



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CONCILIANDO COERÊNCIA E RESPONSABILIDADE EM STORYTELLING  
INTERATIVO PARA TV DIGITAL

Marcelo de Mello Camanho

**Orientador**  
Prof. Dr. Angelo Ernani Maia Ciarlini

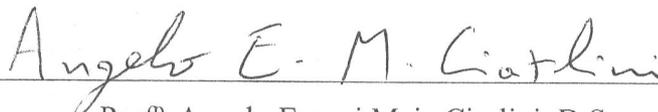
RIO DE JANEIRO, RJ – BRASIL  
ABRIL DE 2009

CONCILIANDO COERÊNCIA E RESPONSABILIDADE EM STORYTELLING  
INTERATIVO PARA TV DIGITAL

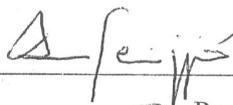
Marcelo de Mello Camanho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM INFORMÁTICA.

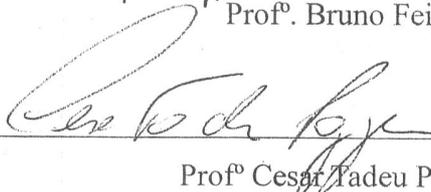
Aprovada por:



Profº. Angelo Ernani Maia Ciarlini, D.Sc.



Profº. Bruno Feijó D.Sc.



Profº Cesar Tadeu Pozzer D.Sc.



Profº Sean W. M. Siqueira D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2009

C172 Camanho, Marcelo de Mello.  
Conciliando coerência e responsividade em storytelling interativo para TV digital / Marcelo de Mello Camanho, 2009.  
xi, 113f.

Orientador: Angelo Ernani Maia Ciarlini.  
Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

1. Sistemas multimídia. 2. Storytelling interativo. 3. Inteligência artificial. 4. Televisão digital. 5. Sistemas de informação. 6. Execução, geração e formação de planos. I. Ciarlini, Angelo Ernani Maia. II. Universidade Federal do Estado do Rio de Janeiro (2003-). Centro de Ciências Exatas e Tecnologia. Curso de Mestrado em Informática. III. Título.

CDD – 006.7

## **Agradecimentos**

Agradeço à minha família; meus pais; minha namorada; meus amigos e meus professores, em especial a meu orientador, por toda a ajuda que me deram ao longo destes anos. Para evitar esquecer alguém, não citarei nomes. Também agradeço à CAPES pela ajuda financeira.

Resumo da Dissertação apresentada ao PPGI/UNIRIO como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CONCILIANDO COERÊNCIA E RESPONSABILIDADE EM STORYTELLING  
INTERATIVO PARA TV DIGITAL

Marcelo de Mello Camanho

Abril/2009

Orientador: Angelo Ernani Maia Ciarlini

A televisão interativa é um novo meio onde muitas formas de entretenimento serão possíveis, juntando interatividade e conteúdo multimídia digital. Uma possibilidade de entretenimento digital promissora para este novo meio é o uso de sistemas de *storytelling* interativo, que permitem a geração, visualização e controle de histórias interativas. O principal desafio para esses sistemas é gerar histórias coerentes enquanto os usuários as assistem e interferem no que está acontecendo. Em um ambiente de TV, a manutenção da responsividade é também essencial, porque o público está acostumado a ter suas expectativas atendidas com rapidez, sem que isso comprometa a qualidade do serviço. Neste trabalho é apresentado um modelo de *storytelling* interativo para atender estas demandas, usando como base o sistema Logtell. O Logtell é um sistema já existente que é focado na simulação baseada em lógica das histórias. A extensão do modelo original tem por intuito atender aos requisitos adicionais de um ambiente de TV interativa.

Abstract of Dissertation presented to PPGI/UNIRIO as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CONCILIATING COHERENCE AND RESPONSIVENESS IN INTERACTIVE  
STORYTELLING FOR DIGITAL TV

Marcelo de Mello Camanho

April/2009

Advisor: Angelo Ernani Maia Ciarlini

Interactive television is a new medium in which many new forms of entertainment will be possible, bringing together interactivity and digital multimedia content. One of the promising alternatives for digital entertainment for this new medium is the use of interactive storytelling systems, which allow generation, visualization and control of interactive stories. The main challenge to these systems is the generation of coherent stories while users are watching and interfering with what is happening. In a TV environment, keeping the responsiveness is also essential, because the audience is used to have their expectations fulfilled without compromising the quality of service. In this work, an interactive storytelling model is presented to meet those demands, based on the Logtell system. Logtell is an already implemented system which is focused on the logic-based simulation of stories. The extension of the original model was intended to cope with the additional requirements of an interactive TV environment.

## Sumário

1. Introdução.....	1
1.1 Motivação.....	1
1.2 Proposta.....	2
1.3 Protótipo.....	3
1.4 Estrutura.....	4
2. Storytelling Interativo.....	5
2.1 Modelos de Geração de Histórias.....	7
2.1.1 Abordagens Centradas em Personagens (Character Based).....	8
2.1.2 Abordagens Centradas na Trama (Plot Based).....	9
2.1.3 Abordagens Híbridas e Outras Abordagens .....	12
2.2 Interactive Storytelling e Planejamento Automatizado.....	16
2.3 Conclusões.....	18
3. Televisão Interativa .....	21
3.1 Interatividade e Televisão.....	23
3.2 Aplicações e Serviços na Televisão Interativa .....	24
3.3 TV Digital Interativa.....	27
3.3.1 Transmissão de Dados.....	27
3.3.2 Middleware.....	28
3.4 Outros Ambientes e Abordagens para TV Interativa.....	29
3.5 Conclusão.....	31
4. Logtell.....	33
4.1 Geração de histórias.....	34
4.2 Interação com o Usuário.....	40
4.3 Dramatização.....	41
4.4 Demais Aspectos.....	44
4.5 Conclusões.....	45
5. Modelo de Storytelling Interativo para TV.....	47
5.1 Requisitos.....	47
5.1.1 Fluxo Contínuo.....	47
5.1.2 Diversidade de Enredos.....	49
5.1.3 Coexistência com Modo Passo-a-passo.....	50
5.1.4 Facilidade de Interação e Possibilidades de Interação em Diversos Níveis.....	51
5.1.5 Escalabilidade.....	52
5.2 Arquitetura Cliente-Servidor.....	52
5.3 Interação Contínua.....	56
5.3.1 Intervenções Fracas: Rewind e Another.....	57
5.3.2 Intervenções Fortes.....	58
5.4 Interação com Múltiplos Usuários.....	59
5.5 Estratégias para Manutenção do Fluxo.....	60
5.6 Mecanismos Avançados de Interação.....	61
5.7 Conclusões.....	62
6. Implementação.....	63
6.1 Ambiente da Aplicação.....	63
6.2 Simulation Controller.....	67

6.3 User Interface.....	72
6.4 Interface Controller.....	76
6.5. Conclusões.....	77
7. Aplicação do Protótipo.....	79
7.1 Continuidade do Fluxo.....	80
7.2 Diversidade e Coerência das Histórias.....	84
7.3 Facilidade e Comodidade da Utilização.....	88
7.4 Escalabilidade.....	89
8. Conclusões e Trabalhos Futuros.....	91
8.1 Considerações Gerais.....	91
8.2 Principais Contribuições.....	91
8.3 Trabalhos Futuros.....	93
Referências Bibliográficas.....	96
Apêndice A – Arquiteturas e Padrões Para a TV Digital Interativa.....	105

## Lista de Figuras

Figura 2.1: Dramatização de algumas histórias de (CAVAZZA et al., 2002).....	9
Figura 2.2: Teatrix em ação.....	10
Figura 2.3: Arquitetura de (GRASBON e BRAUN, 2001).....	11
Figura 2.4: Arquitetura de (SPIERLING et al., 2002).....	12
Figura 2.5: Façade e seus dois atores na visão 3D do usuário.....	14
Figura 2.6: Arquitetura proposta por (SZILAS, 2003).....	14
Figura 2.7: Protótipo de (SZILAS, 2008), modo texto.....	15
Figura 2.8: Uma HTN do protagonista de CAVAZZA et al., 2002.....	16
Figura 2.9: Framework de (SI et al., 2008).....	17
Figura 4.1: Arquitetura inicial do Logtell.....	34
Figura 4.2: Interação passo a passo.....	41
Figura 4.3: História sendo dramatizada no Logtell.....	42
Figura 5.1: Nova arquitetura do Logtell no modelo proposto.....	53
Figura 5.2: Interação contínua.....	57
Figura 6.1: Beans J2EE do protótipo.....	67
Figura 6.2: Métodos principais implementados no SimulationControllerBean.....	68
Figura 6.3: Submódulos do Simulation Controller.....	71
Figura 6.4: Interface inicial do Logtell.....	72
Figura 6.5: Escolha de contexto.....	72
Figura 6.6: Novas classes para a interface com o usuário.....	73
Figura 6.7: Interface do modo passo-a-passo.....	74
Figura 6.8: Interface do modo contínuo.....	75
Figura 6.9: InterfaceControllerBean.....	76
Figura 7.1 Modo multiusuário com dois clientes assistindo à mesma história.....	90

## Lista de Tabelas

Tabela 4.1: Predicados usados no contexto exemplo do Logtell e suas descrições.....	37
Tabela 4.2: Lista de Eventos possíveis no contexto exemplo do Logtell.....	38
Tabela 4.3: Lista de regras no exemplo do Logtell.....	39
Tabela 7.1: Exemplo de narrativa sem interferências.....	81
Tabela 7.2: Alguns tempos para o Rewind.....	82
Tabela 7.3: Alguns tempos para o Another.....	82

## Lista de Abreviaturas

<b>API</b>	Application Programming Interface
<b>ATSC</b>	Advanced Television Systems Committee
<b>DAO</b>	Data Access Object
<b>DLL</b>	Dynamic-link library
<b>DTV</b>	Digital Television
<b>DVB</b>	Digital Video Broadcasting
<b>EJB</b>	Enterprise Java Bean
<b>HDTV</b>	High Definition TV
<b>HTN</b>	Hierarchical Task Network
<b>IPG</b>	Interactive Plot Generator
<b>ISDB</b>	Integrated Service Digital Broadcasting
<b>ISDB-TB</b>	Integrated Services Digital Broadcasting – Terrestrial (do Brasil)
<b>ITV</b>	Interactive Television
<b>JDBC</b>	Java Database Connectivity
<b>JDK</b>	Java Development Kit
<b>JMS</b>	Java Messaging Service
<b>JNDI</b>	Java Naming and Directory Interface
<b>JSP</b>	Java Server Page
<b>MPEG</b>	Moving Picture Experts Group
<b>NPC</b>	Non Player Character
<b>POJO</b>	Plain Old Java Object
<b>RPG</b>	Role Playing Game
<b>SBTVD</b>	Sistema Brasileiro de Televisão Digital
<b>SDTV</b>	Standard Definition TV
<b>UHDV</b>	Ultra High Definition Video
<b>URD</b>	Unidades Receptora-decodificadora

# 1. Introdução

## 1.1 Motivação

Sistemas de *storytelling* interativo são, resumidamente, aplicações que “contam” histórias, que, com base no uso mecanismos automatizados, procuram adaptar as histórias de acordo com as interações que têm com os usuários. São diferentes de jogos porque se focam nas histórias e no seu conteúdo dramático, em vez de em desafios pontuais intercalados com trechos de histórias pré-escritas.

Sistemas de *storytelling* interativo têm sido pesquisados para diferentes aplicações e implementados de diferentes modos. Podem ser aplicados a jogos, auxílio à autoria de histórias e a entretenimento em geral. Uma das dificuldades mais comuns ao se criar um sistema de *storytelling* interativo é fornecer um bom nível de interatividade e cativar o usuário, enquanto se mantém a coerência da história que é gerada.

O advento da TV interativa, seja por meio da TV digital aberta ou de outros ambientes, cria uma oportunidade interessante para a aplicação de técnicas de *storytelling* interativo, pois, se a TV já era um meio clássico para a veiculação de histórias, é de se esperar que, ao se adicionarem mecanismos de interação, consigam-se produzir bons produtos. Nesse ambiente, alguns requisitos, tais como qualidade, coerência e diversidade das histórias, e facilidade e comodidade da interação, tendem a se tornar ainda mais fortes.

O prazer do usuário, ao assistir uma história na TV, está ligado à qualidade da história. Como a coerência do enredo é fundamental para essa qualidade, torna-se ainda essencial que as oportunidades de interação não a comprometam. Além disso, as histórias precisam ser variadas e capazes de surpreender para que o usuário não se canse de assistir a elas.

Para o usuário interagir com as histórias, é preciso encontrar uma forma híbrida de apresentação de conteúdo multimídia, misturando características de jogos e televisão normal. É necessário manter o apelo para o espectador comum, e, ao mesmo tempo, permitir que o usuário seja capaz de interagir de diferentes maneiras com o conteúdo que lhe é apresentado.

Deve-se, inclusive, considerar o fato de que o usuário pode nem mesmo querer interagir de modo algum. A sensação de conforto na interação é fundamental, especialmente quando se considera que as histórias, em geral, serão assistidas por espectadores que podem não ser ávidos usuários de jogos eletrônicos.

Mídias como a TV demandam alta responsividade, ou seja, a satisfação rápida das expectativas dos usuários sem que isso comprometa a qualidade do serviço. Quando se assiste à TV, espera-se que o conteúdo apresentado não sofra interrupções ao longo do tempo. Além disso, programas de TV tendem a ser acompanhados por uma grande quantidade de espectadores. Dessa forma, requisitos de responsividade adicionais relacionados à continuidade do fluxo de apresentação e à escalabilidade precisam ser conciliados com os demais requisitos.

Para garantir a coerência, a qualidade e a diversidade das histórias em conjunto com boas oportunidades de interação, não se pode depender apenas de um esforço autoral capaz de delinear todas as possibilidades de enredo, pois os custos tendem a se tornar facilmente proibitivos. Além disso, as opções de intervenção do usuário ficam mais restritas nesse caso. Uma alternativa é prover mecanismos automáticos de criação de enredos, dando maior liberdade para o usuário interagir e adaptando-se automaticamente os enredos às intervenções que ocorrem. No entanto, esse processamento pode ser custoso e, se não houver uma boa estratégia de coordenação, o fluxo de apresentação do conteúdo pode ter que ser interrompido, o que certamente compromete a responsividade esperada em um ambiente de TV.

A motivação deste trabalho é a proposição de um modelo de *storytelling* interativo para TV onde seja possível conciliar coerência e responsividade e que, ao mesmo tempo, facilite a obtenção automática de histórias variadas, com as quais o usuário pode interagir de forma cômoda.

## **1.2 Proposta**

O modelo de *storytelling* interativo para TV proposto nesta dissertação tem como base o modelo adotado no sistema Logtell, o qual foi estendido para atender aos requisitos de um ambiente de TV interativa.

O Logtell (CIARLINI *et al.* 2005; POZZER 2005) é um sistema de *storytelling* interativo que vem sendo desenvolvido já há algum tempo. A sua abordagem parece ser boa para TV, pois se centra em manter as histórias coerentes enquanto permite que o usuário possa interferir na história em diferentes níveis. Para garantir a coerência das histórias, a interação no Logtell é sempre indireta. O usuário pode escolher não interferir na história, solicitar alternativas ou ter uma participação mais forte, estabelecendo, por exemplo, que certos eventos ou situações ocorram, desde que sejam logicamente coerentes com o modelo especificado para o gênero de história adotado. A dramatização

da história não ocorre em tempo real em relação à geração do enredo, podendo ser ativada após parte do enredo já ter sido criado, ou quando já estiver completo. A dramatização é feita com atores tridimensionais, usando um motor gráfico próprio (POZZER 2005), que foi criado para garantir a consistência do modelo lógico dos enredos com a dramatização.

