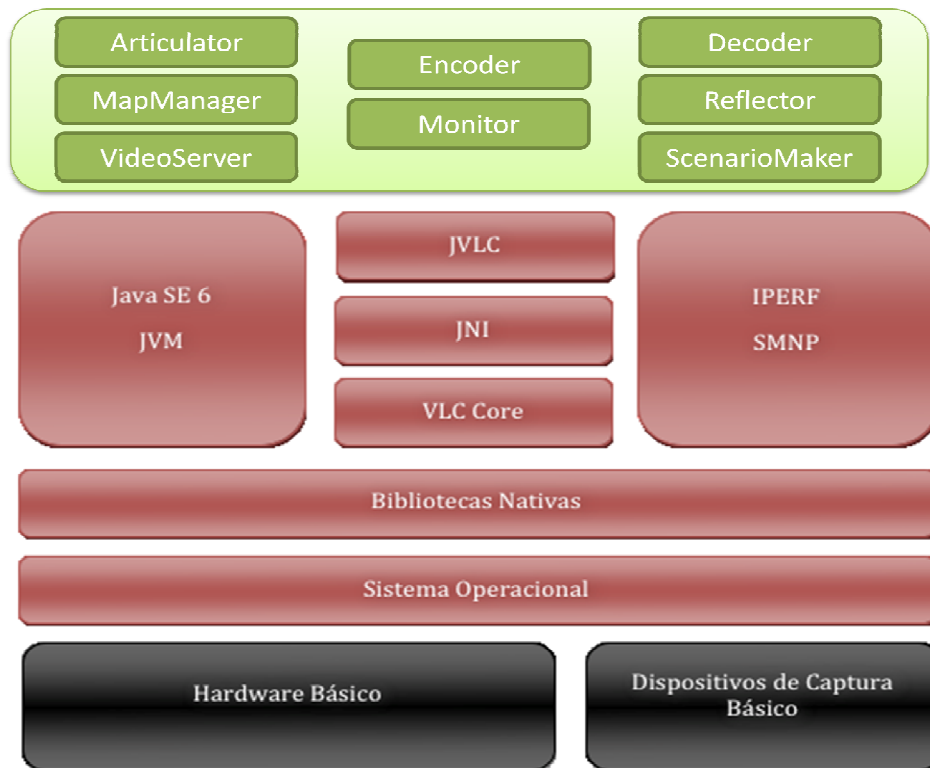


Figura 24 - Pilha de dependência das tecnologias utilizadas pela Arthron.

4.5.1 Java

Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa *Sun Microsystems*. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode" que é executado por uma máquina virtual [39].

Há uma familiaridade da equipe de desenvolvimento de interfaces com essa linguagem de programação e a presença de uma *Application Programming Interface* (API), a *Java Bindings for VideoLAN* (JVLC), permitiu a comunicação do VLC com o Java e por ser multiplataforma fizeram com que o Java fosse escolhido.

4.5.2 VLC

VLC media player é um player multimídia altamente portátil que suporta diversos formatos de áudio e vídeo (H.264, Ogg, DivX, MKV, TS, MPEG-2, mp3, MPEG-4, aac, etc) de arquivos, mídias físicas (DVDs, VCD, Audio-CD), placas de captura de TV e muitos protocolos de streaming.

Ele também converte arquivos de mídia, transcodifica e funciona como um servidor de streaming utilizando unicast ou multicast e IPv4 ou IPv6. Ele não necessita de nenhum codec ou programa externo para funcionar [37].

O VLC foi escolhido por ser livre, de código aberto e multiplataforma permitindo operar com diversos tipos de mídias e dispositivos de captura.

4.5.3 JVLC

O JVLC é uma API que faz associações das funções contidas na biblioteca nativa do VLC, escrita originalmente em C, em chamadas de métodos JAVA utilizando para isso o *Java Native Interface* (JNI). A utilização dessa API facilitou e acelerou o desenvolvimento [40].

4.5.4 Iperf

Iperf foi desenvolvido pela *National Laboratory for Applied Network Research* (NLNR) e *Distributed Applications Support Team* (DAST) como uma moderna alternativa para medir o máximo desempenho de largura de banda via TCP e UDP. Iperf permite o ajuste de vários parâmetros e características UDP. O Iperf gera informações sobre a largura de banda, o atraso médio e sobre quantidade de perdas de pacotes [41].

Ele é código aberto, e roda em diversas plataformas incluindo Linux, Unix e Windows. Utilizamos essa ferramenta para poder analisar e comparar o desempenho da rede alterando os ambientes de testes e podendo assim escolher a melhor configuração possível para execução da performance.

Ele deve ser instalado em todas as máquinas onde se encontram os componentes da ferramenta, pois assim através de uma interface amigável no componente *Articulator* o usuário poderá realizar testes em uma determinada conexão entre os componentes *Encoder*, *Reflector* e *Decoder* na rede para verificar se há perda ou não de datagramas e a partir das informações coletadas, determinar a que taxa de transmissão e que tipo de codificação será mais propícia para o envio de um determinado fluxo de mídia (áudio e vídeo). Assim, o usuário poderá tomar decisões importantes para garantir a transmissão e exibição de fluxo de mídia em um espetáculo.

4.5.5 SNMP

O SMNP é um protocolo de gerência típica de redes TCP/IP, da camada de aplicação, que facilita o intercâmbio de informação entre os dispositivos de rede, como placas e comutadores. O SNMP possibilita aos administradores de rede gerenciar o desempenho da rede, encontrar e resolver seus eventuais problemas, e fornecer informações para o planejamento de sua expansão, dentre outras [42].

Na Arthron ele tem a função de monitoramento remoto dos computadores onde se encontram ativos os componentes *Encoder*, *Decoder* e *Reflector*, obtendo informações como processamento, memória, quantidade de pacotes enviados e recebidos, espaço em disco, etc. Todas essas informações são acessadas em tempo real pelo componente *Articulator*.

5 AMBIENTE DE EXECUÇÃO

Para o funcionamento da Arthron são necessários alguns requisitos de *software*, *hardware* e de rede. Esses requisitos serão discutidos nas seções seguintes.

5.1 SOFTWARE

Os *softwares* necessários para o ambiente de execução da Arthron são compostos por dois Sistemas Operacionais (Linux e Windows) e os pacotes de software necessários instalados em cada SO específico.

- **Sistema Operacional** – para os componentes *Decoder*, *Reflector*, *Manager*, *ScenarioMaker*, *VideoServer*, *Monitor* e *MapManager* é feito o uso da distribuição Linux Ubuntu 10.04 *Desktop*, já o componente *Encoder*, este pode ser executado tanto no SO Windows (XP, Vista, 7) ou no Linux.
- **Pacotes de software** – para a execução de todos os componentes da ferramenta é necessária a instalação dos seguintes pacotes no Linux e Windows: *Java Runtime Environment (JRE)*, *VLC Media Player v1.06* e o *JVLC*, sendo esses três pacotes essenciais para execução da ferramenta, sem eles a mesma não funcionará. Já para as funções de medição e monitoramento será necessária a instalação dos seguintes pacotes no Linux e Windows: *Iperf*, *SNMP*, sem esses pacotes instalados a ferramenta executará sem problemas, porém com as funcionalidades de medição e monitoramento desativadas. Caso algum desses pacotes não estejam instalados a Arthron irá notificar o usuário sobre a falta dos mesmos.

5.2 HARDWARE

Cada componente da ferramenta requer um aparato de *hardware* específico, sendo assim, será apresentado o *hardware* necessário para cada componente da Arthron.

- **Articulator** – para esse componente é necessário o uso de um computador dedicado com a seguinte configuração mínima: processador Core 2 Duo, memória RAM de 3 GBytes, disco rígido de 160 GBytes, monitor LCD de

pelo menos 22 polegadas para visualização da interface e placa de rede Gigabit Ethernet.

- **Encoder** – para esse componente no SO Windows é necessário o uso de uma placa de captura ou placa firewire, quando se está capturando de uma câmera DV ou HDV (*High Definition Digital Video*). Para captura no SO Linux é indicado o uso da placa Hauppauge PVR 350 a partir de câmera DV e Hauppauge HD PVR model 1212 a partir de câmera de alta definição AVCHD (*Advanced Video Codec High Definition*). Para o uso do *Encoder* tanto no Linux quanto no Windows é indicada a seguinte configuração mínima: processador Core 2 Duo 2.1 GHz, memória RAM de 3 GBytes, 160 GBytes de disco rígido e placa de rede Gigabit Ethernet.
- **Decoder** – para esse componente são necessários os seguintes equipamentos: projetor multimídia (quando necessário) acoplado na saída de vídeo do *Decoder* para exibição em telões, computador com a seguinte configuração mínima – processador Core 2 Duo 2.1 GHz, 3 GBytes de RAM, 320 GBytes de disco rígido, placa de vídeo *offboard* (NVIDIA ou ATI – com suporte a decodificação em alta definição) e placa de rede Gigabit Ethernet.
- **Reflector** – para esse componente em máquinas com SO Windows faz necessário o uso de um *Reflector* externo determinado pelo *Articulator*, caso o *Encoder* esteja no Linux não é necessário o uso de um *Reflector* externo, ele executa internamente a inicialização de um *Reflector* interno, porém na interface do *Encoder*, por questões de desempenho da máquina e/ou da rede, pode-se solicitar ao *Articulator* um *Reflector* externo quando necessário. Para a execução desse componente é necessária seguinte configuração mínima: processador Core 2 Duo 2.1 GHz, 2 GBytes de memória RAM, placa de rede Gigabit Ethernet. **Observação:** pode-se rodar até dois *Reflectors* em uma única máquina com as configurações acima, porém deve-se ter em mente a capacidade da rede, visto que, cada componente tem a capacidade de replicar um único fluxo em vários e dependendo do tipo do fluxo, isso sobrecarregará a interface de rede. Por

exemplo, em fluxo em alta definição HDV teríamos 25 Mbps de entrada e se tivéssemos dez destinos de saída teríamos 250 Mbps, somando um total de 275 Mbps com apenas o uso de único *Reflector*.

- **VideoServer** – para esse componente se executarmos apenas a transcodificação do conteúdo para apenas um formato (FLV, OGG ou H.264) é indicada a configuração mínima – Core 2 Duo 2.93 GHz, memória RAM de 4GBytes, 160 GBytes de disco rígido e placa de rede Gigabit Ethernet. Porém se a máquina realizar a transcodificação simultânea de mais de um formato é aconselhável uma máquina mais robusta com a seguinte configuração mínima - Core 2 Quad 2.6GHz, 4GBytes de memória RAM, 160 Gbytes de disco rígido e placa de rede Gigabit Ethernet.
- **MapManager** – para esse componente é indicada a configuração mínima – Core 2 Duo 2.1 GHz, 2 GBytes de memória RAM, 160 Gbytes de disco rígido e placa de rede Fast Ethernet.
- **Monitor** – para esse componente é indicada a configuração mínima – Core 2 Duo 2.1 GHz, 2 GBytes de memória RAM, 160 Gbytes de disco rígido e placa de rede Gigabit Ethernet.
- **ScenarioMaker** – esse componente pode ser executado em paralelo ao componente *Articulator*, ou seja, na mesma máquina, sem que prejudique o desempenho deste.

5.3 REDE

É importante ressaltar que esse tipo de experiência artístico-tecnológica necessita de requisitos críticos em relação à qualidade das mídias manipuladas e aos baixos retardos necessários para sua execução, para obter-se assim um resultado próximo do real. Essa qualidade só pode ser obtida com o uso de uma rede baseada em tecnologia de transmissão óptica, como a rede Ipê. Ela está entre as mais avançadas redes do mundo e possui conexão com redes acadêmicas estrangeiras, tais como Clara (América Latina), Internet2 (Estados Unidos) e Géant (Europa).

6 CENÁRIOS DE APLICAÇÃO

Neste capítulo apresentam-se os possíveis cenários de aplicação da ferramenta, com a identificação do público alvo e os novos cenários para execução da Arthron.

6.1 IDENTIFICAÇÃO DO PÚBLICO ALVO

Inicialmente (GT MDA Fase I) foi proposto um serviço para um público-alvo mais específico – comunidade artística – que trabalha com instalações e espetáculos que envolvam Arte e Tecnologia. Assim, a Arthron prioriza os anseios dessa comunidade que lida com a programação e exibição de fluxos de áudio e vídeo para compor manifestações artísticas. A Arthron é essencial para permitir a manipulação em alto nível dos fluxos nesse contexto.

No entanto, já na etapa de apresentação dos primeiros resultados oriundos do GT MDA durante o WRNP²¹ 2009, realizado em Recife, observou-se que se poderia estender o foco com a utilização ferramenta para transmissão de outros tipos de eventos que envolvam mídias digitais. Nessa ocasião utilizou-se a versão atual da Arthron para transmitir o WRNP internamente uma vez que existiam eventos simultâneos como também para Internet (através de um fluxo de qualidade inferior).

Depois do WRNP, utilizou-se a Arthron para transmissão de outro evento – ENSOL 2009 – evento da comunidade de software livre ocorrido na cidade de João Pessoa-PB na Estação Ciência, Cultura e Artes. Essa última experiência também evidenciou a utilização da Arthron como ferramenta livre para aquisição, controle e distribuição de vídeos ao vivo.

Dessa forma, na Fase II do GT MDA, percebeu-se que é imprescindível que a Arthron ofereça serviços para a comunidade (em especial usuários da RNP) que organiza e participa de eventos de cunho técnico e científico que possa utilizar a Arthron para gerenciar a distribuição das mídias do evento.

Dois diferenciais da Arthron para esse nicho de aplicação:

- Possibilidade de programação de fluxos – nos dias de hoje, os eventos envolvem seções e atividades simultâneas. Através da Arthron além de

²¹ WRNP é o Workshop da Rede Nacional de Ensino e Pesquisa

exibir os fluxos poder-se-ia fazer uma programação interna dos mesmos, viabilizando, por exemplo, a possibilidade de um ponto central (como secretaria do evento) pudesse exibir e controlar imagens ao vivo dos demais pontos do evento como uma mesa de corte em software.

- Possibilidade de manipular fluxos em diferentes formatos – complementando a funcionalidade anterior, a Arthron poderá ser utilizada na exibição de vídeos em alta qualidade para um público interno ou remoto (depende das especificações da rede) e baixa qualidade para o público via Internet, por exemplo.

6.2 CENÁRIOS

Embora, a Arthron tenha nascido com o propósito de suportar eventos artísticos de Arte performática distribuída percebemos que sua utilização é muito mais ampla. Para tanto, pode-se utilizar a mesma arquitetura de software para diferentes cenários de uso.

Os cenários viáveis para aplicação são:

- **Cenário 01:** Transmissão de Eventos Técnico-Científicos.
- **Cenário 02:** Telemedicina.

No Cenário 01: Transmissão de eventos técnico-científicos – pode-se usar a Arthron para capturar, transmitir internamente e externamente (Internet) o evento. Percebeu-se em outros eventos como, por exemplo, o WRNP 2009 o uso de um *hardware* (botoneira), por exemplo, o já citado na seção 3.1 (Grass Valley 3000) para distribuição dos vídeos capturados nas salas de apresentação. Esse equipamento tem custo elevado e a ferramenta poderia substituí-lo de forma eficaz e com custo reduzido, já que utilizar-se software livre e computadores pessoais. Além disso, pode-se facilmente exibir as apresentações de todas as salas do evento na *Internet*.

No Cenário 02: Telemedicina – pode-se usar a Arthron para capturar, transmitir internamente e externamente (*Internet*) uma aula prática ou cirúrgica realizada nas salas de cirurgia, por exemplo, do Hospital Universitário da UFPB. Nesse ponto contribui-se para ampliar a exibição das aulas práticas realizadas em âmbito cirúrgico dentro do próprio hospital através da transmissão para salas de aula do Campus universitário, bem como, em baixa definição para *Internet*.