Uma das principais características da primeira versão do Logtell, que não é apropriada para a televisão interativa, é que, nele, a geração e a dramatização da história são separadas, concentrando-se a interatividade na fase de geração. Parece pouco razoável que o usuário de TV aceite aguardar o fim de uma fase de simulação para assistir à dramatização. Além disso, deve-se ressaltar que, embora possibilite ao usuário a intervenção em graus variados de intensidade, os meios de interação no Logtell não têm o nível de comodidade que se espera de uma aplicação para TV interativa.

A adaptação de um modelo de *storytelling* interativo como o do Logtell a um ambiente de TV interativa demanda que se encontrem meios para que a geração do enredo, a interação com o usuário e a dramatização ocorram em paralelo. Nesse contexto, as oportunidades de interação viáveis devem ser cuidadosamente estudadas, de modo a permitir interações em diferentes níveis na evolução da história, mas de forma confortável para o usuário. Além disso, a coordenação desses processos deve levar em conta os requisitos de responsividade, que são próprios do ambiente, como a continuidade do fluxo e a escalabilidade. O novo modelo proposto procura levar em conta todas essas questões, dando ênfase à conciliação da coerência das histórias com a responsividade.

### **1.3 Protótipo**

Para a validação das principais idéias do novo modelo, um novo protótipo do Logtell foi implementado. As modificações no sistema foram feitas na parte do sistema correspondente ao módulo *Plot Manager* que coordenava a interação com o usuário e a ativação dos módulos de geração de enredos e de dramatização. O *Plot Manager* foi desmembrado em módulos em arquitetura distribuída, com o controle e a geração das histórias ocorrendo no servidor e a interface com o usuário e a dramatização sendo executados nos clientes.

No protótipo que foi desenvolvido foram mantidas as funcionalidades pré-existentes do Logtell, e a ela adicionaram-se duas novas maneiras de gerar histórias: o modo contínuo em multiusuário e também no modo de apenas um usuário. Desta forma,

o protótipo é capaz de mostrar novas maneiras de se ter uma experiência de *storytelling* interativo, usando uma arquitetura, como será mostrado nos capítulos adiante, que é distribuída, para dar suporte ao modelo proposto.

Mais detalhes sobre a implementação do protótipo estão presentes nos Capítulos 6 e 7.

#### **1.4 Estrutura**

No capítulo 2, apresenta-se uma visão de geral da pesquisa em *storytelling* interativo descrevendo-se as principais questões envolvidas e os enfoques que vêm sendo adotados em diferentes sistemas.

No capítulo 3, é feito um levantamento das principais questões relacionadas à TV interativa. Inicialmente, são descritas as possibilidades de interação na TV. Em seguida, são descritos ambientes alternativos para a implementação da TV interativa, com arquiteturas e padrões, dando destaque para a TV digital interativa.

No capítulo 4, o modelo original de *storytelling* interativo do Logtell é apresentado, de modo a dar a base necessária para a compreensão do novo modelo proposto no capítulo seguinte.

No capítulo 5, o novo modelo de *storytelling* interativo para atender aos requisitos de ambientes de TV interativa é descrito.

No Capítulo 6, explicam-se os principais pontos da implementação do protótipo que foi feita para validar as principais idéias do modelo proposto no capítulo 5.

No capítulo 7, é feita uma análise dos testes realizados no protótipo no que tange ao atendimento aos requisitos do modelo, procurando identificar os pontos implementados e os pontos ainda não resolvidos.

O capítulo 8 contém as conclusões da pesquisa com a descrição de suas principais contribuições e dos trabalhos futuros potenciais, os quais foram identificados ao longo da elaboração do modelo e dos testes realizados com o protótipo.

## 2. *Storytelling* Interativo

Com o advento das novas tecnologias digitais e também de suas respectivas inovações a respeito de novas interfaces com o usuário, cada vez mais expandem-se as possibilidades de interação. Sistemas interativos são de grande apelo para muitos: grande prova disso são os jogos eletrônicos, cada vez mais presentes na sociedade, e em diversos meios e formatos, que vão desde o telefone celular e o computador até a televisão digital.

Em relação ao que se pode se chamar de uma das mais antigas artes, a de contar histórias, existe a recente área de pesquisa do “*Interactive Storytelling*”. Trabalhos fundamentais como (GLASSNER, 2004) discutem as possibilidades e dificuldades do casamento entre interação e *storytelling*, histórias não-lineares, e ficção interativa.

Histórias são representações de seqüências de eventos que formam padrões reconhecíveis à mente humana, que evocam nossos arquétipos culturais intrínsecos e nos ensinam lições na medida em que são contadas. Tais efeitos são pesquisados principalmente na área da psicanálise e afins, como em (FRANZ, 1980), livro da discípula de Carl Jung, este por sua vez, ensinado por Freud, o pai da Psicanálise. No entanto, o próprio advento da interatividade é quase que antitético ao objetivo de contar histórias, já que permitir que um usuário tome decisões que afetem uma história parece ser quase que uma garantia de se poder desestabilizar o difícil equilíbrio exigido pelo autor da narrativa.

Chris Crawford em (CRAWFORD, 2005) define interatividade como “*um processo cíclico entre dois ou mais agentes ativos no qual cada agente alternadamente escuta, pensa e age*”. Partindo desta definição, deve-se esperar de um sistema de *storytelling* interativo que este “escute” os desejos do usuário, e, na medida do possível, tente “pensar” sobre eles, e que então “aja” para contar sua história de maneira compatível com os anseios do usuário. Nessa visão, tanto o usuário como o sistema que conta a história são agentes que se relacionam. O usuário também terá, do seu lado, a oportunidade de “escutar” a história narrada, “pensar” sobre ela e seus significados, e então “agir” de forma a repetir o ciclo de interação enquanto for desejado (ou possível).

Um ambiente onde se encontra algo análogo ao que se deseja para o *storytelling* interativo digital, é o dos Role Playing Games, ou RPGs. Nos RPGs, os jogadores participam de uma história que é feita em parte pelo Mestre do Jogo, que geralmente já tem um certo “roteiro” escrito de coisas que devem ou podem acontecer, em linhas

gerais; e parte pelas vontades e ações dos jogadores, estas sempre submetidas à regras do Jogo e probabilidades, através de dados, por exemplo. Os RPGs, porém, quando na forma eletrônica, comportam-se, em sua maioria, como os jogos eletrônicos comuns. Têm uma série de eventos roteirizados ou então alguns “ramos” de séries de eventos, alternados por momentos de controle dos protagonistas pelo usuário, com um certo grau de influência sobre a história, mas nunca com o potencial fornecido pela sua versão não digital, que tem um “autor” pronto para adaptar a história que se desenrola.

Em *storytelling* interativo, o motor gerador da história deve ser capaz de gerar histórias *on-the-fly*, ou seja, em tempo real, baseado no *feedback* do usuário, sem utilizar apenas seqüências de ações previamente escritas ou séries de eventos amarradas, como vemos em geral nos demais produtos multimídia existentes, tais como jogos eletrônicos, filmes, etc. Para conseguir isso, é necessária uma profunda abstração do processo de contar histórias. A melhor maneira de alcançar isto é pensando não em um roteiro, mas sim em um *storyworld* (CRAWFORD, 2005).

Em um *storyworld*, um usuário deve ser capaz de fazer escolhas que determinem os eventos futuros deste mundo. As escolhas devem ser interessantes dramaticamente, variadas e balanceadas – ou seja, a história deve ser capaz de ir em várias direções de acordo com as decisões do usuário.

Este conceito de pensar nas histórias como um mundo com suas próprias regras, em vez de apenas em uma seqüência predeterminada, não é inconsistente com o processo de autoria de uma história normal. Muitos autores de literatura e dramaturgia, dependendo do seu método criativo, já fazem isso internamente ao escrever uma história, como por exemplo sugere Robert McKee, roteirista, em (MCKEE, 2006), quando diz "*Uma estória deve obedecer a suas próprias leis internas de probabilidade. A escolha de eventos do escritor, portanto, é limitada às probabilidades e possibilidades contidas no mundo que criou*". Ou seja, toda história deve obedecer a suas próprias regras e lógica interna, se bem escrita. Caso contrário, mesmo um espectador comum, leigo na escrita literária e nas artes dramáticas, ao ver um filme com “furos”, ou inconsistências no roteiro, fica com uma estranha sensação de que algo está errado (“Isso não faz sentido, Fulano nunca agiria assim!”), mesmo quando não consegue explicitar exatamente o que há de inconsistente na história que lhe foi narrada.

Existem três focos principais nas pesquisas de *storytelling* interativo: geração, visualização e interação. De certo modo, pode-se fazer uma analogia ao cinema: a geração faz o “roteiro”; a visualização produz as imagens dos atores; e a interação

corresponde a um diretor que, atento aos anseios do espectador, coordena a realização do processo como um todo. Em maior detalhe:

- *Geração* – Lida com a questão dos rumos para a onde a história converge, seus personagens, ações e relacionamentos.
- *Visualização* – Lida com a representação gráfica dos elementos da história, de forma coerente com seu “roteiro” subjacente.
- *Interação/Direção* – Lidam com a dificuldade de se conciliar a interação entre os agentes da história, levando-se em conta seu nível de autonomia. A interação entre o usuário e a história bem como entre os personagens estão intimamente ligados à manutenção da coerência da história, demandando a compatibilização com o processo de geração da história.

Para que seja possível o uso de sistemas de *storytelling* interativo em mídias novas como a Televisão Digital, além da dificuldade de se equilibrar coerência e interatividade, deve-se considerar que se precisa encontrar uma forma híbrida de apresentação de conteúdo multimídia, misturando características de jogos e televisão normal. É necessário manter o apelo para o espectador comum, e, ao mesmo tempo, permitir que o usuário seja capaz de interagir de diferentes maneiras com o conteúdo que lhe é apresentado.

Um sistema de *storytelling* interativo, quando aplicado a este caso, deve inclusive considerar o fato de que o usuário pode nem mesmo querer interagir de modo algum. O sistema precisa ter tanta responsividade quanto possível e a história deve ser fluente, sempre mantendo a simplicidade e a sensação de conforto na interação, especialmente quando se considera que as histórias em geral serão assistidas por espectadores que podem não ser ávidos usuários de jogos eletrônicos.

Em jogos, a coerência tem sua importância, mas de forma limitada, pois, usualmente, o prazer de jogar está mais ligado a vencer desafios, mesmo que entremeado com pedaços previamente escritos de histórias. Em *storytelling* interativo o prazer está em assistir a uma história com a possibilidade de intervir no processo, o que demanda um nível de coerência comparável ao de textos literários ou filmes.

## **2.1 Modelos de Geração de Histórias**

Os Sistemas de *storytelling* interativo têm sido pesquisados para diferentes aplicações e implementados de diferentes modos. Podem ser aplicados a jogos

(YOUNG, 2001), auxílio à autoria de histórias, entretenimento em geral e até a aplicações comerciais (SPIERLING *et al.*, 2002). Dentre as coisas que distinguem um sistema de *storytelling* interativo do outro, destacam-se a perspectiva pela qual a história é contada e o foco da geração dos enredos, que pode ser nos personagens ou na trama.

Descrevemos a seguir as principais abordagens que vêm sendo propostas e implementadas em sistemas de *storytelling* interativo. Uma das dificuldades mais comuns ao se criar um sistema de *storytelling* interativo é fornecer um bom nível de interatividade e cativar o usuário, enquanto se mantém a coerência da história que é gerada. Cada uma das abordagens tende a favorecer aspectos diferentes, ora privilegiando a interação, ora privilegiando a coerência, ora privilegiando o poder dramático. A abordagem particular que adotamos para um ambiente de TV interativa é baseada em uma modificação do sistema Logtell (CIARLINI *et al.* 2005) que é descrito em maiores detalhes no capítulo 4.

### **2.1.1 Abordagens Centradas em Personagens (*Character Based*)**

Alguns sistemas de *storytelling* se focam mais em personagens (*character-based*), como em (CAVAZZA *et al.*, 2002; CHARLES *et al.*, 2001; YOUNG, 2001) onde a história emerge da interação em tempo real de agentes autônomos, cada qual com seus objetivos e seu comportamento. Isto facilita a intervenção do usuário, pois geralmente as ações de qualquer personagem da história podem ser mudadas. Esta abordagem tende, no entanto, a tornar mais difícil garantir que as situações emergentes permaneçam coerentes com o gênero e sejam suficientemente interessantes.

No trabalho do grupo de Cavazza, os usuários são espectadores que podem interagir com objetos da história e também dar sugestões a personagens, de modo a afetar suas decisões e histórias resultantes. Desta maneira, pode-se reduzir o impacto do dilema interação versus coerência, pois limita-se o envolvimento do usuário na história, mas permite-se que ocorram interações a todo momento. A maior questão contra o uso de abordagens puramente focadas em personagens é até que ponto podem emergir situações dramáticas e cativantes. Não à toa, este tipo de abordagem foi justamente utilizado por Cavazza em um tipo de história *sitcom*, ou seja, comédia de situações, onde o clímax de uma história não é tão facilmente distinguível.



Figura 2.1: Dramatização de algumas histórias de (CAVAZZA *et al.*, 2002)

### 2.1.2 Abordagens Centradas na Trama (*Plot Based*)

Outros trabalhos se focam mais na trama (*plot-based*), como em (GRASBON e BRAUN, 2001; PAIVA, 2001), que se baseiam em regras estruturais mais rígidas e pré-estabelecidas na trama para guiar a história. Isso normalmente facilita a coerência e a autoria de histórias com boa carga dramática, mas acaba restringindo o potencial de intervenção do usuário. Geralmente, nesta abordagem a trama possui diversos pontos de ramificação predefinidos, um início e um fim, e é exigido um grande esforço do construtor da história para permitir um número mínimo de variações, pois elas precisam ser consideradas previamente. Uma das principais fontes de inspiração para este modelo é o trabalho literário de Vladimir Propp (PROPP, 1968) no começo do século XX. Propp analisou um conjunto grande de contos de fadas Russos e mostrou que todos podiam ser descritos por especializações de 31 funções típicas, como vilania, partida do herói, recompensa, etc. Nas abordagens puramente *plot-based*, a intervenção do usuário é mais limitada porém garante-se com mais facilidade a coerência.

Neste tipo de abordagem, ao usuário geralmente só é permitido fazer sutis interferências, de modo que a história não se afaste da estrutura predefinida. Nesta forma de abordagem, a autonomia dos agentes é, ao contrário da abordagem *character-based*, um risco, pois seu comportamento reativo e as ações locais podem entrar em conflito com aspectos fundamentais da trama da história (MATEAS e STERN, 2000).



Figura 2.2: Teatrix em ação – Criando uma história. Na parte central da figura, vê-se o personagem controlado e também os outros personagens no local.

Na pesquisa de (PAIVA, 2001; PRADA, 2000), foi criado o ambiente Teatrix, que usa as funções de Propp para modelar personagens artificiais, ou NPCs (*non player characters*) que interagem com outros personagens, criados por crianças, num mundo virtual. Cada criança controla um personagem e os NPCs são autônomos. Cada personagem tem um papel na história, especificando as funções de que fazem parte. Os NPCs têm objetivos que se adaptam às situações, planejando e tentando executar ações/funções de acordo com seus papéis, que são baseados no trabalho de Propp, dentro de um conjunto que inclui um vilão, uma princesa, um ajudante, etc. É uma abordagem interessante principalmente para educação, mas o controle da consistência das ações e objetivos e a geração das situações dramáticas não são garantidas. Para melhorar as chances de coerência, há um agente “Diretor”, que tem privilégios adicionais, podendo controlar outros agentes em prol da coerência, e é quem determina o fim da história. Este agente diretor também pode ser controlado pelos usuários (as crianças).