7 EXPERIÊNCIAS REALIZADAS

Neste capítulo apresentam-se as experiências realizadas com o uso da ferramenta, através da realização de um experimento no contexto artístico, o espetáculo e-Pormundos Afeto e outro experimento no contexto de transmissão de eventos ao vivo, transmissão do WRNP 2010.

7.1 E-PORMUDOS AFETO

Este espetáculo faz uma reflexão sobre o crescimento de dispositivos autômatos na sociedade contemporânea e a distância física que tem sido imposta neste novo cotidiano. O trabalho volta-se para discutir sobre como o afeto, as emoções e as relações interpessoais passaram a ser vivenciadas e administradas nessa cultura do virtual, a qual reposiciona a compreensão sobre a presença e a ausência, sobre o tato e o contato em contrapartida ao visual. Passamos grande parte do nosso tempo em contato com dispositivos digitais que estendem, reduzem, lançam, transportam nossas vidas para outras dimensões.

A tecnologia está cada vez mais móvel e miniaturizada. Vivemos no mundo dos avatares²² do vídeo game, da videoconferência pelo skype ou msn, das comunidades virtuais que ao mesmo tempo que cabem numa tela de computador são do tamanho do mundo (ou maior já que o todo é maior do que a soma de suas partes). São esses redimensionamentos e transformações da vida contemporânea norteadas pela Cultura Digital que o espetáculo de Arte da Telepresença em dança com mediação tecnológica e-Pormundos Afeto pretende abordar. A obra traz os questionamentos sobre as mudanças no nosso comportamento, no nosso entendimento sobre perto e longe, presente e ausente, junto e separado.

7.1.1 Concepção Artística

O experimento proposto teve como foco a articulação de ambientes com dimensões diferentes – chamados aqui de micro-mundo e macro-mundo, ou seja, com seus eventos interferindo um no ambiente do outro, por exemplo: a ação do robô modifica a imagem

²² É uma representação virtual do humano que simula as ações do usuário.

gerada no ambiente do humano/dançarino e vice-versa. O ambiente do macro-mundo (pontos A e C) refere-se ao espaço e a dimensão humana – a dos Dançarinos, Figura 25. O ambiente do micro-mundo (pontos B) está o agente inteligente (robô) que tem uma simulação em ambiente 3D (ponto D). Foi colocada a imagem final recebida pelo usuário da internet como ponto E. Contando com essas dimensões físicas diferenciadas cuja delimitação (paredes/tela) é preenchida por imagens transmitidas em tempo real pelo fluxo de vídeo transmitido pela telemática, ou seja, de um ponto remoto ao outro (veja ilustração em anexo). As imagens existentes no ambiente do Dançarino servirá como estímulo e ignição para sua performance, sendo que essas imagens são geradas pela atuação do robô. Em contrapartida, as ações do Dançarino serão responsáveis pela movimentação do robô no espaço. Desta forma entra-se em uma rede de implicações de um mundo no outro, ou seja, entre o ambiente do robô e o do Dançarino. Além disso, a música é composta a partir da relação entre esses meios, ou seja, as imagens finais geradas pela intersecção entre esses dois ambiente é responsável pela geração do som e ainda conta com as interações dos usuários da internet quando estes interagem com as imagens da simulação 3D.

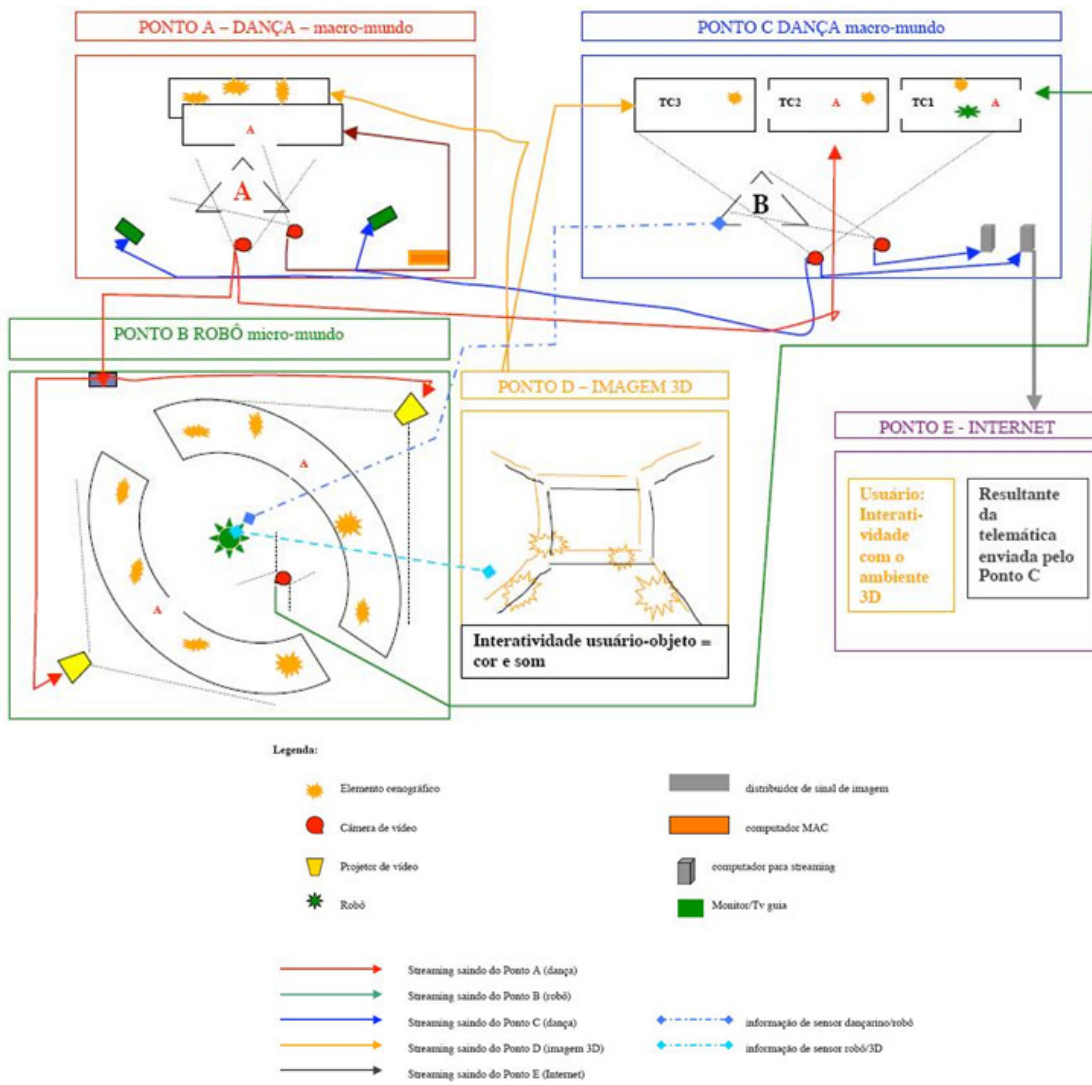


Figura 25 - Mapa esquemático e-Pormundos Afeto.

A configuração cênica de dois pontos envolvidos no espetáculo pode ser observada detalhadamente nas Tabelas 3, 4.

Tabela 3 - Configuração cênica ponto A.

Ponto A (Konic/Espanha)	
Atuante	Dançarino A
Recebimento de streaming	vindo do ponto D simulação 3D do robô vindo do ponto C (câmera geral do Dançarino C para servir como guia)
Envio de streaming	Câmera que capta telas em superposição e envia para ponto B
Telas em superposição	1o. tela tem a imagem desse Dançarino A processada pelo Isadora 2a. tela tem a imagem do streaming do ponto D (simulação 3D do robô)

Tabela 4 - Configuração cênica ponto C.

Ponto C (Fortaleza/CE/Brasil)	
Atuante	Dançarino C
Recebimento de streaming	Tela C1 – vindo do ponto B da câmera que grava robô no espaço dele Tela C2 – Vindo do ponto A (imagem das telas sobrepostas do ponto A com Dançarino e 3D) Tela C3 – streaming do ponto D (simulação 3D do robô)
Envio de streaming	Através da mesa de corte segue para a finalização na internet (ponto E)
Sensor	Através do sensor de “localização” no Dançarino altera as movimentações do robô.
Câmeras	Câmera C1 capta geral do Dançarino C e seu ambiente e: - envia para ponto A (para guia do Dançarino A) - envia para mesa de corte Câmera C3 capta imagens em close da cena e envia para a mesa de corte

7.1.2 Concepção Técnica

Para concepção técnica do espetáculo, o ambiente de transmissão utilizado foi a rede da Empresa de Tecnologia da Informação do Ceará (ETICE²³) que se encontra ligada a rede acadêmica IPÊ da RNP.

Foi utilizado também um aparato tecnológico (*hardware* e *software*) como os já descritos no capítulo 5. A Figura 26 ilustra o cenário resumido da atuação da Arthron dentro do espetáculo.

Além dos pontos A e C, já descritos no item 7.1.1, havia o ambiente de transmissão do robô Galatéia que estava localizado em Natal (ponto B) e que interagiu com o evento através da captura de movimentos de um dispositivo com acelerômetro e conectividade sem fio, para permitir movimentos livres, que estava conectado a uma dançarina que se apresentava no mesmo local em Fortaleza.

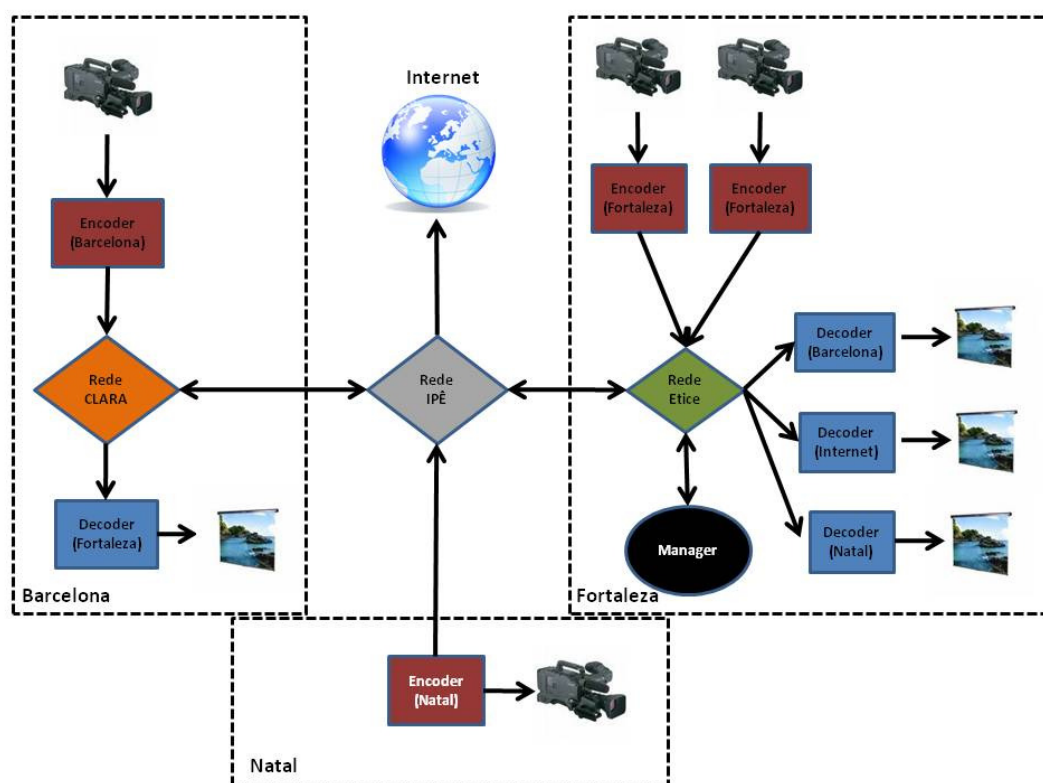


Figura 26 - Cenário de atuação da Arthron.

Ao mesmo tempo, era exibido em Barcelona o Museu Virtual que foi desenvolvido pelo Grupo de Trabalho de Museus Virtuais (GT MV²⁴) do laboratório Natalnet,

²³ <http://www.etice.ce.gov.br/>

localizado na Universidade Federal do Rio Grande do Norte (UFRN), onde os ambientes virtuais relacionados com cada cena representada no espetáculo e um avatar reproduziam os movimentos reais da Galatéia, ligando o virtual ao real.

Devido à diferença de fuso horário de 5 horas entre o Brasil e a Espanha, o show teve que ser realizado duas vezes, para diferentes públicos, um espetáculo em Barcelona às 18:00 horas e outro em Fortaleza às 21:00 horas.

Na Figura 27 está ilustrado o componente *Articulador* controlando todos os fluxos gerados e manipulados durante o evento.

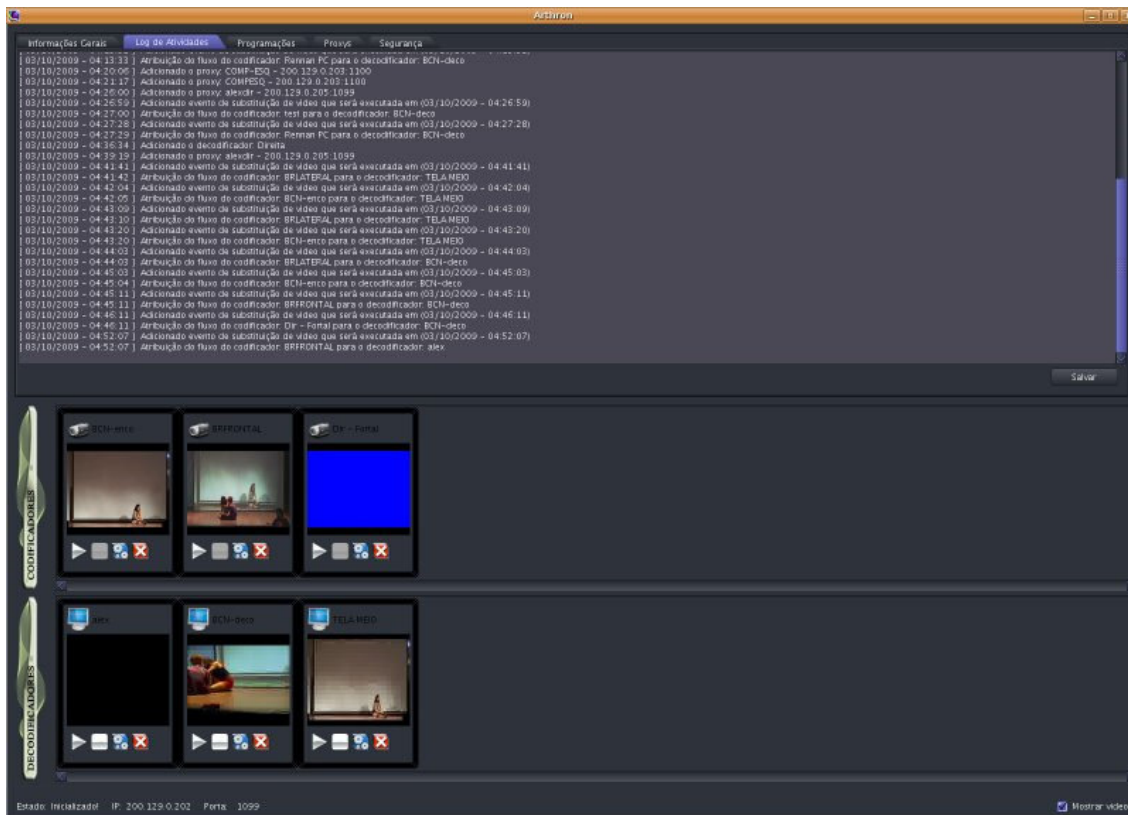


Figura 27 - Tela do Articulador da Arthron controlando o “backstage” do espetáculo e-Pormundos Afeto.