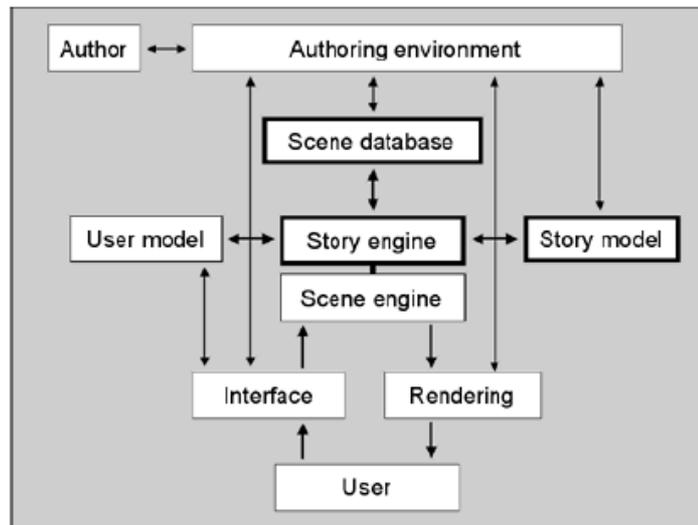


Figura 2.3: Arquitetura de (GRASBON e BRAUN, 2001)

Na pesquisa de (GRASBON e BRAUN, 2001), também baseada em Propp, tem como fundamento que o autor defina cenas e as associe a funções narrativas. Sua pesquisa questiona a forma de geração da configuração inicial das histórias, visto que os autores não acreditam na geração automática de todos os detalhes das histórias. Como apresentado na figura, a plataforma, proposta nesse trabalho, baseia-se fortemente no processo autoral, ao definir um modelo global para a história, e gerenciar a trama em alto nível - no nível das funções morfológicas definidas por Propp. Nessa plataforma, o usuário não é autor, ele apenas interage sobre os dados definidos pelo autor. O protótipo funciona com uma interface em modo texto, mas prevendo-se a sua utilização em um sistema de realidade aumentada.

Em trabalho posterior do mesmo grupo (SPIERLING *et al.*, 2002), tenta-se explorar *storytelling* interativo em diferentes níveis em um “palco experimental”, com o intuito de equilibrar flexibilidade e pré-determinação. Seu trabalho sugere a criação de camadas que permitam aos autores trabalharem com a direção em níveis de detalhe que vão desde as situações dramáticas em alto nível até a direção de atores animados. O trabalho argumenta que somente a forma de narrar a história deve ser interativa, e seus módulos operam sobre histórias pré-definidas, com tramas definindo os eventos e fatos, deixando apenas ao autor o poder modificar aspectos mais gerais da história. Nesse esquema, é obviamente necessário um grande esforço autoral para construir os modelos das histórias. Na Figura 2.4, mostra-se a arquitetura do sistema. Cada camada usa os dados da camada acima de si. O *Story Engine* controla o fluxo da narrativa, ou seja, a ordem dos eventos, baseados nas funções proppianas, em um nível abstrato. As funções

são específicas para seus gêneros. O *Scene Action* é responsável pela escolha das cenas, que incluem controles de câmera e ações dos personagens, descritos em *scripts*. No nível seguinte, tratam-se as relações, diálogos e interações entre os agentes. No nível inferior, *Presentation*, estão recursos adicionais para sons e animações. Em cada estágio, o autor pode definir o nível de autonomia que cada motor pode exercer, podendo ser totalmente autônomo, manual, ou uma combinação.

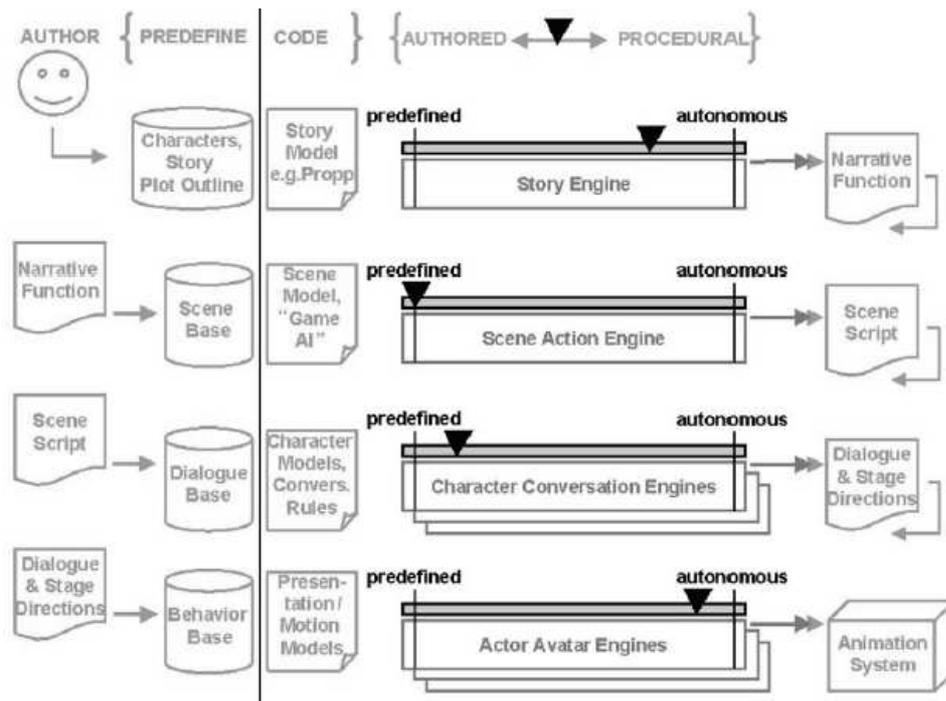


Figura 2.4: Arquitetura de (SPIERLING *et al.*, 2002).

### 2.1.3 Abordagens Híbridas e Outras Abordagens

No sistema DEFACTO (SGOUROS, 1999) é utilizada uma abordagem onde são usadas sucessivas avaliações de regras para controlar a geração de uma história interativa onde o usuário é o protagonista. A interação entre os objetivos dos personagens é representada explicitamente e uma concepção Aristotélica de trama é usada para levar a história a um clímax e resolvê-lo. O encadeamento de eventos, contudo, não é explicado por pré- e pós-condições, tornando o controle do que pode ou não acontecer mais complexo. Além disso, não se permite o uso de algoritmos de planejamento para explorar seqüências de eventos, ao invés de eventos únicos, para alcançar objetivos. A necessidade de intervenção do usuário parece ser alta, para a

criação de uma história completa. Objetivos são inferidos através de regras que analisam a situação atual, mas as escolhas de ações para alcançar objetivos parecem ser mais reativas do que deliberativas.

Existem também alternativas que integram características tanto *plot-based* quanto *character-based*. O sistema interativo Façade (MATEAS e STERN, 2000), por exemplo, tem um gerenciador de dramas que permite que os personagens sejam autônomos a maior parte do tempo, mas que muda seus comportamentos para adiantar a trama, conciliando objetivos de nível maior, que são essenciais à história, com objetivos menores, específicos dos personagens autônomos. Já no Erasmatron (CRAWFORD, 1999), usa-se a noção de verbos e sentenças. Ações são representadas por verbos com papéis atribuídos aos personagens de modo a formar sentenças.

A implementação do Erasmatron (CRAWFORD, 2005) trabalha em cima do conceito de verbos e frases, tentando balancear os aspectos da trama e de personagens da narrativa. As ações são representadas por verbos, com listas de papéis que podem ser atribuídos aos personagens para formar frases. Funções são implementadas como operadores lógicos, com parâmetros e pré- e pós-condições.

No Façade, lançado em 2005 (MATEAS e STERN, 2005) o usuário é um convidado para tomar *drinks* na casa de um casal de personagens (*Trip* e *Grace*). Através de interações num ambiente 3D, que permitem manipular alguns objetos e conversar com os personagens através de linguagem natural, o usuário pode influenciar o rumo da trama: o casal pode vir a brigar, pode vir a se entender, ou pode acontecer ainda de o usuário ser simplesmente expulso da casa deles. O sistema faz o uso de um gerenciador de drama (*drama manager*) para manter o estado da história, conciliando aspectos *character-based* e *plot-based*. Os personagens são autônomos a maior parte do tempo, mas seus objetivos e comportamento podem ser modificados pelo *drama manager*, de modo a mover a trama adiante. O usuário é o protagonista da história, e o *drama manager* seleciona as cenas a serem apresentadas. As cenas são compostas de *beats*, unidade usualmente utilizada no cinema (porém nesse caso os *beats* duram em torno de um minuto, um pouco menos que em filmes). Os *beats* definem a granularidade da interação entre os personagens e a trama. O usuário pode interferir na execução de um *beat*, determinando como o resto da cena será interpretada pelos dois atores, *Trip* e *Grace*. Esta abordagem separa objetivos de alto nível, importantes para o enredo da história, de objetivos menos importantes, mais específicos dos comportamentos dos personagens. Para modelar esse comportamento dos personagens, foi criada uma

linguagem própria, ABL – *A Behaviour Language* (MATEAS e STERN, 2004), capaz de definir os objetivos compartilhados e seqüências de ações para alcançar os objetivos. Note-se que o Façade adota a estratégia de trabalhar com uma situação dramática contida e bem delineada, que leva algo em torno de 20 minutos para ser resolvida, dependendo do usuário. Mesmo assim, a sua implementação exigiu um grande esforço autoral, de mais de 4 anos e mais de 100 mil linhas de código escritas. Apesar disso, o Façade é um dos trabalhos mais conhecidos no ramo do *storytelling* interativo, estando disponível na Internet e de fácil instalação e uso.



Figura 2.5: Façade e seus dois atores na visão 3D do usuário.

No IDTension de Nicholas Szilas (SZILAS, 2003; 2008), uma arquitetura é proposta, a qual se concentra não apenas na geração de seqüências de eventos ou ações, mas também na simulação de “leis da narrativa” que guiam o processo de *storytelling*.

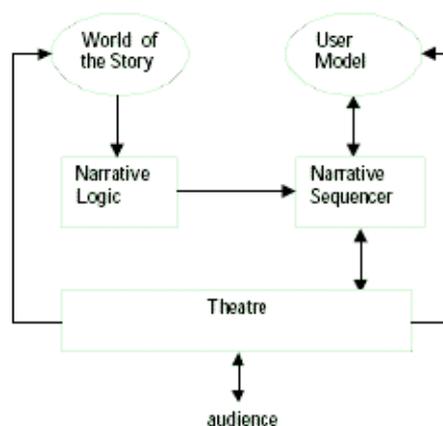


Figura 2.6: Arquitetura proposta por (SZILAS, 2003)

Na arquitetura de Szilas, existem vários módulos. O módulo *World of the Story*

contém entidades básicas (personagens, objetivos, tarefas, sub-tarefas, segmentos e obstáculos), estados dos personagens (definidos por predicados) e fatos sobre a situação material do mundo da história. O módulo *lógica narrativa* (*Narrative Logic*) determina, a partir dos dados no *mundo da história*, o conjunto das ações possíveis para cada personagem. O módulo *Narrative Sequencer* decide que ações executar com base em quesitos de consistência, conflito, surpresa, etc., buscando criar tensões dramáticas. Para a criação dos conflitos, é usado um modelo de valores morais e então promovem-se situações para obrigar personagens a realizarem ações que vão contra seus valores para alcançar os objetivos. O módulo *User Model* contém o estado de um usuário em determinado momento na narrativa e identifica seus valores mais importantes. Provê, dessa forma, ao *Narrative Sequence*, uma estimativa do impacto sobre o usuário de cada possível ação. O módulo *Theatre* cuida da exibição das ações e gerencia a interação com o usuário.

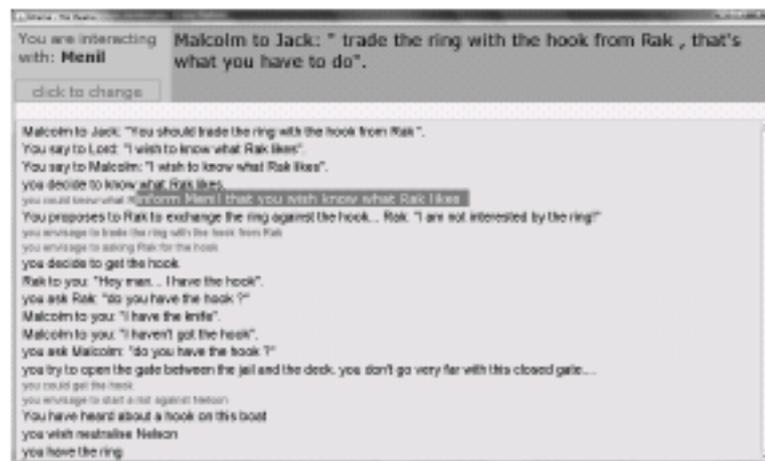


Figura 2.7: Protótipo de (SZILAS, 2008), modo texto: um drama interativo aonde o jogador deve fazer um motim num navio.

Em SI *et al.* (2008) é descrita uma abordagem que tem como foco a autoria e controle de personagens virtuais em dramas interativos. Processos centrados na trama são integrados com processos centrados nos personagens através da criação de um *framework* para a autoria/criação de dramas interativos. Esse *framework* integra um planejador de ordem parcial (POP), normalmente usado em abordagens centradas na trama, com o sistema Thespian (SI *et al.*, 2005) que usa principalmente a abordagem centrada em personagens. O Thespian garante a consistência das motivações dos personagens durante a interação e usa agentes autônomos orientados a objetivos para

controlar personagens virtuais. As motivações dos personagens são codificadas como objetivos dos agentes. O Thespian fornece um procedimento automatizado para regular as motivações dos personagens virtuais com um conjunto de caminhos da história (seqüências de ações dos personagens). O ponto inicial para o processo da autoria da história é um conjunto de *scripts* padrão. Um algoritmo então facilita o processo da autoria ajustando automaticamente os objetivos dos agentes de forma que os agentes executem seus papéis de acordo com os *scripts*. Isto também garante que os agentes vão agir de forma compatível com suas motivações quando o drama interativo foge dos *scripts*.

## 2.2 Interactive *Storytelling* e Planejamento Automatizado

O uso de técnicas de planejamento automatizado é parte importante de alguns sistemas de *storytelling* interativo, pois são algoritmos feitos justamente para criar seqüências lógicas de eventos.

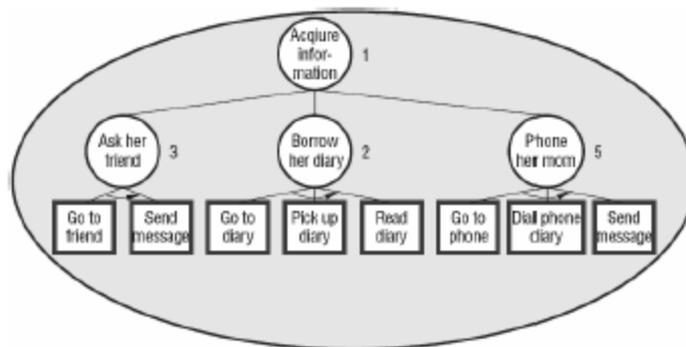


Figura 2.8: Uma HTN do protagonista. Reproduzida de CAVAZZA, 2002 – pg 19.

Em (CAVAZZA *et al.*, 2002), planejamento com redes hierárquicas de tarefas (HTN) é usado para controlar a forma como personagens alcançam seus objetivos em concordância com as intervenções do usuário. Planejamento com HTN costuma ser eficiente, mas menos geral, pois é fortemente preso ao conhecimento do domínio, exigindo um esforço alto na construção de métodos para realizar cada tarefa na história.

Algoritmos de planejamento mais flexíveis foram adotados no desenvolvimento de outros sistemas. Tais algoritmos não se limitam a soluções previamente pensadas. Em vez disso, combinam eventos conciliando objetivos diferentes com pré-condições de cada evento. Isso aumenta ainda mais o esforço computacional dada à natureza complexa de planejamento automatizado, mas torna o sistema mais flexível e, portanto, capaz de gerar maior variedade de histórias. Na maioria das abordagens que fazem uso

desta técnica de planejamento, são usados algoritmos de planejamento de ordem parcial, onde só é estabelecido que um evento deve ocorrer antes de outro se isso for estritamente necessário para o alcance dos objetivos. Em ambientes de *storytelling* interativo, o uso de ordem parcial é uma alternativa interessante porque dá mais flexibilidade para interagir com o usuário.

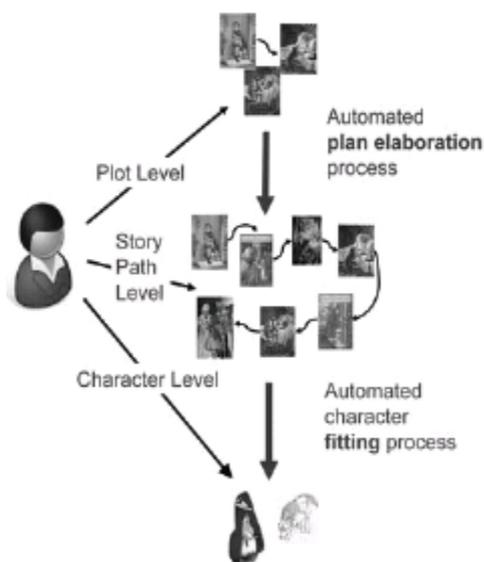


Figura 2.9: Framework de SI *et al.* (2008) que usa planejamento.

No trabalho de SI *et al.* (2008), o *framework* proposto usa planejadores parcialmente ordenados para configurar os agentes baseados em objetivos do Thespian. Esse *framework* permite o uso de um planejador que modela a história em um nível maior de abstração do que o Thespian, que controla os agentes. Esta abordagem poupa o autor de ter que construir um modelo completo no nível de detalhe demandado pelo Thespian, o que permite que o autor rapidamente esboce a experiência interativa.

Em (RIEDL e YOUNG, 2004) é descrito o uso de um planejador parcialmente ordenado para *storytelling* interativo, que incorpora informações correspondentes às intenções dos personagens, sob a forma de relações causais entre os efeitos de um evento e as pré-condições de outro. Com base na análise de relações causais em tempo de execução, é possível verificar quando uma ação de um usuário pode afetar a história que está sendo contada. Dessa forma, mecanismos de mediação, como os do Mimesis (YOUNG, 2001) (YOUNG *et al.*, 2004) podem ser adotados. O Mimesis é um sistema para controlar ambientes virtuais em jogos que usa *storytelling* com planejamento. Nele, as narrativas são geradas por um algoritmo de planejamento, a princípio sem considerar

os objetivos dos personagens. É, então, feito o uso de mecanismos para a detecção de exceções e mediação. Exceções são as ações que podem comprometer as seqüências causais no plano da história; e mediação é a forma usada para tratar estas exceções. Para a mediação há duas alternativas básicas. A primeira é a acomodação, onde o plano é modificado de modo não atrapalhar a história previamente planejada. A segunda é a intervenção, onde o personagem controlado pelo usuário é impedido de executar uma ação que comprometeria demais a história.

Em (RIEDL e YOUNG, 2006) é descrito mecanismo onde, para cada ponto onde pode ocorrer uma exceção, usa-se o planejador para obter uma história alternativa começando naquele ponto. Esse mecanismo de antecipação cria uma árvore onde os caminhos correspondem às narrativas que podem ocorrer. Para a árvore não crescer demais, utiliza-se intervenção em certos pontos. Com a geração prévia da árvore de narrativas, evita-se a necessidade de planejamento em tempo real. No entanto, o direcionamento da narrativa tende a ser mais rígido do que quando se opta pelo planejamento em tempo real. Em (RIEDL *et al.*, 2008), é apresentado um *framework* para criar narrativas em um mundo virtual para entretenimento, educação e treinamento. O *framework* usa essas técnicas de antecipação de exceções para controlar a experiência do usuário de modo a que os objetivos da aplicação sejam atingidos.

### **2.3 Conclusões**

Neste capítulo, procurou-se dar uma visão geral da pesquisa em *storytelling* interativo descrevendo-se as principais questões envolvidas e os enfoques que vêm sendo adotados em diferentes sistemas.

Há um consenso dentre os pesquisadores de que não há condições de existir um processo completamente automático: sempre é necessário usar estratégias que combinem um esforço autoral com a criação de *frameworks* que formalizem de algum modo as regras necessárias para a criação de narrativas minimamente interessantes para seres humanos. Grande prova disso é o próprio Façade, que é tido como a mais bem sucedida experiência de *Interactive Storytelling* até então, mas que precisou, como mencionado anteriormente, de mais de 4 anos de desenvolvimento e um grande esforço autoral. Soma-se a isso o fato de que, no Façade, todo o esforço autoral foi feito pelos próprios pesquisadores que fizeram os algoritmos que gerenciam a narrativa e a interação com o usuário. Isto ilustra o fato que a autoria das histórias interativas ainda envolve grande conhecimento das tecnologia criadas. Para a criação de histórias

interessantes, a experiência de autores mais focados nas histórias do que nos aspectos da implementação é fundamental de modo a se obter uma produção em boa quantidade de novas obras de *storytelling* interativo.

Também é importante demonstrar as diferenças que há nas abordagens quanto à geração das histórias e à forma como elas lidam com as interferências do usuário. Parece haver um consenso de que é preciso tentar balancear os objetivos dos personagens com os da trama em geral, e diferentes abordagens lidam com essa dificuldade de maneiras diferentes.

Há uma tendência em sistemas de *storytelling* de se usar algoritmos de planejamento automatizado, em especial algoritmos de planejamento de ordem parcial ou com redes hierárquicas de tarefas. Algoritmos de ordem parcial parecem ser uma escolha lógica, dada a sua natureza, pois são capazes de retornar ordens parciais de eventos que devem ocorrer para alcançar objetivos, de forma bem semelhante ao que ocorre na própria estrutura e lógica interna das narrativas convencionais. Redes hierárquicas de tarefas também mostram-se compatíveis, pois são capazes de criar seqüências de eventos mais abstratos que se decompõem em outros mais específicos, o que se mostra compatível com a lógica das narrativas, ao se pensar que uma história pode ser decomposta em conceitos abstratos que podem ser realizados por diferentes métodos.

Existe uma gama de diferentes opções de implementação no que concerne à maneira como as narrativas são apresentadas, e também à maneira como o usuário pode interagir sobre elas. Em algumas pesquisas, como no Façade e no IDTension, a perspectiva da narrativa é em primeira pessoa; já no sistema de Cavazza e no Teatrix, é na terceira pessoa. Além disso, varia-se muito a forma de apresentação. Boa parte dos sistemas descritos apresenta narrativas através de animações 3D, mas alguns estão orientados à geração de texto, como no IDTension. Esses aspectos também estão ligados à forma de interação que pode ser direta, através de diálogos ou da manipulação direta do cenário e de avatares, ou indireta, através de diferentes métodos. Em enfoques *character-based*, sente-se a inspiração direta de jogos, o que leva a narrativas com intervenções diretas que podem acontecer a qualquer momento. Em enfoques *plot-based*, prevalece a inspiração de textos literários e do cinema, havendo predominância de narrativas em terceira pessoa com intervenções indiretas.

O sistema usado nessa pesquisa foi o Logtell. Este sistema de *storytelling* tem semelhanças com as pesquisas de Grasbon e Spierling (SPIERLING *et al.*, 2002;

GRASBON e BRAUN, 2001), por ser primeiramente *plot-based* e inspirado inicialmente na teoria de Propp. Por outro lado, é fortemente pautado na formalização lógica do contexto, de modo a utilizar planejamento automatizado para explorar virtualmente todas as possibilidades logicamente coerentes para esse contexto. O uso de planejamento para garantir a geração de seqüências coerentes de eventos se assemelha ao adotado em (YOUNG *et al*, 2004), mas se baseia também na formalização, em lógica temporal, das situações que levam ao surgimento de objetivos. O processo de geração é feito através de múltiplos estágios que incluem fases de inferência de objetivos, planejamento e interação com o usuário. A formalização dos objetivos de personagens e da história aproxima o Logtell de sistemas que tentam conciliar aspectos *character-base* e *plot-based*, como no Façade. No aspecto da visualização, o Logtell se assemelha ao sistema de Cavazza, usando visualização 3D e uma perspectiva em terceira pessoa.

Para a utilização de *storytelling* interativo em um ambiente de TV interativa, a diversidade e coerência dos enredos, bem como a possibilidade de interações com o usuário em variados níveis de intensidade é importante. No Logtell, é possível gerar uma boa diversidade de histórias coerentes, desde que haja um esforço autoral para definir as regras do contexto, dificuldade compartilhada por todas as pesquisas em *storytelling* interativo. Interações com o usuário podem ocorrer em níveis diferentes, mantendo a consistência desejada. Dessa forma, a arquitetura do Logtell parece fornecer uma boa base para os propósitos de utilização de *storytelling* interativo na TV. O capítulo 4 fornece maiores detalhes sobre a versão inicial do Logtell, explicando suas potencialidades e as limitações para a sua aplicação na TV interativa, as quais motivaram o desenvolvimento desta pesquisa para a sua extensão.

O capítulo seguinte expõe uma visão geral sobre as tecnologias utilizadas na televisão interativa e suas perspectivas, de forma a demonstrar sua compatibilidade com o trabalho proposto.

### 3. Televisão Interativa

Nos últimos anos, a TV analógica tradicional vem caminhando para ser substituída por uma TV digital aberta com qualidade de som e imagem bastante superior e possibilidades de interação. Além disso, temos visto uma ampla disseminação do uso de novos meios de comunicação, tais como a internet banda larga e celulares 3G, que servem também como alternativas interessantes para a veiculação de programas em princípio “televisivos”. Esses fenômenos resultam de grandes avanços nas tecnologias digitais e da convergência digital, que permite juntar diversas melhorias e inovações em diferentes frentes, tais como na infra-estrutura das redes de comunicação, no software e no hardware para compressão e transmissão de dados e nos serviços de *broadcasting* (FURHT, 1996). Como resultado, abre-se, um leque de novas tecnologias prontas a revolucionar a forma como as pessoas vêem e interagem com os aparelhos de TV. As mudanças dizem respeito não apenas à qualidade de imagem e som (DRISCOLL, 2000), mas também ao conteúdo disponível e à relação com o espectador.

Dentro desse cenário, uma questão central é a capacidade de interatividade, o que levou ao surgimento da TV interativa ("*interactive TV*", "*enhanced TV*" ou simplesmente *iTV*) (SWEDLOW, 2000) como uma área de interesse importante tanto para a pesquisa quanto para a indústria.

A própria televisão analógica convencional apresenta cada vez mais programas em que os espectadores são convidados a participar, *reality* shows, etc., o que demonstra a existência de um potencial a ser explorado no que tange às possibilidades de interação. A iTV não seria nada mais do que a evolução natural do meio analógico, agregando interatividade ao próprio vídeo e trazendo a oportunidade de um *feedback* imediato do espectador às distribuidoras de conteúdo que, muitas vezes, costumam ser as próprias produtoras. Possibilidades de interação incluem:

- uma interatividade mais fraca correspondente à simples possibilidade de se rever programas na hora desejada, pular comerciais, executar comandos de VCR, etc.; a obtenção de mais informações sobre o que é apresentado, sejam filmes, notícias, esportes, etc.;
- a veiculação de propaganda direcionada e individualizada (em conjunto com mecanismos de vendas);
- a interferência efetiva no conteúdo apresentado, mudando por exemplo o final de uma história que se está assistindo;

- a interação conjunta de um grupo de usuários com um conteúdo compartilhado.

Esses dois últimos itens têm forte apelo para o desenvolvimento de aplicações voltadas tanto ao entretenimento quanto à educação e são de particular interesse neste trabalho, onde se estuda o uso de *storytelling* interativo dentro de uma perspectiva de TV interativa.

Em se tratando de aplicações para TV, é necessário estudar modelos que permitam garantir a responsividade dos sistemas, ou seja, o atendimento aos anseios dos usuários de forma rápida e sem alterações no nível de qualidade. O usuário de TV está habituado a assistir um fluxo contínuo de apresentação dos programas sem quebra na qualidade de som, do vídeo e da diversidade e coerência do conteúdo. Por outro lado, a interação com a TV tende a ser diferente da interação com computadores pessoais, pois, usualmente, não é tolerável que se demande do usuário um grande esforço de atenção.

A TV digital aberta é naturalmente um dos principais meios de comunicação onde a pesquisa em TV interativa deverá ser aplicada. No contexto brasileiro, em particular, a ampla penetração da TV em todas as camadas sociais, se comparada por exemplo à penetração da internet, faz com que seja de interesse estratégico, tanto para o governo quanto para a indústria, a pesquisa e o desenvolvimento nessa área (CPqD, 2005). É preciso explorar potencialidades e oportunidade que se abrem (TOME et al., 2001), tanto para fins de entretenimento quanto de educação (SANTOS *et al.*, 2005) (SANCRINI, 2005). Deve-se notar, no entanto que a convergência digital faz com que aplicações desenvolvidas com foco na TV digital possam ser adaptadas para outros meios, tais como TV via Internet e TV via celular, e vice-versa.

Este capítulo aborda diversos aspectos relacionados à TV interativa. Inicialmente, são tratadas as possibilidades de interação e de desenvolvimento de aplicações e serviços, levando em conta os requisitos de facilidade de interação e responsividade. Em seguida, são investigados ambientes e abordagens para a TV Interativa, dando destaque ao ambiente de TV digital aberta. Por fim, são apresentadas as conclusões no que diz respeito à adequação de modelos de *storytelling* a ambientes de TV interativa.

### 3.1 Interatividade e Televisão

Uma das principais mudanças que surgem com a iTV é justamente a forma como é utilizada a televisão. Agora, o usuário vai além de um espectador no sentido mais puro da palavra, surgindo a possibilidade de se interagir com o que se assiste.

É importante ressaltar que a interatividade com programas de televisão já existe há um certo tempo, embora de forma mais improvisada e bem menos dinâmica. Um primeiro exemplo corresponde a programas de televisão onde o espectador pode ligar para tomar decisões, como em *reality shows*. *Game shows*, onde o telespectador responde a perguntas e pode ganhar prêmios são exemplos ainda mais antigos. Telejogos como "Hugo" que chegou a ser exibido no Brasil por um tempo, onde o espectador participavam via telefone dando comandos através do tom das teclas, são exemplos também interessantes. Por fim, outro tipo de programa que serve de analogia para o produto desta pesquisa foi o "Você Decide", onde o espectador podia, através de votação por telefone, escolher uma alternativa de decisão a ser tomada pelo protagonista de uma história, assistindo a seguir às conseqüências da alternativa mais votada. A característica interessante nesse caso era que não havia pausa: o espectador via a história e isto o ajudava a tomar a decisão.

De todo modo, a demanda por essa "interatividade improvisada" serve pra mostrar que há um potencial de atratividade em programas onde os telespectadores podem interagir, em vez de permanecerem apenas como espectadores passivos. Com o advento de tecnologias cada vez mais potentes e ao alcance de cada vez mais pessoas, novas funções interativas são disponibilizadas nas diversas mídias e os usuários tornam-se mais habituados com a interação através de *gadgets*, internet, videogames, telefones celulares com múltiplas funções, etc. Cria-se dessa forma um cenário onde a interatividade na TV surge como evolução natural.

Não há consenso sobre como se dará a incorporação de mecanismos de interação na TV nem sobre qual será o impacto disso na vida dos usuários. Há enfoques que defendem uma maior interatividade, outros que advogam uma capacidade de interação menor. A capacidade de interação vai depender do poder computacional dos dispositivos que recebem o sinal de TV e o processam (normalmente chamados de *set-top boxes*). Por um lado há a chamada interatividade preguiçosa (*lazy interactivity*), na qual a capacidade de interação seria mais limitada, exigindo dispositivos receptores de sinal menos poderosos e não demandando muita atenção do usuário, o que tenderia a ser mais natural para os telespectadores, mais acostumados, em princípio, a uma postura

passiva. Por outro lado, há outra vertente, que sustenta o uso de um maior poder de processamento nos dispositivos receptores para dar suporte a uma interação mais potente. Em vez de uma interatividade preguiçosa, o foco das aplicações seria em interfaces intuitivas e fáceis, de modo a não atrapalhar a diversão e o entendimento do espectador. A maior capacidade de processamento, ao aproximar os dispositivos receptores e PCs, traz naturalmente vantagens, mas aumentam também as chances de erros de programação e falhas, o que pode comprometer a responsividade. De qualquer modo, é provável que as duas vertentes coexistam, pelo menos por algum tempo. Eventualmente, o mercado poderá fazer com que uma prevaleça sobre a outra ou que, devido à segmentação da economia e dos hábitos da sociedade, continue havendo espaço para ambas.