Na Figura 28 pode-se observar uma visão geral de todos os objetos de cena do espetáculo. Em (1) observa-se a dançarina localizada no teatro Dragão do mar em Fortaleza. A tela central que recebe o fluxo de vídeo oriundo de Barcelona onde e possível visualiza a bailarina de Barcelona (3). A tela da esquerda tem-se o fluxo de vídeo gerado a partir do robô Galateia (2) com origem no NatalNet. Na tela da direita observa-se o vídeo (4) que era transmitido para Internet, onde os internautas puderam

²⁴ [http:// www.natalnet.br/gtmv/](http://www.natalnet.br/gtmv/)

acompanhar ao vivo a exibição do espetáculo dentro do portal do GT MDA. Em (5) observa-se o músico que estava localizado em Fortaleza.

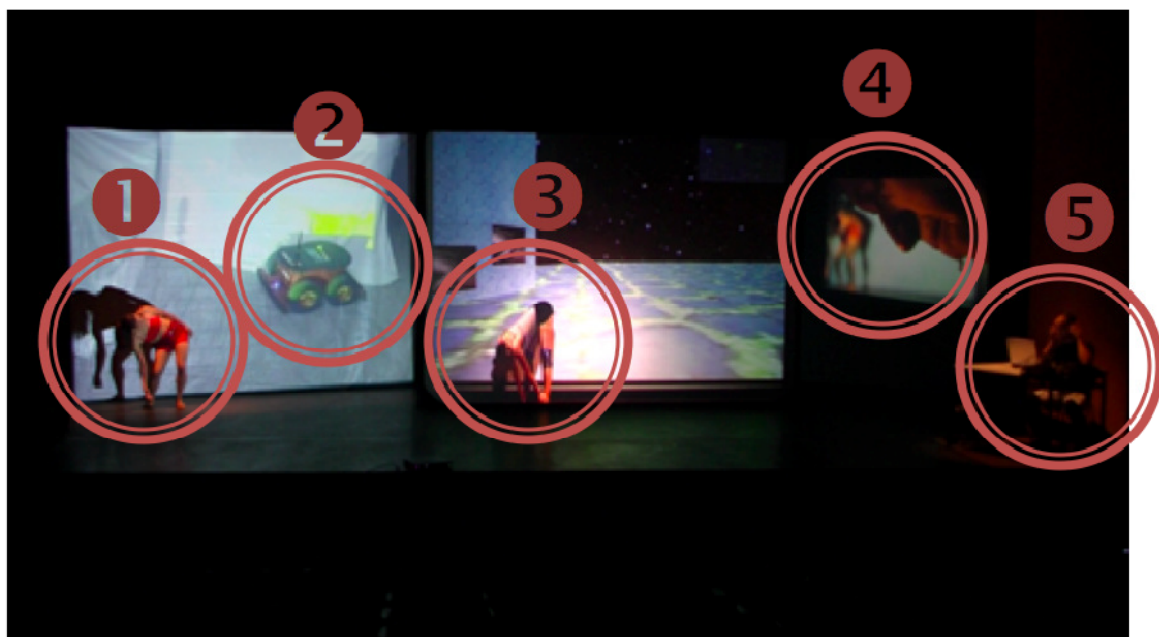


Figura 28 - Elementos do espetáculo e-Pormundos Afeto no palco do Teatro do Dragão do Mar.

7.1.3 Dificuldades e Problemas

A dificuldade encontrada durante a execução da ferramenta no espetáculo foi em relação às diversas trocas de fluxos entre os *Decoders* envolvidos, já que se exigia uma atenção redobrada pelo usuário que manipulava o componente *Articulator*, pois a trocas eram executadas manualmente. Na Tabela 5 pode-se visualizar uma parte do roteiro de câmeras entre Brasil e Espanha da cena 1 do espetáculo.

Dois problemas foram detectados, o primeiro em relação à perda de pacotes da rede e o segundo em relação à parada repentina de um componente *Decoder* durante a execução do espetáculo para o público do Brasil. A perda de pacotes se dava devido ao horário de pico de uso da rede e quando essa perda era significativa o *Decoder* simplesmente parava e algumas soluções foram utilizadas para contornar esses problemas.

7.1.4 Soluções

O problema de parada no funcionamento do componente *Decoder* foi contornado de duas formas distintas. Primeiro, quando eram detectadas perdas no recebimento dos fluxos de mídias, através das medições com o Iperf, era realizada a diminuição na taxa de transmissão a um limite aceitável de qualidade antes da realização dos ensaios e do espetáculo final. Segundo, quando os ensaios e o espetáculo estavam acontecendo e foi detectada a perda de pacotes nos decodificadores, não se podia executar a diminuição da taxa de transmissão, visto que se teria que parar esses fluxos para realizar tais alterações, prejudicando o andamento dos ensaios e do espetáculo. No entanto, quando ocorria alguma parada em algum dos decodificadores era realizado o reinício do módulo defeituoso remotamente, retornando ao seu estado funcional rapidamente de forma que não fosse percebido pelos telespectadores.

7.1.5 Resultados

Dentro do contexto técnico do espetáculo, a Arthron mostrou-se muito eficaz administrando todos os fluxos locais e remotos de áudio e vídeo, e de rápida recuperação quando ocorreram falhas. Além disso, o uso de equipamentos como um comutador de áudio e vídeo profissional não foi necessário para execução desse espetáculo, o que não aconteceu com os Espetáculos “Versus” e “(IN)Toque”, ou seja, os custos finais foram minimizados. Já em relação ao contexto artístico, ela se mostrou também muito importante, pois eram realizadas várias trocas de vídeos entre as diversas telas do espetáculo localizadas entre os diversos pontos, sendo de vital importância para realização do conjunto da obra artística. Além disso, a realização do espetáculo foi uma forma de externar a produção dos GTs da RNP, mostrando como resultados obtidos através da realização de um GT podem ser revertidos em resultados para sociedade através das manifestações artístico-culturais. Por fim, contribui-se também para divulgação dos GTs e da própria RNP atraindo a atenção da imprensa local para cobertura do espetáculo.

Tabela 5 - Roteiro de câmeras Brasil/Espanha.

CENA		OBSERVAÇÕES/ DESCRIÇÕES	BRASIL		ESPANHA	
			CAMERA 1 CENTRAL	CAMERA 2 LATERAL	CAMERA 1	CAMERA 2
CENA 1	Parte 1	Manipulação quadril, Fernando em primeiro plano	Plano geral nas duas telas		<i>?Plano geral com imagem da Carmem e screen (imagem camera robo)</i>	
		Fernando descendo para o palco		Plano geral na tela lateral (c/robô) com fernando entrando em cena	<i>?zoom fechando no corpo da Carmem?</i>	
	Parte 2	Fernando sobre Aline até descerem para o chão	Foco fechado nas costas do Fernando +background (bg) c/Espanha			<i>?Close fechado no corpo Carmem?</i>
		Fernando + Aline (F+A) entrelacados		Close (A+F) + bg c/ carmen em close		<i>?Close fechado no corpo Carmem?</i>
		(F+A) em espiral rodando pelo espaço		Plano geral (F+A) entram e saem de quadro (bg/tela 1) Câmera com ondulação	<i>?Plano geral Carmem + bg imagem 3D?</i>	

7.2 TRANSMISSÃO DO WRNP 2010

Para execução desse experimento foram utilizados 7 computadores e duas filmadoras digitais, Figura 29. Um computador ficou responsável pela execução do componente *Articulator*, e este responsável pelo controle dos outros componentes (*Encoder*, *Decoder*, *VideoServer*, *Reflector*) que estavam nos 6 computadores restantes, dois computadores com execução do componente *Encoder*, este responsável pela captura e transmissão das mídias audiovisuais, estando uma filmadora ligada a um *Encoder* no auditório do WRNP durante os dois primeiros dias do evento e outra ligada a um *Encoder* no stand do GT MDA durante os quatro dias totais do evento.