Independente do tipo de abordagem para TV Interativa, o termo telespectador tende a perder o sentido, pois o usuário deixará de ser apenas um indivíduo que assiste à programação à distancia. O usuário passa, da mesma forma que quando utiliza PCs, a ser um agente ativo no processo, o que torna evidente o anacronismo do termo. A diferença entre as abordagens corresponde ao nível de atividade que pode demandar maior ou menor atenção e pode levar a mudanças mais ou menos significativas no conteúdo que é apresentado.

De acordo com uma evolução natural, o que se espera, é que a interatividade vá crescendo com o tempo, por questões logísticas, que incluem o desenvolvimento de novos modelos de negócio, e de recursos de infra-estrutura. O próprio poder computacional dos *set-top boxes* tende a ser um gargalo para o desenvolvimento de aplicações mais complexas. A rede de comunicação disponível precisa estar adequadamente dimensionada para responder e processar a tempo os desejos e demandas dos usuários, pois não se estará mais lidando com um único conteúdo estático transmitido unidirecionalmente.

### **3.2 Aplicações e Serviços na Televisão Interativa**

Além das questões de infra-estrutura, são necessários esforços para a criação de conteúdo apropriado para o novo meio criado. A geração de conteúdo para esse novo ambiente exige todo um trabalho de projeto e programação, além de envolver a criação de novas áreas de pesquisa e a integração de conhecimentos multidisciplinares, de forma a adaptar os conhecimentos para o ambiente que surge.

Surgem ainda questões a respeito dos fins comerciais que podem ser afetados

pela televisão interativa. A veiculação de propaganda na TV aberta, por exemplo, precisa ser reestudada. Novos modelos de negócio com certeza precisarão ser elaborados e testados.

As aplicações para a TV interativa podem ser dos mais variados tipos. A seguir são apresentados exemplos dos tipos de interação que podem ocorrer nessas aplicações.

### **Interação para Escolha de Conteúdo**

Uma das principais formas de aplicação da iTV é na forma dos guias de programação (BECKER E MONTEZ 2004), que exibem, através de interfaces gráficas, informações variadas sobre a programação e guias de auxílio ao espectador/usuário. Tais informações possibilitam ao usuário escolher programas e horários, comprar conteúdo através de *pay-per-view*, solicitar vídeo sob demanda (Video on Demand, ou *VoD*) e outros conteúdos como horóscopo e programação de rádio de vários estilos. Esses guias também são conhecidos pela sigla *EPG - Eletronic Program Guide* e um exemplo deles é a MSN TV2 (MSN TV2, 2008).

Como as informações estão num meio digital, surge a possibilidade do uso de *personal video recorders*, aproveitando-se que os dados sobre os programas de TV e filmes estarão disponíveis também na forma digital. Esses metadados poderão ser usados como parâmetros para busca, fazendo com que um usuário possa buscar e assistir filmes de algum determinado diretor, ator, tema, etc. O *VoD (video on demand)*, é uma forma de serviço que permite que o usuário possa comprar e/ou assistir a conteúdo, e então assistir quando quiser. É muito usado em novas funcionalidades de *pay-per-view*, com venda de filmes, por exemplo.

Mecanismos mais avançados de interação permitirão o surgimento de uma televisão mais individualizada. Nos noticiários, poder-se-á escolher quais categorias de notícias se quer assistir, documentários interativos poderão funcionar como enciclopédias digitais, e outras formas similares de acesso a conteúdos informativos sobre o que se assiste poderão ser implementadas. Além disso, será possível individualizar a forma como um mesmo conteúdo é apresentado. Usuários poderão assistir a jogos esportivos por ângulos diferentes de câmera e fazer o *replay* de momentos importantes no instante que desejarem.

### **Serviços de Internet e Portais de Televisão Interativa**

Com a presença de um canal de retorno, também surge a possibilidade de se

disponibilizarem outros serviços como internet para navegação e email e também serviços de mensagem instantâneas (*instant messaging*), além de bate-papo e redes sociais.

Como forma de serviços interativos, também surge a opção dos portais, também conhecidos como *walled garden*, que funcionam como os portais de internet, provendo uma série de serviços agregados. Dentro desses portais, o usuário poderá acessar aplicações interativas que provêm uma miscelânea de serviços, jogos, compras pela televisão, anúncios do governo, notícias, etc.

### **Jogos**

Outra forma de aplicação que pode ser disponibilizada na TV interativa é na forma de jogos interativos. Os jogos podem ser usados para entreter, educar, ou mesmo como forma de divulgação de filmes e produtos. Com a sua disponibilização através da TV interativa, pode-se atingir um grande potencial de público.

Uma primeira possibilidade é a transmissão dos jogos para serem jogados individualmente nas *set-top boxes*. Outra possibilidade, que demanda um canal de retorno, é a de jogos onde múltiplos usuários interagem, de forma análoga ao que ocorre na Web.

### **Elementos Mesclados ao Conteúdo da Programação**

Uma forma de serviço também disponível é o uso de elementos interativos mesclados ao conteúdo, ou seja, elementos interativos que aparecem durante a exibição de filmes e programas de televisão, em paralelo ao vídeo exibido. Para fazer uso desses elementos, geralmente é preferível utilizar gráficos semi-transparentes, que aparecem em momentos específicos nos cantos do televisor. É possível, por exemplo, durante a apresentação programa, oferecer ao usuário a possibilidade de comprar um produto que aparece no programa ou de pedir uma pizza quando alguém aparece comendo em um filme. Essas oportunidades para anúncios e comércio pela TV recebem nome de *t-commerce*. Além de servirem para *t-commerce*, os elementos gráficos também podem ser usados para mostrar estatísticas, ou simplesmente qualquer outro recurso onde mais informações estejam disponíveis, caso o usuário realmente as deseje, como funções de histórico em jogos esportivos.

### **Interação com o Conteúdo da Programação**

A perspectiva mais revolucionária para TV interativa é a interação com o conteúdo que é assistido. Essa possibilidade, no entanto, é a mais desafiadora e a mais ligada ao trabalho apresentado nesta dissertação. Procuram-se prover meios para que, por exemplo, o fim de uma história que está sendo assistida sofra a interferência do usuário.

Para a implementação de mecanismos de interação avançada na TV como esses, há, no entanto, dificuldades relacionadas à arquitetura dos sistemas e ao modelo comercial. Por um lado, *set-top boxes* precisam ter algum tipo de padronização e precisa haver um canal de retorno ao provedor do conteúdo. Por outro lado, em ambientes onde a TV é gratuita e financiada por anúncios comerciais, o modelo de negócio atual precisa ser reestudado, pois passa a haver muito mais opções de conteúdo e o conteúdo passa a ser muito mais controlado pelo usuário.

### **3.3 TV Digital Interativa**

A televisão tradicional analógica está com os dias contados no Brasil. O decreto presidencial número 5820, em 29/06/2006, estabelece a data limite de 10 anos até que se encerre a transmissão analógica, que deverá ser substituída totalmente pela transmissão digital. Um aspecto muito importante para a implantação deste sistema é justamente a grande disseminação do meio atual, analógico, no Brasil. Conforme demonstrado por pesquisa do IBGE (IBGE, 2007), a televisão encontra-se em 94,5% dos domicílios particulares permanentes, sendo importante ressaltar que, em contrapartida, apenas 20,2% possuem computador com acesso à Internet. Dessa forma, a TV digital aberta ganha uma importância maior e cabe investigar o potencial de interatividade desse ambiente em particular.

#### **3.3.1 Transmissão de Dados**

Uma das principais mudanças dentro do novo meio digital da iTV é o próprio canal de transmissão, fundamental para a construção da DTV, ou *Digital Television*. O modelo de referência para a DTV na maioria dos sistemas, se faz através do uso de algoritmos de compressão, em especial o MPEG-2 para imagem e MP3 pra áudio, reduzindo o uso da banda enquanto se preserva a qualidade.

Para existir interatividade em uma ambiente de TV Digital, não é estritamente necessária a aquisição de novos televisores digitais. Até televisões analógicas podem ser utilizadas em conjunto com *set-top boxes* ou URDs - “Unidades Receptoras-

decodificadoras” que contenham conversores de sinais digitais para sinais analógicos. *Set-top boxes* (SCHWALB, 2003) têm a capacidade de processamento de sinais de vídeo e áudio, além de poderem executar programas. São capazes de recepção, demodulação, decodificação e remodulação do sinal digital, gerando sinal de áudio e vídeo compatível com televisores analógicos, além de dar suporte às formas de interação denominadas "Pseudo" e "Real".

Na "pseudo" interatividade, o *set-top box* se comunica com a central de produções do canal desejado e processa os fluxos de dados multiplexados, sendo capaz de exibir na televisão uma interface para o usuário, que pode então interagir com o programa de TV através do controle remoto, ou teclado. A interatividade "real" é viabilizada através de uma conexão com um canal de retorno, geralmente por internet ou no caso de TV a cabo por exemplo, transmitindo pelo próprio meio por onde recebe o sinal de TV; e possibilita uma maior gama de oportunidades de interação. É possível a instalação dinâmica no *set-top box* de uma cópia de um sistema de arquivos produzido no estúdio de dados. No *set-top box* podem então ser exibidos textos transmitidos e recebidas aplicações para serem executadas.

O sistema final definido para Brasil foi o ISDB-TB (ISDB-T do Brasil), também denominado SBTVD (Sistema Brasileiro de Televisão Digital) (CPqD, 2006). Ele incorporou atualizações ao modelo japonês nas partes de áudio, vídeo e interatividade e traz uma série de vantagens para a realidade brasileira, como, por exemplo, a redução no pagamento de *royalties* e o preço do *set-top box* mais acessível, o que possibilita uma maior inclusão digital. Ao utilizar *set-top boxes* para a conversão do sinal digital para analógico, o parque instalado é aproveitado, permitindo uma implantação gradual, que leve em conta as diferentes condições sócio-econômicas do Brasil. Com o uso do *set-top box* ligado à internet, é possível ainda utilizar a televisão como um *browser* (contudo, ainda que normalmente mais limitado que o de um PC).

### **3.3.2 Middleware**

Para implementar aplicativos para televisão digital existem duas abordagens principais: o uso de aplicações declarativas ou de aplicações procedurais. Define-se por aplicação declarativa aquela cujo conteúdo é projetado de forma declarativa, enquanto que, nas aplicações procedurais, seu conteúdo é projetado de maneira procedural. Esta diferença de padrões define a própria filosofia e utilidade para cada aplicação feita.

No Brasil, através de uma pesquisa conjunta de diversas instituições foi criado o

*middleware* Ginga, para a implementação de aplicativos interativos para a TV Digital. Em concordância e com a filosofia do Java, o Ginga busca o desenvolvimento independente da plataforma de hardware dos diferentes *set-top boxes*. O Ginga é aplicável aos receptores para sistemas de transmissão terrestre, e busca cumprir completamente uma série de especificações para serem usadas em aparelhos de TV Digital integrados, computadores multimídia e *clusters* locais de aparelhos em redes domésticas (HAN). O núcleo do Ginga, a Ginga-Core é composta por diversos decodificadores de conteúdos comuns, como imagens JPG, PNG, etc. e procedimentos para obter conteúdos transportados em canais (*Streams*) de transporte do MPEG-2 e através do canal de retorno.

O Ginga-NCL (SOARES *et al.*, 2007) é a porção declarativa do Ginga. Ele provê uma estrutura para a apresentação de aplicações declarativas escritas na linguagem NCL. Através de um decodificador, também conhecido como *NCL formatter*, o conteúdo pode ser processado. Outros módulos importantes do Ginga-NCL incluem o agente de usuário, baseado em XHTML, que inclui folhas de estilo (CSS) e um interpretador ECMAScript, além da *engine* LUA, capaz de interpretar *scripts* LUA, muito utilizados em jogos e outros produtos interativos. Além disso, é possível o uso de outros padrões declarativos através de implementações diferentes baseadas em XHTML.

O aspecto procedural do Ginga é implementado através do Ginga-J (SOUZA *et al.*, 2007), o qual inclui diferentes APIs baseadas em padrões JavaTV. O Ginga-J é projetado para ser capaz de suprir as funcionalidades necessárias para a criação de aplicações para a TV Digital, com funcionalidades para protocolos de acesso, interação, manipulação de objetos multimídia, etc. O Ginga-J utiliza aplicativos procedurais, também conhecidos como Xlets Java. Como toda implementação Java, utiliza uma máquina virtual Java (JVM), que estabelece uma camada de abstração sobre o sistema operacional subjacente, o que facilita a compatibilidade e portabilidade das diferentes implementações em diferentes hardwares, dado que se siga corretamente a sua especificação. Uma aplicação Ginga-J deve-se basear nas definições GEM 1.1 (ETSI, 2005) (*Globally Executable MHP*), que são especificações unificadas para a criação de *middlewares* para TV Digital. Uma aplicação compatível com este padrão será, em princípio, compatível com o Ginga-J.

### **3.4 Outros Ambientes e Abordagens para TV Interativa**

Existem outras formas para a transmissão de conteúdo de televisão. Exemplos

desses ambientes são a Internet TV e a IPTV, que se definem como sistemas onde um serviço de televisão digital é transmitido através do protocolo IP (*Internet Protocol*) através de uma infra-estrutura de rede, geralmente sendo entregue através de uma conexão de banda larga, com a diferença que na IPTV a transmissão é numa rede própria, já a Internet TV usa a Internet convencional. Em linhas gerais, a diferença é que, em vez de ser transmitida através dos formatos tradicionais como *broadcast* e TV à cabo, seu conteúdo é recebido através de tecnologias usadas para redes de computadores. Nas plataformas baseadas em IP, surgem diversas oportunidades de interatividade, da mesma forma que nos padrões de televisão digital interativo descritos, como guias de programação, recursos interativos dando informações sobre os programas exibidos, etc.

Um exemplo de implementação de IPTV é o AppleTV (APPLE, 2009). O AppleTV é um aparelho que funciona em rede e que permite que usuários usem um televisor HD para ver fotos, tocar música, e assistir a vídeos de fontes da Internet ou de rede local. Dentre essas fontes, incluem-se a loja iTunes, Youtube, Flickr e MobileMe, por exemplo. Ao se conectar na loja da Apple, a iTunes, os usuários podem comprar e alugar filmes, programas de televisão, músicas, videoclipes, grande parte do conteúdo na qualidade HD.

Projetos como o ShapeShifting Media (URSU *et al*, 2008), questionam as formas de interatividade que vêm sendo criadas no ambiente de televisão interativa digital, como *t-commerce*, *e-mail*, *browsers* de internet, etc., dado que são serviços que, embora aumentem as possibilidades de experiências dos espectadores, são meras adaptações de serviços provenientes de outras plataformas, como a Web. O projeto então busca, através de pesquisa colaborativa, a produção de novas formas de mídia interativas criadas especificamente para a IPTV, em especial usando novas formas de *storytelling* que se adequem às preferências do espectador. O ShapeShifting Media trabalha em cima de modelos de narrativas onde autores definem: uma narrativa global representando o mundo da história; estruturas e regras usadas para transmitir narrativas personalizadas; os objetos narrativos; grafos estruturados que especificam que caminhos a narrativa pode tomar para usuários individuais; pontos de decisão na narrativa global; grupos de objetos narrativos escolhidos segundo regras; e uma estrutura de camadas que estabelece os objetos narrativos que são transmitidos em paralelo.

Dentro da Shapeshifting TV, foram criadas algumas aplicações interessantes:

- Em *Gormenghast Explore*, que é baseada em livro de mesmo nome, ocorre uma

visita interativa a um castelo 3D. Os usuários podem acessar as histórias de diferentes personagens, que são reconfiguradas a cada visita.

- Na aplicação *My News & Sports My Way*, é implementado um arquivo digital interativo que permite a descoberta, seleção e recombinação de notícias. Tenta-se criar um programa de notícias com uma narração contínua, em vez de histórias isoladas, e que dá aos espectadores a habilidade de explorar questões em maiores detalhes quando desejado.
- Em *Accidental Lovers*, foi criada uma comédia romântica de humor negro, que foi transmitida por Internet, telefone celular e televisão. Os usuários acompanhavam em tempo real o desenrolar da relação de um casal, enviando mensagens incentivando ou não a relação dos dois. Depois de moderadas, essas mensagens influenciavam o andar da trama e/ou apareciam na tela durante a exibição do programa.

O projeto ShapeShifting Media tem uma série de semelhanças com a pesquisa desta dissertação, inclusive nas preocupações com a manutenção da continuidade da recepção de conteúdo que é gerado dinamicamente. Para tal, o projeto apresenta como uma opção o uso de pequenos fragmentos de vídeo que são criados no servidor e então enviados em partes, as quais vão sendo apresentadas em seqüência.

### **3.5 Conclusão**

Neste capítulo foi feito um levantamento do cenário atual da TV interativa. Fica bem evidente o fato de que se trata de uma área ainda nova, com muitas pesquisas e padrões diferentes. No mundo todo, as potencialidades das diversas alternativas ainda não estão completamente claras.

Os modelos de programas e as respectivas aplicações em TV interativa ainda precisam ser implementados e testados de modo se verificar a sua atratividade e a capacidade de garantir um grau de responsividade mínimo para programas de TV. O poder de interação precisa ser devidamente balanceado com a facilidade de interação. O fluxo de apresentação dos programas, por outro lado, precisa ser compatível com a produção e a veiculação dinâmica de conteúdo. Por fim, a coerência e diversidade do conteúdo são fatores obviamente importantes, pois, ao contrário do que ocorre em jogos eletrônicos onde o prazer está em vencer desafios e as histórias têm função secundária, o

prazer ao se assistir TV está em acompanhar as histórias. Se essas não tiverem coerência ou se tornarem repetitivas, o interesse do usuário tende a ser facilmente perdido.

No que tange à TV digital interativa, deve-se destacar que a implementação de programação interativa no SBTVD tenderá a ser facilitada com a utilização do Ginga. Há, contudo, desafios a serem vencidos no que tange à comercialização de *set-top boxes*, à criação de modelos de negócio e, finalmente, ao desenvolvimento de aplicações fora da esfera acadêmica pela indústria.

Este trabalho visa utilizar a televisão digital interativa como mote, propondo a criação de um modelo distribuído de *storytelling* interativo compatível com requisitos de responsividade e escalabilidade típicos da TV, com variados níveis de interação (inclusive entre múltiplos usuários) e que garanta a coerência e diversidade das histórias. Para tal, um alto custo de processamento para a criação de enredos e sua dramatização através de animações gráficas 3D é demandado. Tendo em vista o poder computacional limitado das *set-top boxes* e as limitações de banda para transmissão de dados, arquiteturas para a distribuição do processamento e a interação com o conteúdo precisam ser estudadas, como será melhor explicado nos próximos capítulos.

De todo modo, é importante ressaltar que, embora inspirado no contexto de TV digital interativa, o trabalho não está atrelado ao uso de *middleware* específico de TV digital. A adequação do protótipo para a TV digital interativa é uma das possibilidades mais atraentes, mas não é a única, pois os conceitos de TV interativa também podem ser aplicados, por exemplo, à TV via internet e via celular.

No capítulo 4 será apresentado o Logtell, o sistema de *storytelling* interativo usado como base para a implementação das idéias apresentadas nesta dissertação.

## 4. Logtell

Dentre os sistemas para *storytelling* interativo existentes, foi escolhido como base para esta pesquisa o Logtell. Ele apresenta algumas semelhanças e diferenças com os outros sistemas de *storytelling*, porém contém uma série de características que são compatíveis o objetivo deste trabalho: demonstrar a viabilidade de um modelo de *storytelling* interativo que concilie os requisitos demandados por um ambiente de TV interativa. Neste modelo, é preciso que a coerência e a diversidade dos enredos sejam garantidas, pois a importância dos enredos tende a ser maior nesse ambiente do que em jogos, por exemplo. Por outro lado, mecanismos de interação adequados precisam ser oferecidos de modo a não demandar grande esforço de atenção. Os processos de geração, interação e apresentação das histórias precisam ocorrer em paralelo, em tempo real e atendendo a requisitos de responsividade próprios do meio de comunicação. Por fim, o modelo deve prever uma arquitetura distribuída e escalável.

O Logtell (POZZER 2005) (CIARLINI *et al.* 2005) é um sistema de *storytelling* interativo que vem sendo desenvolvido já há algum tempo. A sua abordagem parece ser boa para TV interativa, pois tem o foco na manutenção de histórias coerentes, mas permitindo que o usuário possa interferir na história em diferentes níveis. O usuário pode escolher não interferir na história ou ter uma participação mais forte, estabelecendo, por exemplo, que certos eventos ou situações ocorram, desde que sejam logicamente coerentes com o modelo especificado para o gênero de história que está sendo usado. Uma das principais características do Logtell atual que não é apropriada para a televisão interativa é que, nele, a geração e a dramatização da história são separadas, concentrando-se a interatividade na fase de geração. Para levar o Logtell para um ambiente de TV interativa, é preciso realizar essas atividades em paralelo, encontrando mecanismos confortáveis para a interação.

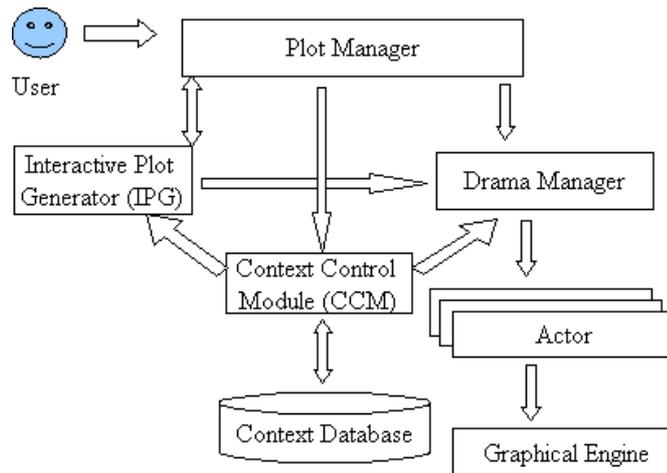


Figura 4.1: Arquitetura inicial do Logtell

Na arquitetura inicial do Logtell, as responsabilidades são divididas entre diferentes módulos, responsáveis pela geração, interação e visualização de histórias, como demonstrado na Figura 4.1. A seguir são apresentados esses processos indicando o papel de cada módulo.

#### 4.1 Geração de histórias

A abordagem utilizada pelo Logtell procura gerar e dramatizar histórias variadas de um determinado gênero, tomando como base a especificação deste por meio de lógica formal. A idéia principal é permitir que o usuário possa interferir na história desde que se mantenha a coerência das histórias.

O Logtell, ao privilegiar a coerência lógica na sua estratégia de geração de narrativas, tem um aspecto predominantemente *plot-based*, mas que é conciliado com características *character-based*. O Logtell inspirou-se inicialmente nas idéias de Propp, estendendo, porém, a sua noção um tanto informal de funções, adotada em sua pesquisa. Os eventos típicos são descritos através de operações parametrizadas com pré-condições e pós-condições, de forma a serem aplicáveis por algoritmos de planejamento. A faceta *character-based* é definida através de regras de inferência de objetivos que fornecem os objetivos a serem alcançados pelos personagens quando determinadas situações são observadas.

As funções *proppianas* são base para a geração de enredos em diversas pesquisas de *storytelling* interativo, conforme mencionado no capítulo 2. No trabalho de Propp, um conjunto grande enredos de contos de fadas russos foi analisado e daí foram extraídas as funções que correspondem a eventos típicos que tendem a se combinar

segundo certos padrões. De forma análoga, outros gêneros literários podem ser analisados de modo a se extraírem os eventos típicos e os padrões de combinação.

Em histórias convencionais, os eventos que acontecem geralmente não ocorrem de forma caótica e aleatória. Na maioria das histórias convencionais interessantes, há sempre uma lógica por trás dos eventos, que em geral é tácita, mas compreensível. Espectadores/leitores já acostumados com histórias de um determinado gênero podem facilmente determinar se certa história está de acordo com os padrões de lógica implícitos no gênero. Eventos ocorrem para que objetivos (dos personagens ou da história como um todo) sejam atingidos ou para produzir as condições que permitam o acontecimento de outros eventos. Os eventos, por sua vez, acabam modificando o mundo, o que pode propiciar o aparecimento de novos objetivos que fazem com que a história se desenvolva. Dessa forma, se um modelo para a geração automática de enredos deseja garantir a coerência das narrativas, esse modelo precisa, de algum modo, capturar a “lógica” embutida no gênero com que trabalha.

O contexto das histórias no Logtell contém as seguintes informações sobre a lógica dos enredos a serem gerados:

- um conjunto de operações parametrizadas com pré- e pós-condições, especificando logicamente quais eventos podem ocorrer
- um conjunto de regras de inferências de objetivos especificadas com uma lógica temporal modal, declarando situações que levam os personagens a buscarem objetivos; e
- um conjunto de fatos que definem a configuração inicial das histórias a serem geradas.

A experiência de *storytelling* interativo no Logtell possui uma diferença de paradigma nítida tanto em relação a jogos quanto à experiência de se ler um livro ou assistir a um filme. Em jogos, o usuário interage constantemente, mas o conteúdo da história tem importância secundária. Na experiência de se assistir a um filme ou ler um livro, o conteúdo das histórias é fundamental, mas não há interação. No Logtell, o usuário passa a ser também um pouco autor, na medida em que as possíveis seqüências de eventos não são estabelecidas previamente e podem ser decisivamente influenciadas por suas intervenções. Para que histórias interessantes surjam dinamicamente, contudo, é necessário um esforço autoral prévio para definir um bom conjunto de regras e

eventos.

Os enredos no Logtell são gerados por um módulo implementado em Prolog, chamado IPG (*Interactive Plot Generator*) (CIARLINI 1999). O IPG gera enredos em múltiplos estágios que alternam fases de inferência de objetivos, planejamento e interação com o usuário.

O IPG possui dois submódulos: um para inferência de objetivos e um para planejamento. O seu planejador é uma extensão do Abtweak (YANG *et al.*, 1996), que é um planejador hierárquico e não-linear. O planejador permite a definição de uma hierarquia de pré-condições, de modo que haja uma priorização na busca de soluções. Por outro lado, por ser um planejador não-linear (ou planejador que planeja no espaço de planos) trabalha com ordens parciais de eventos. No planejamento no espaço de planos, adota-se uma abordagem de comprometimento mínimo, segundo a qual relações de ordem entre os eventos e restrições sobre os valores de variáveis são estabelecidos apenas quando estritamente necessárias. A abordagem adotada facilita, em especial, a conciliação de múltiplos objetivos, a geração de alternativas e a flexibilidade para a dramatização dos enredos. O planejador do IPG estendeu o Abtweak com a incorporação de *Constraint Logic Programming* (MARRIOT e STUCKEY, 1998), que facilita o tratamento de pré-condições envolvendo expressões numéricas, e conceitos que permitem o abandono de objetivos.

A geração de uma história começa com a inferência de objetivos dos personagens ou da história a partir da configuração inicial. O sistema então usa o planejador que, respeitando as pré-condições e pós-condições, insere eventos no enredo para permitir o alcance dos objetivos. Quando o planejador percebe que todos os objetivos foram alcançados ou abandonados, o primeiro estágio do processo termina. O enredo parcial então é apresentado através do *Plot Manager* e pode, se o usuário assim desejar, ser dramatizado. Quando o usuário não gosta do enredo parcial, pode solicitar ao IPG a geração uma alternativa. Se o usuário aceita o enredo gerado até então, o processo continua, inferindo novos objetivos, decorrentes de situações criadas pelos eventos inseridos na etapa anterior. Se novos objetivos são inferidos, o planejador é utilizado novamente, e assim sucessivamente a história vai sendo gerada. O processo continua até o momento em que o usuário decide parar ou nenhum objetivo novo é inferido. Nas fases de interação, o usuário pode também tentar inserir forçosamente eventos ou situações. Se essas inserções respeitam a lógica do gênero, o enredo é adaptado para acomodá-las, caso contrário são rejeitadas.

Note que, nesse processo de geração das histórias, são usados tanto *forward* quanto *backward reasoning*. Na fase de inferência de objetivos, é usado o *forward reasoning*: as situações do passado geram objetivos a serem alcançados no futuro. Na fase de planejamento, um evento inserido no enredo, para realizar algum objetivo, pode ter pré-condições não satisfeitas, tratadas através de *backward reasoning*. Para estabelecê-las, o planejador pode inserir eventos anteriores com suas próprias pré-condições não satisfeitas e assim sucessivamente.

A arquitetura do Logtell prevê que as informações de contexto sejam armazenadas em um banco de dados acessado através do *Context Control Module* (CCM). O CCM provê facilidades de interface e de armazenamento para a autoria de contextos, mas não chegou a ser plenamente integrado ao Logtell. Esse módulo é capaz de gerar um arquivo de contexto em Prolog com as informações necessárias para o IPG. Em futuras versões, o CCM deverá ser plenamente integrado na arquitetura, armazenando também informações relativas à dramatização dos enredos.

As configurações iniciais e os demais estados das histórias são compostos por conjuntos de fatos que descrevem a situação de personagens e lugares e os seus relacionamentos. No Prolog, um fato expressando, por exemplo, que o *lugar atual de Brian é o Gray Castle* seria denotado pela cláusula de predicado `current_place('Brian','Gray_Castle')`, aonde *Brian* é um personagem e *Gray Castle* é um local (usados no contexto testado usado na pesquisa). Os fatos da configuração inicial podem ser modificados pelos eventos gerados (que são as execuções dos operadores do planejador), conforme a história se desenrola. Na tabela 4.1, há uma lista dos predicados Prolog usados para representar situações no contexto exemplo de contos de fadas (subgênero “espadas e dragões”) usado no Logtell.

<b>Predicado</b>	<b>Descrição</b>
<i>knight</i>	Indica que um personagem é um cavaleiro
<i>princess</i>	Indica que um personagem é uma princesa
<i>magician</i>	Indica que um personagem é um mago
<i>dragon</i>	Indica que um personagem é um dragão
<i>nature</i>	Indica a natureza de um personagem (se é bom, mau, ou neutro)
<i>strength</i>	Indica a força de um personagem
<i>alive</i>	Indica se um personagem está vivo

<i>place</i>	Indica a existência de um local
<i>protection</i>	Indica o nível de proteção de um local
<i>home</i>	Indica o local inicial de um personagem
<i>current_place</i>	Indica o local atual de um personagem
<i>affection</i>	Indica o nível de afeição de um personagem por outro
<i>hero</i>	Indica que um personagem é um herói
<i>victim</i>	Indica que um personagem é uma vítima
<i>villain</i>	Indica que um personagem é um vilão
<i>donor</i>	Indica que um personagem é um doador (de força)

Tabela 4.1: Predicados usados no contexto exemplo do Logtell e suas descrições

Os eventos inseridos na história são instâncias das operações definidas no contexto. Uma operação é um padrão de evento típico do gênero (contendo variáveis). Pré-condições e pós-condições são especificadas usando os predicados de descrição de situações do contexto. Na tabela 4.2 temos uma lista dos possíveis eventos que podem ocorrer no contexto exemplo.