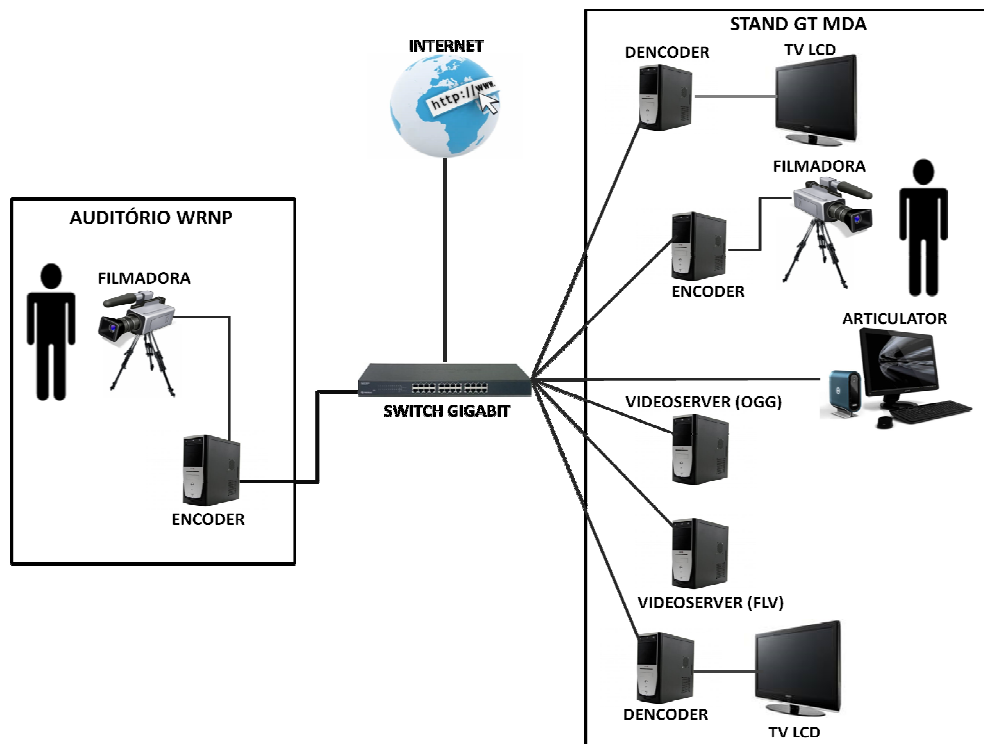


Figura 29 – Cenário de execução do experimento no WRNP 2010.

Dois computadores com a execução do componente *Decoder*, este responsável pela exibição dos conteúdos capturados pelos componentes *Encoder* em duas TVs de LCD no stand do GT MDA, sendo que um desses computadores rodava em paralelo o componente *MapManager*, este responsável pela exibição de uma mapa interativo nos componentes (*Articulator*, *Decoder* e *Encoder*) e os dois restantes pela execução do

componente *VideoServer*, este responsável pela transcodificação dos vídeos capturados pelos *Encoders* para os formatos OGG e FLV que foram publicados pela ferramenta na *Internet* para que as pessoas pudessem assistir e acompanhar o evento ao vivo.

7.2.1 Análise da Transmissão

Em relação à publicação dos conteúdos audiovisuais na *Internet* a medição da quantidade de acesso foi realizada pelo *Google Analytics*, ferramenta esta integrada a funcionalidade de geração de página automática da Arthron, e foram computados durante os quatro dias do evento 392 acessos. Na Figura 30 podemos visualizar parte da página de análise do Google.



Name↑	Reports	Status	Visits	Avg. Time on Site	Bounce Rate	Completed Goals
http://200.132.251.153/wmp2010 UA-15856255-4						
★ 200.132.251.153/wmp2010	View report	✓	392	00:08:04	15.56%	0

Figura 30 – Análise do número de acesso gerado pelo Google Analytics.

Apesar da pouca divulgação da transmissão do GT MDA, observa-se uma relevante quantidade de acessos ao portal de publicação dos conteúdos audiovisuais.

A partir de dados coletados através de formulário gerado pelo *Google Analytics* e das avaliações geradas por 40 usuários na *Internet* obtiveram-se resultados em relação à conectividade e a qualidade de transmissão em relação aos *players* usados para exibição dos conteúdos audiovisuais gerados pela Arthron. Esses resultados serão descritos nas subseções a seguir.

7.2.1.1 Conectividade

Em relação à conectividade dos usuários observa-se na Figura 31 que a maioria possuía uma conexão superior a 1Mbps.

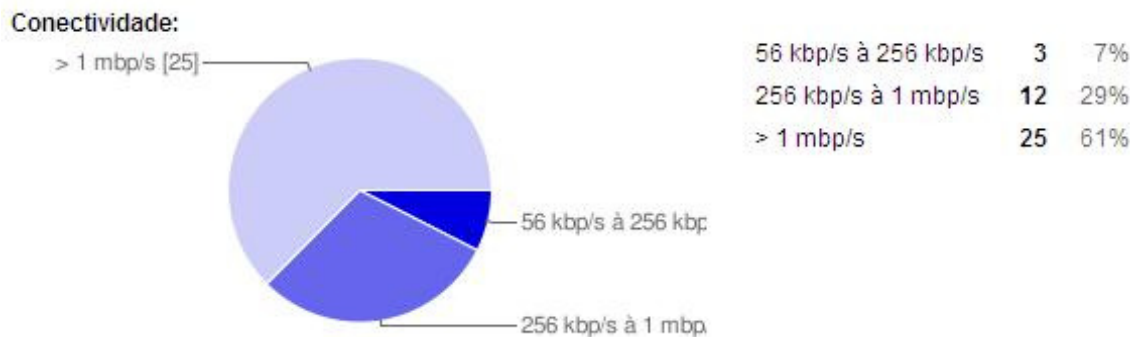


Figura 31 - Gráfico com a descrição da conexão dos usuários.

7.2.1.2 Qualidade da Transmissão

Em relação à qualidade de exibição dos conteúdos audiovisuais pelos *players*, observou-se que a maioria dos usuários aprovou a transmissão no formato FLV com o uso do *player flowplayer*²⁵, Figura 32. Resultado já esperado, pois é um método de encapsulamento de vídeo muito difundido na *Internet* e usado por grandes portais de disponibilização de conteúdos audiovisuais, tais como o *youtube*²⁶. A taxa de transmissão gerada pelo *VideoServer* foi de 700Kbps para o vídeo no formato FLV e 64Kbps para o áudio no formato MP3.



Figura 32 - Gráfico com a análise da qualidade de transmissão com o flowplayer.

²⁵ <http://flowplayer.org/>

²⁶ <http://www.youtube.com/?gl=BR&hl=pt>

O segundo *player* mais bem avaliado em relação à qualidade de transmissão foi o gerado com a nova *tag* do HTML 5²⁷, a *tag* de vídeo, pois entre bom e excelente tivemos 44% de avaliações positivas e 29% não conseguiram assistir, Figura 33. Observa-se também que deveria ter sido avaliado qual o navegador utilizado pelo usuário já que nem todos navegadores atualmente dão suporte ao HTML 5, pois essa porcentagem de 29% poderia ser de um navegador com o *Internet Explorer* que até na sua última versão não tinha suporte ao HTML 5. A taxa de transmissão gerada pelo *VideoServer* foi de 256Kbps para o vídeo no formato THEORA e 64Kbps para o áudio no formato VORBIS.

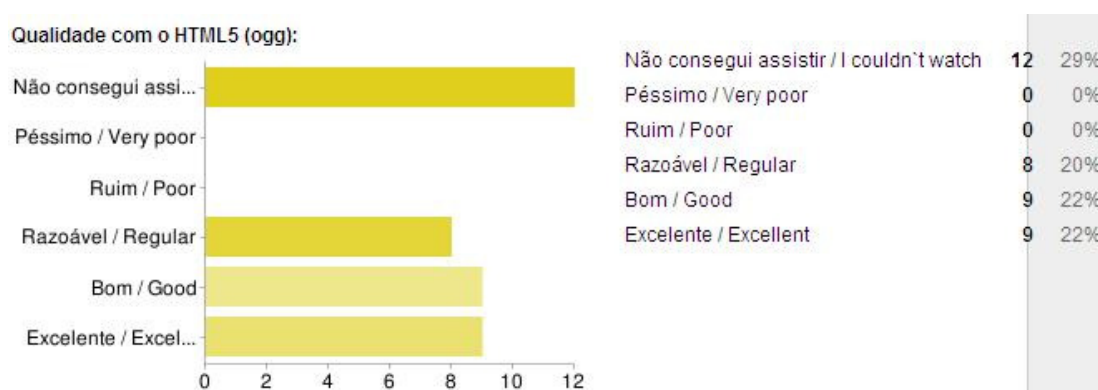


Figura 33 - Gráfico com a análise da qualidade de transmissão com o HTML 5.