<b>Evento</b>	<b>Descrição</b>
<i>go</i>	Indica que um personagem vai a algum local
<i>reduce_protection</i>	Indica que um personagem reduz a proteção de um local (ex: dispensa os guardas)
<i>kidnap</i>	Indica que um personagem seqüestra o outro
<i>attack</i>	Indica que um personagem ataca as defesas de um local
<i>fight</i>	Indica que um personagem ataca outro
<i>kill</i>	Indica que um personagem mata outro
<i>free</i>	Indica que um personagem liberta outro
<i>marry</i>	Indica que um personagem casa-se com outro
<i>donate</i>	Indica que um personagem aumenta a força de outro
<i>bewitch</i>	Indica que personagem enfeitiça o outro

Tabela 4.2: Lista de Eventos possíveis no contexto exemplo do Logtell

As regras de inferência de objetivos usadas pelo IPG, especificadas num formalismo de lógica temporal modal (CIARLINI *et al.*, 2000), definem os objetivos que os personagens de determinadas classes (herói, vilão, vítima) passam a perseguir quando certas situações se observam. Essas regras usam meta-predicados para falar sobre a ocorrência de um evento em um determinado tempo ou sobre a veracidade de um fato (ou sua negação) em determinado momento. O formalismo incorpora o tratamento das ordens parciais de eventos, uma vez que a veracidade de um fato pode depender ou não da ordem dos eventos.

Uma regra de inferência de objetivos tirada do contexto exemplo diz que “se uma vítima fica desprotegida e há um vilão, este vai querer raptá-la”. Outra regra, quase que complementar, diz que “se uma vítima é raptada e há um herói que gosta dela, este vai querer resgatá-la”. É importante ressaltar que as regras não ditam diretamente as reações dos personagens. As regras apenas indicam objetivos a serem perseguidos. Os eventos que vão conseguir realizar os objetivos são preenchidos pelo algoritmo de planejamento.

A tabela 4.3 a seguir apresenta uma lista com as regras que foram modeladas para o contexto exemplo.

<b>Regra</b>	<b>Descrição</b>
<i>The strongest hero wants to become stronger than the villain</i>	Quando existir um herói e um vilão em um determinado momento, e a força do vilão for maior do que a do herói, este tentará ficar mais forte
<i>Victim spontaneously reduces the protection at her current location</i>	Uma vítima, quando em local da mesma natureza que a sua (boa, má, etc), por algum motivo tolo acaba ficando em uma situação desprotegida, ou seja, se põe em perigo
<i>If victim's protection is reduced, villain will want to kidnap her</i>	Se uma vítima estiver num local suficientemente desprotegido, um vilão tentará seqüestrá-la
<i>If victim is kidnapped, hero will want to rescue her</i>	Se uma vítima for seqüestrada, um herói tentará libertá-la
<i>If victim is killed, hero will want to avenge her</i>	Se uma vítima for morta, o herói tentará vingá-la
<i>If the affection between two persons is high they will want to get married</i>	Se a afeição entre 2 personagens for grande eles vão querer se casar

Tabela 4.3: Lista de regras no exemplo do Logtell

## 4.2 Interação com o Usuário

O foco principal do sistema no quesito interatividade é dar os meios para o usuário explorar qualquer alternativa de história coerente com o contexto especificado logicamente. Para garantir a coerência das histórias, a interação é sempre indireta. O usuário não intervém na história como um personagem nem manipula os objetos do cenário diretamente. Por outro lado, ele pode selecionar alternativas e interagir de forma mais ativa com intervenções fortes, sujeitas à validação pelo IPG.

Na versão inicial do Logtell, a interação com o usuário ocorre através do *Plot Manager* que é um módulo implementado em Java. O *Plot Manager* é também responsável pelo controle geral da experiência de *storytelling* interativo, acionando o IPG para gerar enredos e o *Drama Manager* para dramatizá-los. A experiência de interação nessa versão ocorre apenas através de um processo de simulação *passo-a-passo*, pois a dramatização da história não ocorre em paralelo com a geração do enredo. Partes do enredo são geradas e apresentadas ao usuário, que tem a oportunidade de interagir e de solicitar a dramatização do enredo gerado até então, podendo retomar o processo de geração em seguida.

O *Plot Manager* possui uma interface gráfica, mostrada na Figura 4.2. Os eventos e objetivos do enredo parcialmente gerado são informados pelo IPG ao *Plot Manager* que os apresenta sob a forma de um grafo onde as restrições de ordem temporal entre os eventos são explicitadas.

Para acomodar mais facilmente a busca por objetivos diferentes, o IPG trabalha com ordens parciais dos eventos, onde é estabelecido que um evento deve preceder outro apenas quando necessário. Para que a dramatização seja possível, no entanto, é preciso que os eventos estejam totalmente ordenados, de forma compatível com as restrições de ordem já impostas pelo IPG. Para determinar a seqüência, o usuário traça arestas conectando os eventos em uma ordem de seu gosto. A seqüência encadeada, corresponda ela a um enredo parcial ou total, pode ser dramatizada usando o comando *render*.

A interação utiliza dois comandos principais da interface: o comando *another* e o comando *continue*. O comando *another* pede ao Logtell que faça um retrocesso e forneça outra alternativa para o passo de simulação recentemente terminado, mas que ainda não foi confirmado. O comando *continue* confirma o enredo até o ponto

apresentado e solicita a continuidade do processo de simulação passo-a-passo, com nova inferência de objetivos e planejamento subsequente.

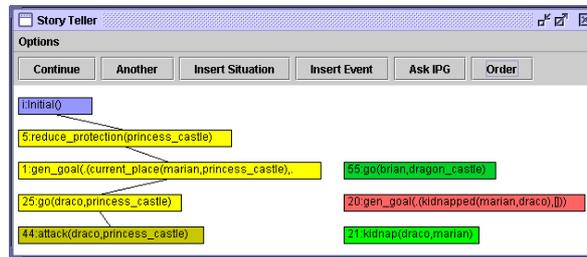


Figura 4.2: Interação Passo a Passo

As intervenções do usuário podem ser classificadas como fracas ou fortes. Na intervenção fraca, o usuário aciona apenas os dois comandos citados anteriormente e, para permitir a dramatização, adiciona restrições de precedência entre os eventos. A definição de restrições de ordens é encarada como uma intervenção (no caso fraca) porque, além de indicar a ordem de dramatização, pode levar a mudanças na inferência futura de objetivos, se o processo de geração for continuado. Os enredos obtidos apenas com intervenções fracas tendem a não se afastar muito das histórias típicas do gênero.

Intervenções fortes podem ocorrer de duas maneiras. Uma delas é com o comando *insert situation*, que permite a especificação de objetivos a serem alcançados, ficando a forma como serão alcançados a cargo do IPG. Note que a situação inserida pode falhar caso não se ache nenhum plano possível ou se o esforço computacional exceder a configuração inicial do planejador. Outro modo de se fazer interações fortes é a inserção explícita de eventos, usando o comando *insert event*. Assim como no caso anterior, o comando *continue* deve ser usado para validar as interações desejadas. O usuário pode também desfazer a intervenção, removendo os eventos ainda não incorporados e/ou rejeitados.

### 4.3 Dramatização

A dramatização é feita pelo *Drama Manager* usando um motor gráfico próprio (POZZER, 2005), implementado em C++. O motor gráfico usa a API OpenGL para suportar a renderização em tempo real dos elementos 3D. Os personagens de um enredo gerado são interpretados por atores virtuais. Cada ator é implementado como um agente reativo 3D virtual. A dramatização não realiza nenhum processamento inteligente no que diz respeito ao enredo, pois segue a seqüência ordenada de eventos gerados previamente

pelo processo de simulação. O *Drama Manager* traduz eventos simbólicos em animações visuais, garantindo a sincronia e a coerência entre o contexto lógico e a sua representação gráfica. Cada evento é renderizado, em tempo real, sincronizando as ações dos personagens e controlando a interação deles com o cenário. Outros aspectos da representação gráfica, como a renderização de legendas, enviadas pelo *Plot Manager*, também são de responsabilidade do *Drama Manager*.



Figura 4.3: História sendo dramatizada no Logtell

Durante a geração dos enredos, a duração de cada evento não é considerada. A mudança no mundo da história causada por um evento é considerada como instantânea para fins de planejamento e inferência de objetivos. Para propósitos de dramatização, o conceito de tempo é um pouco diferente. Eventos precisam ter uma duração de modo que possam ser compreendidos pelo usuário. Atributos variáveis mudam conforme um evento é representado. Para fazer com que as representações lógica e gráfica sejam compatíveis, os valores dos atributos antes da dramatização de cada evento devem ser válidos em relação às pré-condições do evento e os valores ao final de sua dramatização, compatíveis com suas pós-condições. O *Drama Manager* se encarrega de garantir essa compatibilidade. Ao inicializar o motor gráfico, o *Drama Manager* consulta o IPG sobre o estado dos atores, locais e outros atributos necessários. Feito isto, o sistema é capaz de manter-se consistente com o contexto lógico.

O processo de dramatização inicia com a seleção de um evento pelo usuário e o acionamento do comando *render* na interface do *Plot Manager*. A partir do ponto selecionado, a história será exibida, respeitando-se a ordem dada aos eventos pelo usuário, a menos que o usuário a interrompa. Para a renderização, é preciso definir os valores corretos dos atributos gráficos. No caso da renderização a partir do primeiro

evento da história, os valores são definidos através de consulta ao IPG como já foi explicado. Renderizações a partir de pontos intermediários também são possíveis, mas o evento correspondente a esse ponto precisa já ter sido previamente renderizado. Isso é possível porque, quando cada evento começa a ser apresentado, o motor guarda, em um banco de dados local, todos os atributos de todos os atores e objetos em cena. É usada, como chave primária para a recuperação dos valores, o momento do evento (representado por seu *timestamp*, na terminologia mais usual de bancos de dados).

O usuário pode sempre alternar entre a geração do enredo e sua dramatização. Nesse caso, depois da dramatização, novos eventos e restrições temporais de ordem entre os eventos podem ser adicionadas pelo usuário ou pelo próprio IPG. Se a dramatização for reativada, ela pode somente iniciar em eventos que ocorrem antes das modificações.

O *Drama Manager* converte todos os eventos em ações de menor granularidade, que são delegadas aos atores específicos. Quando um evento termina, o *Drama Manager* pergunta ao *Plot Manager* qual será o próximo evento a ser apresentado. Nesse momento, também é enviada ao *Drama Manager* a legenda que descreve o próximo evento. Se não houver mais eventos a serem apresentados, a dramatização termina. A dramatização de um evento termina quando os atores envolvidos representam as animações associadas à sua execução. Nos testes realizados com o contexto exemplo, a dramatização de cada evento pode levar de alguns segundos a alguns minutos, dependendo da complexidade do evento.

O motor gráfico utiliza cenários específicos para a representação gráfica das histórias. Esses cenários são representados por modelos 3D em um ambiente apropriado para os eventos e personagens que a história pode conter, de acordo com as convenções do gênero, como castelos e florestas em contos de fada.

Como a maioria dos eventos tem relação direta com os locais onde acontecem, os atores devem ser controlados restritamente conforme se movem durante os eventos, garantindo a coerência com o enredo enquanto o interpretam. As construções, tais como castelos e igrejas, servem não apenas para embelezar o cenário. Elas são fundamentais para a orientação das posições dos personagens, o tratamento de colisões, e o condicionamento da forma como os atores se movimentam. O motor gráfico usa um planejamento de baixo nível baseado em *waypoints* (POZZER *et al.*, 2004) e a técnica de *terrain reasoning*, para controlar a movimentação dos atores.

Durante a representação gráfica, o *Drama Manager* atua como um diretor,

coordenando as seqüências de ações lineares ou paralelas, realizadas pelo elenco de atores. O *Drama Manager* monitora o processo da dramatização, ativando novas tarefas na ordem em que as anteriores terminam. Também se inclui nas suas responsabilidades, o controle de câmera, que o Logtell também permite que seja exercido pelo usuário.

Além dos atores que representam os personagens da história, o *Drama Manager* também inclui figurantes em suas representações. Esses figurantes ajudam a enriquecer a experiência de visualização da história e possuem um certo grau de autonomia, porém restrito a seus papéis. No contexto exemplo de “espadas e dragões”, esses figurantes são os soldados, que servem como forma de representação do nível de proteção de determinados locais, como guardas de um castelo, por exemplo. Dessa forma, podem ajudar a complementar a história sem causarem grandes mudanças na sua estrutura dramática. Por não serem de fato personagens, reduzem o esforço computacional na criação do enredo, pois, para o IPG, eles não existem. Além disso, trata-se de um recurso largamente utilizado em outras mídias narrativas como filmes e séries de televisão, em momentos em que se precisa contextualizar o enredo com figuras humanas realizando determinadas ações, mas sem um papel dramático de fato.

#### **4.4 Demais Aspectos**

Diversas outras pesquisas na área de *storytelling* interativo foram e vêm sendo desenvolvidas utilizando o Logtell e o IPG, das quais poderão resultar mecanismos a serem incorporados ao sistema no futuro.

Na seqüência da implementação do IPG, foi feito um trabalho inicial em (FURTADO e CIARLINI, 2000) para a narração dos enredos na forma de textos. As narrativas geradas tinham uma boa dose de redundância, mas o trabalho mostrou a viabilidade da utilização das informações lógicas contidas no enredo para a obtenção de textos que contam as histórias.

O IPG contém um mecanismo de reconhecimento de planos típicos com base na observação de eventos. Planos típicos podem ser incorporados, adaptados e estendidos nos enredos. Foi feito inicialmente um estudo sobre a criação automática de bibliotecas de planos típicos (FURTADO e CIARLINI, 2001). Melhorias para a interface atual do Logtell baseadas também no uso de planos típicos e no seu reconhecimento pelo IPG (KARLSSON *et al.*, 2006) também vêm sendo estudadas. Elas permitirão que, durante o processo de criação, novos comandos interativos incorporem ao enredo estruturas recorrentes nos gêneros narrativos (motivos).

Em (RODRIGUES *et al.*, 2006) é descrita a criação de uma narradora virtual em 3D, capaz de expressar determinadas emoções conforme a história é representada, o que é uma alternativa interessante, particularmente no que tange ao uso de *storytelling* interativo com intuítos educacionais ou para crianças. Esse trabalho abre também possibilidades de investigação interessantes no que se refere ao uso e à representação lógica de emoções em *storytelling* interativo.

Em (DORIA *et al.*, 2008), um novo modelo para controlar o processo de dramatização de histórias interativas é proposto e aplicado no Logtell. Esse modelo se baseia na associação de cada evento a um autômato não-determinístico que descreve ações mais básicas. Através disso, procura-se obter maior variedade na dramatização e maior flexibilidade para o controle do tempo de cada evento. Abre-se também a possibilidade de algum tipo de interação também no nível de dramatização, mantendo, porém, a coerência com o enredo.

Ainda restam muitos outros aspectos a serem pesquisados, como, por exemplo, um melhor uso das câmeras, o uso de músicas e trilhas sonoras, diálogos e todos os outros aspectos que fazem parte da experiência multimídia já presente em outros meios como em filmes, jogos e programas de televisão. Grande parte desses aspectos passa pela dificuldade de se modelar e tratar adequadamente emoções tanto na geração de enredos quanto na sua dramatização.

Por fim, um grande desafio é a questão autoral: o contexto exemplo de “Espadas e Dragões” tem servido para testar conceitos e avaliar questões fundamentais. No entanto, a produção em larga escala de enredos interativos demanda a implementação de ferramentas que facilitem a especificação do contexto das histórias.

#### **4.5 Conclusões**

Este capítulo demonstrou como funciona o Logtell, explicando o seu modelo de geração e dramatização de histórias interativas. O Logtell é um sistema de *storytelling* interativo que apresenta algumas características interessantes para a utilização em um ambiente de TV interativa.

Em primeiro lugar, ao gerar variados enredos logicamente consistentes por construção, possibilita um melhor atendimento ao requisito de coerência da apresentação de narrativas na TV. Em segundo lugar, ao contar histórias interativas em terceira pessoa, facilita o compartilhamento de uma mesma história por vários usuários.