Com o *plugin* do *player* VLC para o navegador *Mozilla Firefox* obteve-se o pior desempenho em relação à qualidade de transmissão, Figura 34. 51% dos usuários não conseguiram assistir e apenas 30% avaliaram como bom e excelente. É importante destacar que esse comportamento de 51% (não conseguiram assistir) se deve em grande parte ao usuário, pois o mesmo tem dificuldade no uso e na instalação deste *plugin* para o *Mozilla Firefox*, tanto no Sistema Operacional Linux como no Windows. A taxa de transmissão gerada pelo *VideoServer* foi de 256Kbps para o vídeo no formato THEORA e 64Kbps para o áudio no formato VORBIS.

²⁷ Esta especificação define a quinta revisão importante da linguagem básica da World Wide Web: o Hypertext Markup Language (HTML). Nesta versão, novas características foram introduzidas para ajudar os desenvolvedores de aplicativos Web, tais como: tags canvas, tags de vídeo, geolocalização, caching de aplicações, etc.



Figura 34 - Gráfico com a análise da qualidade de transmissão com o plugin do vlc.

7.2.2 Resultados

A primeira versão da ferramenta (Arthron) desenvolvida na Fase I do GT MDA, apenas contemplava a distribuição de conteúdo audiovisual para *Internet* com a opção do *plugin* do *player* VLC para o navegador *Mozilla Firefox* e foi observado durante a transmissão do conteúdo audiovisual do espetáculo e-Pormundos Afeto na *Internet* que a maioria dos usuários não conseguiu assistir e conseqüentemente não puderam interagir com o espetáculo através do ambiente virtual, pois eles não tinham o *feedback* do que acontecia com o vídeo a partir da interação deles no ambiente. Foi pensando nisso e com as experiências captadas com a realização do espetáculo que foi implementado na Fase II do projeto um novo componente e uma nova funcionalidade do *Articulator* que pudesse suprimir esse problema. O componente foi *VideoServer*, que é capaz de transcodificar os conteúdos capturados de um evento para três formatos (FLV, OGG ou H.264) que são escolhidos pelo usuário da *Internet* através da funcionalidade de geração de página web incorporada no componente *Articulator* da Arthron. Esse componente e essa funcionalidade aumentaram substancialmente a capacidade de divulgação e de interação do usuário na *Internet* junto aos espetáculos e também em relação a outros eventos quaisquer como os já citados.

8 CONSIDERAÇÕES FINAIS

No contexto atual de pesquisa e desenvolvimento em Arte e Tecnologia, ferramentas computacionais para suportar as necessidades de performances midiáticas e distribuídas são cada vez mais necessárias. A Arthron é uma ferramenta que visa contemplar esse cenário apresentando uma solução integrada para gerenciar fluxos midiáticos distribuídos em performances artísticas em tempo real.

O uso da Arthron diminuiu os custos finais de recursos humanos e a complexidade operacional dos experimentos e fez com que fossem aumentadas as possibilidades artísticas.

Em comparação com as ferramentas apresentadas nesse trabalho, na Tabela 1, que foram usadas em experimentos artísticos, a Arthron mostrou-se superior na maioria dos aspectos explorados, como se pode observar na Tabela 6, pois ela conseguiu contemplar mais funcionalidades essenciais para composição de eventos que necessitam de manipulação de mídias em tempo real.

Tabela 6 – Comparação da Arthron com outras ferramentas.

	Suporte TCP/IP	Streaming Áudio	Streaming Vídeo	Gerador de Estatísticas	Sistema Distribuído	Codecs
Arthron	Sim	Sim	Sim	Sim	Sim	Sim
Grass Valley	Não	Sim (apenas local)	Sim (apenas local)	Não	Não	Não
SuperCollider	Sim (apenas áudio)	Sim	Não	Não	Sim	Não
DICE	Não	Não	Sim (apenas local)	Não	Não	Não
INTERACT	Não	Sim (apenas local)	Sim (apenas local)	Sim	Não	Não
SLTV	Sim	Sim	Sim	Não	Sim	Não

No decorrer de dois anos de desenvolvimento contínuo da Arthron, foram publicados vários trabalhos em congressos, simpósios e sessões de tecnologia [29, 30, 38, 43]. Isso demonstra a relevância e importância do trabalho ora apresentado.

No campo da arte e da tecnologia a Arthron apoiou em dois momentos distintos eventos nunca antes realizados no Brasil e na Argentina, através da realização do espetáculo e-Pormundos Afeto, já apresentado neste trabalho.

8.1 TRABALHOS FUTUROS

São diversos os caminhos abertos para intensificação dessa pesquisa. Igualmente, foi descrita a especificação e criação de uma ferramenta que garante um processo sistêmico para o provimento da infraestrutura tecnológica que forneça o arcabouço ferramental imprescindível para apoiar a realização desse tipo de manifestação artístico cultural como um projeto já em andamento no GT MDA.

Na primeira fase do desenvolvimento da Arthron o foco foi na manipulação dos fluxos de mídias. Também foram contempladas outras funcionalidades já descritas nesse trabalho, tais como monitoramento e medição entre os pontos envolvidos de um espetáculo.

Durante a realização do espetáculo “e-Pormundos Afeto” o GT MDA teve a oportunidade de validar a proposta atual como também pensar em novas oportunidades para a Arthron, que foram desenvolvidas e colocadas em prática durante o WRNP 2010.

Outros cenários são promissores como possíveis uso para a ferramenta, como o de Telemedicina, já citado no na subseção 6.2 e para o contexto de TV como é o caso da TV da UFBA, que já se encontra em testes com Arthron e que levou ao início da implementação de uma nova funcionalidade para ferramenta, a inserção de *playlist* de conteúdo audiovisual.

9 REFERÊNCIAS

- [1] RNP (Brasil). **Internet2**. Disponível em: <<http://www.rnp.br/redes/internet2.html>>. Acesso em: 29 jan. 2010.
- [2] RNP (Brasil). **Projeto Giga**. Disponível em: <<http://www.rnp.br/pd/giga/>>. Acesso em: 29 jan. 2010.
- [3] RNP (Brasil). **GT Vídeo digital**. Disponível em: <<http://www.rnp.br/pd/gts2002-2003/gt-vd.html>>. Acesso em: 29 jan. 2010.
- [4] RNP (Brasil). **GT Vídeo digital 2**. Disponível em: <<http://www.rnp.br/pd/gts2003-2004/gt-vd2.html>>. Acesso em: 29 jan. 2010.
- [5] RNP (Brasil). **GT TV digital**. Disponível em: <<http://www.rnp.br/pd/gts2005-2006/tvdigital.html>>. Acesso em: 29 jan. 2010.
- [6] RNP (Brasil). **GT TV digital 2**. Disponível em: <http://www.rnp.br/pd/gts2006-2007/GT_TV_Digital.html>. Acesso em: 29 jan. 2010.
- [7] RNP (Brasil). **GIGA VR – Plataformas para o Desenvolvimento de Aplicações de Realidade Virtual Imersiva e Distribuída sobre Redes de Altíssima Velocidade**. Disponível em: <<http://indico.rnp.br/getFile.py/access?contribId=11&resId=0&materialId=0&confId=33>>. Acesso em: 29 jan. 2010.
- [8] BRASIL. Ministério da Cultura. Cultura Digital. **Ministro de Ciência e Tecnologia lança nova rede multigigabit da RNP**. 3ª Conferência Nacional de Ciência, Tecnologia e Inovação. Disponível em: <<http://www.cultura.gov.br/site/2005/11/21/ministro-de-ciencia-e-tecnologia-lanca-nova-rede-multigigabit-da-rnp/>>. Acesso em: 30 jan. 2010.
- [9] RNP (Brasil). Sala de Imprensa. **Espetáculo interativo de dança será apresentado em workshop de redes avançadas**. 9º WRNP. Disponível em: <<http://www.rnp.br/imprensa/2008/rel-080521.html>>. Acesso em: 30 jan. 2010.
- [10] AUGUSTO, Sergio. **Arte e Tecnologia**: Um pouco da história da fusão entre Arte e Tecnologia. Disponível em: <http://www.ateliogavetas.jex.com.br/noticias/detalhes.php?id_jornal=10046&id_noticia=2>. Acesso em: 29 jan. 2010.

- [11] MACHADO, Arlindo. **Tecnologia e arte contemporânea**: como politizar o debate. 2005, Revista de Estudios Sociales, Vol. 22, pp. 71-79.
- [12] BERNARDINO, Paulo. Aveiro, Portugal. **O Hábito da Imagem**. Representação e Tecnologia na Arte. : Universidade de Aveiro, 2004. 4 SOPCOM. pp. 83-93.
- [13] SANTANA, Ivani. **A imagem do corpo através das metáforas (ocultas) na dança-tecnologia**. Belo Horizonte - Minas Gerais - Brasil. : s.n., 2003. INTERCOM – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação. p. 17.
- [14] CHAPMAN, Nigel e Chapman, Jenny. **Digital Multimedia**. s.l. : John Wiley & Sons, 2000.
- [15] RIBEIRO, N.M., GOUVEIA, L. B., **Proposta de um Modelo de Referência para as Tecnologias Multimédia**, Revista da Faculdade de Ciência e Tecnologia, Nº 1, pp. 109-115, Edições Universidade Fernando Pessoa, Porto, Portugal, 2004, ISSN: 1646-0499.
- [16] SOARES, Luiz Fernando Gomes, RODRIGUES, Rogério Ferreira e MORENO, Márcio Ferreira. 2007. **Ginga-NCL**: the declarative Environment of the Brazilian Digital TV System. Journal of the Brazilian Computer Society. 2007, Vol. v12, pp. 37-46.
- [17] NETWORK WORKING GROUP. **Real Time Streaming Protocol (RTSP)**. Disponível em: <<http://www.ietf.org/rfc/rfc2326.txt>>. Acesso em: 03 fev. 2010.
- [18] NETWORK WORKING GROUP. **RTP: A Transport Protocol for Real-Time Applications**. Disponível em: <<http://tools.ietf.org/pdf/rfc1889.pdf>>. Acesso em: 03 fev. 2010.
- [19] NETWORK WORKING GROUP. **SDP: Session Description Protocol**. Disponível em: <<http://tools.ietf.org/pdf/rfc4566.pdf>>. Acesso em: 03 fev. 2010.
- [20] A. Yarali and A. Cherry. **Internet protocol television (IPTV)**. TENCON 2005. 2005 IEEE Region 10, November 2005, pp. 1 – 6.
- [21] THE INTERACTIVE TELEVISION DICTIONARY & BUSINESS INDEX (USA). **Web TV**. Disponível em: <http://www.itvdictionary.com/web_tv.html>. Acesso em: 08 fev. 2010.

- [22] DULTRA, Maruzia & SANTANA, Ivani. **Um olhar que dança:** a experiência telemática ‘Versus’ do ponto de vista da Câmera. In: IX Congresso Brasileiro de Ciências da Comunicação da Região Nordeste – Salvador – BA.
- [23] COUCHOT, Edmond. **A tecnologia na arte:** da fotografia à realidade virtual. Porto Alegre: UFRGS, 2003.
- [24] BASBAUM, Sérgio Roclaw. **Sinestesia, Arte e Tecnologia:** fundamentos da cromossonia. Selo Universidade: 2002.
- [25] **Information Overload:** A Collaborative Dance Performance. Mandilian, Lauren, Diefenbach, Paul e Kim, Youngmoo. Vancouver : ACM Press, 2008. SAME’08.
- [26] **Mixing Dance Realities:** Collaborative Development of Live-Motion Capture In a Performing Arts Environment. MEADOR, SCOTT W, et al. 2004. ACM Computers in Entertainment. p. 15.
- [27] **Movement-based Interactive Dance Performance.** James, Jodi, et al. Santa Barbara, California, USA : ACM, 2006. MM’06.
- [28] **An In-depth Case Study of Art-Technology Collaboration.** Zhang, Yung e Candy, Linda. Washington, DC, USA. : ACM, 2007. C&C’07. p. 10.
- [29] MELO, E. A. G. ; PINTO, A. A. ; SILVA, J. C. F. ; TOSCANO, R. N. ; MARIZ, D. ; TAVARES, T. A. ; SOUZA FILHO, G. L. . Arthron: Uma Ferramenta para Performances Artístico-Midiáticas Distribuídas. In: Simpósio Brasileiro de Sistemas Multimídia e Web (Webmedia), 2009, Fortaleza.
- [30] MELO, E. A. G. ; PINTO, A. A. ; SILVA, J. C. F. ; TOSCANO, R. N. ; MARIZ, D. ; TAVARES, T. A. ; SOUZA FILHO, G. L. . Arte e Tecnologia: Lições Aprendidas com a Realização de Performances Artísticas baseadas na Distribuição de Conteúdo Multimídia. In: XXXV Conferencia Latinoamericana de Informática (CLEI2009), 2009, Pelotas.
- [31] THOMPSON, John Henry. **JHT Other work.** Disponível em:
 <<http://www.j4u2.com/jht/newwork.html>>. Acesso em: 12 fev. 2010.
- [32] GRASS VALLEY (França). **GVG 3000 User Manual.** Disponível em:
 <http://www.grassvalley.com/docs/Manuals/DigitalSwitcher/0159_00.PDF>. Acesso em: 12 fev. 2010.

- [33] MCCARTNEY, James. **SuperCollider**. Disponível em:
<<http://supercollider.sourceforge.net/>>. Acesso em: 13 fev. 2010.
- [34] CENTER FOR NEW MUSIC AND AUDIO TECHNOLOGY (CNMAT). **Open Sound Control**: Introduction to OSC. Disponível em:
<<http://opensoundcontrol.org/introduction-osc>>. Acesso em: 13 fev. 2010.
- [35] MANGOLD INTERNATIONAL (USA). **INTERACT**. Disponível em:
<<http://www.mangold-international.com/en/products/interact.html>>. Acesso em: 13 fev. 2010.
- [36] PROJETO SLTV. Disponível em: <<http://br.gnome.org/TV/ProjetoSLTV>>. Acesso em 17 mai. 2010.
- [37] VLC MEDIA PLAYER. **VLC**. Disponível em: <<http://www.videolan.org/vlc/>>. Acesso em: 15 fev. 2010.
- [38] MELO, E. A. G. ; PINTO, A. A. ; SILVA, J. C. F. ; TOSCANO, R. N. ; BRANCO, A. C. ; VIEIRA FILHO, E. . ARTHRON 1.0: Uma Ferramenta para transmissão e gerenciamento remoto de fluxos de mídia. In: XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2010, Gramado/RS.
- [39] JAVA. Disponível em <<http://www.sun.com/>>. Acesso em: 13 fev. 2010.
- [40] JVLIC - CORE. **JVLIC**. Disponível em: <<http://jvlc.ihack.it/maven-site/jvlc-core/project-info.html>>. Acesso em: 15 fev. 2010.
- [41] NLANR/DAST. **Iperf**. Disponível em: <<http://iperf.sourceforge.net/>>. Acesso em: 15 fev. 2010.
- [42] NETWORK WORKING GROUP. **A Simple Network Management Protocol (SNMP)**. Disponível em: <<http://tools.ietf.org/html/rfc1157>>. Acesso em: 15 fev. 2010.
- [43] MELO, E. A. G. ; SOUZA FILHO, G. L. ; TAVARES, T. A. . Arthron A Tool for Video Streaming Remote Management in Artistic Performances Experiences. In: Fall 2010 Internet2 Member Meeting Session, 2010, Atlanta, Georgia.