Para se poder usar sistemas de *storytelling* interativo em mídias como a TV

interativa, além da dificuldade de se equilibrar coerência e interatividade, deve-se considerar que se precisa encontrar uma forma híbrida de apresentação de conteúdo multimídia, misturando características de jogos e televisão normal. É necessário manter o apelo para o espectador comum, e, ao mesmo tempo, permitir que o usuário seja capaz de interagir de diferentes maneiras com o conteúdo que lhe é apresentado. Um sistema de *storytelling* interativo, quando aplicado a este caso, deve inclusive considerar o fato de que o usuário pode nem mesmo querer interagir de modo algum. O sistema precisa ter tanta responsividade quanto possível e a história deve ser fluente, sempre mantendo a simplicidade e a sensação de conforto na interação, especialmente considerando-se que as histórias em geral serão assistidas por espectadores que podem não ser ávidos usuários de jogos eletrônicos. Ao usuário deve ser permitido interagir de acordo com seu gosto. Dessa forma, o Logtell, ao possibilitar variados níveis de interação, dá uma boa base para se tentar obter mecanismos de interação adequados para diferentes tipos de usuários.

Uma das características do Logtell atual que não é apropriada para a TV interativa é que a geração e a dramatização da história são separadas, concentrando-se a interatividade na fase de geração. Para levar o Logtell para um ambiente de TV interativa, é preciso realizar essas atividades em paralelo, encontrando mecanismos confortáveis para a interação.

Para isso, no próximo capítulo, é apresentado um novo modelo de sistema de *storytelling* interativo adequado para a TV, onde o usuário pode assistir e interagir com a história em paralelo com a geração do enredo, mantendo um fluxo contínuo de apresentação. Além disso, o modelo procura ser escalável para um número grande de usuários.

## 5. Modelo de *Storytelling* Interativo para TV

Neste capítulo, é apresentada uma nova proposta de modelo de *storytelling* interativo que procura ser compatível com um ambiente de TV interativa (CAMANHO *et al.*, 2008). Para tal, esse modelo de *storytelling* deve:

- possibilitar a geração de histórias variadas, coerentes e que resultem da interação com o usuário;
- permitir um fluxo contínuo, onde a criação de histórias ocorra em paralelo com a sua apresentação e a interação com o usuário;
- possuir uma arquitetura escalável;
- apresentar novas maneiras de interatividade, adequadas a diferentes modelos de espectador e formas de apresentação, tendo em mente especialmente modelos que sejam apropriados para a televisão interativa.

Como ponto de partida para a definição do modelo, usou-se a abordagem adotada na primeira versão do Logtell. O Logtell, com sua ênfase em uma simulação baseada em lógica para a geração dos enredos, tem um bom potencial para atender ao requisito de coerência. Por outro lado, ao oferecer tipos de intervenção em variados níveis, fornece flexibilidade para que o usuário possa interagir tanto de forma mais ativa quanto mais passiva. Por outro lado, o modelo do Logtell precisa ser devidamente modificado para se obter um resultado compatível com requisitos de responsividade, como a continuidade do fluxo e a comodidade durante a interação.

No restante deste capítulo, discorreremos sobre os requisitos do modelo e então apresentamos uma nova arquitetura projetada para atender a esses requisitos. Em seguida, detalhamos mecanismos básicos para a interação no novo modelo, a coordenação da interação com múltiplos usuários e possíveis estratégias para a manutenção de um fluxo contínuo de apresentação. Por fim, abordamos outros mecanismos avançados de interação que podem ser viabilizados.

### 5.1 Requisitos

#### 5.1.2 Fluxo Contínuo

No intuito de proporcionar ao usuário final, que assiste à TV interativa, uma

experiência agradável, é necessário que o conteúdo seja apresentado de forma contínua, sem interrupções, e que o usuário possa interagir a qualquer momento. Para tal, são necessárias estratégias de coordenação, de modo que o conteúdo da história seja gerado em partes e que, enquanto uma parte do enredo é dramatizada, novas partes sejam produzidas. Na primeira versão do Logtell, o enredo é gerado em separado da dramatização. Entre uma etapa e outra da geração, o usuário pode interagir e até acionar a dramatização, para depois voltar ao processo de geração. O autor do contexto pode definir regras e explorar a coerência lógica dessas regras verificando apenas o enredo, descrito sob a forma de um grafo, sem ter que acionar a dramatização. Para o usuário comum, que normalmente não seria o autor do contexto, esse processo, com o fluxo interrompido, ainda mais em um ambiente onde está acostumado a “assistir às histórias”, parece ser pouco usual. O usuário se sente mais autor do que espectador, por ter que preparar o enredo e somente então assistir à sua dramatização.

Com a implementação de um fluxo contínuo no processo de geração e dramatização da história, obtém-se experiências potencialmente mais interessantes para o usuário final da TV interativa, pois expandem-se as possibilidades de interatividade.

O usuário passa a de fato assistir a uma história, e, segundo seus desejos, modificá-la, se assim for do seu interesse. O usuário pode ser um espectador passivo na maior parte do tempo, se assim o desejar, aderindo-se dessa forma ao conceito de interatividade preguiçosa, citado anteriormente, e que tem sido adotado em algumas outras aplicações para a TV interativa. Por outro lado, intervenções fortes, no estilo do Logtell, podem ser feitas com pouco esforço, levando o enredo para caminhos completamente distintos. Além disso, intervenções podem acontecer diversas vezes e a qualquer momento, possibilitando que usuário assuma uma postura mais ativa.

Para que a idéia de fluxo contínuo fosse viável, introduziu-se no novo modelo o conceito de capítulos. Esse conceito, próprio da estrutura de narrativas, já existia de certa forma no modo passo-a-passo do Logtell, pois uma fase de geração era composta pela interação com o usuário, seguida da inferência de objetivos e do planejamento, vindo depois uma nova fase de geração. Um capítulo, no fluxo contínuo, corresponde a uma fase de geração seguida de uma ordenação automática dos eventos, ou seja é um conjunto de eventos ordenados que se traduziram em cenas (uma por evento). A ordenação dos eventos precisa ser automática, dado que o usuário não tem tempo para fazer isso, como acontece no modo passo-a-passo. Como a história é apresentada em paralelo, assume-se que qualquer interação só pode mudar capítulos posteriores, pois

não faz sentido mudar a parte da história que já se assistiu (exceto se o usuário desejar repetir todo o processo de geração). Um capítulo delimita então um pedaço da história onde um ou mais objetivos foram inferidos ou decorreram de intervenções do usuário, e, por meio de planejamento, eventos foram inseridos para alcançar esses objetivos, os quais receberam automaticamente uma ordenação total, compatível com a ordem parcial produzida pelo planejador.

Uma das maiores preocupações no que tange à continuidade do fluxo é evitar interrupções durante o processo de geração e dramatização das histórias. Como será explicado em maiores detalhes adiante, estratégias específicas precisam ser adotadas para escrever a história que está por vir com pelo menos um capítulo de folga à frente do capítulo que o usuário esteja assistindo. Mesmo assim, há também a possibilidade de que, devido ao excesso de carga, e também devido ao fato de que o processo de geração do enredo ser custoso em termos de processamento, existam breves interrupções. Para lidar com esse problema, é possível pensar em mecanismos que alonguem o capítulo corrente através de uma dramatização mais flexível, que receba comandos para acelerar ou retardar os eventos conforme a necessidade. Outras opções correspondem à introdução de propagandas, num contexto comercial, ou ainda à introdução de *fillers*, ou seja, momentos em que nada de significativamente importante acontece, mas imagens e sons são exibidos, no mesmo estilo do que é feito em filmes, novelas, etc., quando se mostram paisagens, cenários etc.

### **5.1.3 Diversidade de Enredos**

Para que o modelo proposto proporcione uma experiência interessante na TV interativa, também é importante que o sistema utilizado para a geração das histórias tenha um mínimo de diversidade nos enredos gerados. O usuário facilmente se cansará de assistir às histórias interativas, se não houver a oportunidade de interagir com diferentes histórias.

Aplicações como o Façade, citado no capítulo 2, por exemplo, são capazes de prover uma experiência interessante de aproximadamente 20 minutos, porém o seu grau de variação é um tanto limitado, por conter basicamente apenas uma situação dramática. A sua estrutura não é muito generalista e, para a criação de novas histórias, imagina-se que existirá um grande esforço, dado que na história atual houve um forte trabalho autoral e muitas linhas de código tiveram que ser escritas para suportar toda a estrutura dramática da situação modelada.

Somado a isso, deve-se ter a garantia, em um ambiente como o da TV, onde a qualidade da história tem importância maior do que em jogos eletrônicos, por exemplo, de que as histórias não perderão a coerência com os pressupostos do gênero. Há uma intrínseca impossibilidade de se manter a interatividade e a coerência em plenitude, dada a natureza um tanto contraditória desses dois atributos. É, portanto, admissível que se restrinjam as interações do usuário quando elas não forem possíveis ou não fizerem sentido.

Nesta pesquisa, ao se adotar o Logtell como base, usufrui-se da vantagem de seu motor gerador de enredos, o IPG, trabalhar com uma simulação baseada em lógica formal que concilia abordagens *plot-based* e *character-based*. Ao combinar regras de inferência de objetivos (dos personagens e da história) e planejamento, consegue-se uma boa variedade de enredos em um único contexto, todos eles coerentes por construção. Praticamente qualquer enredo coerente com o gênero pode ser obtido. A conciliação da interatividade com a coerência é feita simplesmente rejeitando intervenções que o usuário tente fazer que não sejam lógicas. Continua sendo necessário, no entanto, um forte esforço autoral (como, de fato, é exigido em qualquer história, seja ela interativa ou não). A diferença é que, ao se modelar um contexto logicamente, os variados enredos que podem ser obtidos não precisam necessariamente ter sido considerados previamente pelo autor.

#### **5.1.4 Coexistência com Modo Passo-a-passo**

O modelo de *storytelling* da primeira versão do Logtell previa a geração dos enredos alternando fases de simulação e interação.

Nas fases de interação, era possível assistir, se desejado, à dramatização do enredo gerado até o momento, mas sem interação enquanto a história era apresentada. Através desse processo, o usuário pode, com calma, definir exatamente o que quer, de acordo com o enredo que é obtido através do sistema. Pode também investigar a razão pela qual as regras lógicas em conjunto com a interação guiam a história em um certo rumo.

Tal forma de interação é justificada por seus diferentes usos possíveis, especialmente se aplicado num contexto não necessariamente de entretenimento. Por exemplo, o próprio IPG já foi pensado também como uma ferramenta para sistemas de apoio à simulação em geral (como ambientes corporativos) e como ferramenta para educação e a tomada de decisão (CIARLINI e FURTADO, 2002).

Em um contexto de TV interativa, um modo contínuo onde geração dos enredos, dramatização e interação ocorrem continuamente tende a ser o modo principal de utilização do sistema. No entanto, um modo passo-a-passo, onde se pode examinar a geração dos enredos de forma análoga ao processo de depuração de programas, ainda mantém sua utilidade.

O novo modelo proposto deverá, portanto, manter a funcionalidade de geração de histórias em modo passo-a-passo, alternando passos de interação e criação das narrativas, o que não é comum em outros sistemas de *storytelling* interativo, e que por aproximar mais ainda o espectador do autor, abre possibilidades de uso que não devem ser desperdiçadas, caso fosse totalmente substituída pela geração em fluxo contínuo. É fundamental ressaltar, porém, que são estratégias completamente opostas, e que, portanto, embora o modelo deva suportar as duas, é razoável assumir que somente uma seja usada em uma determinada história de cada vez.

#### **5.1.5 Facilidade de Interação e Possibilidades de Interação em Diversos Níveis**

Em um ambiente de TV interativa, deve-se permitir que o usuário interaja com facilidade, ou seja, que não lhe seja exigido um grande esforço, dado que, diferentemente do que ocorre em jogos eletrônicos, o usuário de TV tende a estar com o foco da atenção voltado para a história e não para a sua agilidade na interação.

Por outro lado, é preciso buscar atender aos anseios de diferentes estilos de usuários. Usuários podem querer assumir uma postura básica de espectador, interferindo muito pouco no rumo da história ou querer assumir uma postura mais ativa, onde as suas intervenções acontecem continuamente e determinam o desencadeamento do enredo.

É necessário, portanto, que exista suporte para variadas maneiras de interação, com graus diferentes, no que concerne à intensidade da intervenção do usuário. A visão do Logtell de interações fracas e fortes é compatível com essa flexibilidade demandada pelo ambiente. O usuário deve em um modo de interação contínuo, do mesmo jeito que no modo passo-a-passo, ser capaz de fazer interferências na história que têm menos força, pedindo, por exemplo, ao gerador de enredos que forneça uma nova versão para uma parte da história recentemente apresentada. Alternativamente, o usuário deve ser capaz de inserir eventos ou situações específicas, desde que válidos e coerentes com o contexto. O desafio é que todas essas alternativas de interação sejam possíveis em um modo de interação contínuo, sem demandar atenção exagerada do usuário.

O ambiente de TV interativa é ainda um ambiente novo, que herda características da TV tradicional e de sistemas computacionais em rede. Para se testar a viabilidade do novo modelo de *storytelling* interativo é preciso fornecer um conjunto de mecanismos básicos de interação que dêem um mínimo de comodidade aos usuários para a interação. Novos mecanismos de interação direta e indireta com as histórias podem ser mais eficientes ou complementar bem o conjunto de mecanismos inicialmente definidos. Deseja-se, portanto, que o novo modelo seja aberto, na medida do possível, à incorporação de novos métodos, de modo que sua eficácia e aceitabilidade por parte dos usuários possam ser testadas.

### **5.1.6 Escalabilidade**

Em um ambiente de *storytelling* interativo para TV, espera-se que haja muitos usuários utilizando o mesmo sistema, compartilhando os mesmos contextos e até mesmo interagindo em conjunto sobre uma mesma história. É fundamental, portanto, prover um suporte robusto e escalável que atenda às demandas de processamento.

O processo de geração de histórias, por ser baseado em planejamento automático, tende a consumir uma boa quantidade de recursos de tempo e memória. Como o número de usuários do sistema pode variar de um usuário único a milhões de usuários, é necessário que o sistema seja implementado segundo uma arquitetura escalável, onde, pela simples incorporação de novos recursos computacionais, consiga-se garantir a responsividade do sistema.

Na versão atual do Logtell, e também nos demais sistemas de *storytelling* interativo estudados, as soluções são, em geral, não escaláveis e utilizam um modelo convencional de uma única máquina. Nesses casos, há uma limitação evidente ao atendimento de maiores demandas ou a diversos usuários simultâneos. O novo modelo, portanto, ao ser proposto para a TV interativa, precisa levar em conta a questão da escalabilidade.

### **5.2 Arquitetura Cliente-Servidor**

O novo modelo propõe uma arquitetura cliente-servidor. Tal arquitetura procura atender tanto a requisitos funcionais como a requisitos não funcionais, como a continuidade do fluxo, a confiabilidade, a escalabilidade, e a possibilidade de incorporação de novas alternativas de interação e de diferentes usos da estrutura do sistema de *storytelling*.

Em uma arquitetura cliente-servidor (TANENBAUM, 2003) trabalha-se com um cenário onde há um ou mais computadores controlando e disponibilizando o acesso a recursos computacionais, os servidores, e outros que requisitam esses recursos, os clientes. Nos sistemas que trabalham com esse tipo de arquitetura, o processamento fica dividido entre os servidores e os clientes. As formas de distribuição do processamento e de alocação dos recursos podem ser variadas e dependem dos objetivos que se deseja atingir. Uma das maiores vantagens, para o caso do *storytelling* interativo na TV, é que diferentes clientes podem ser implementados para acessar os mesmos servidores, permitindo a coexistência de clientes para celular, TV digital e outras opções. Mais detalhes sobre o servidor de aplicação utilizado no protótipo resultante dessa pesquisa são dados no capítulo 6.

A nova arquitetura proposta, usada para implementar uma nova versão do Logtell (2.0), é apresentada na Figura 5.1. Verifica-se que há vários módulos novos, com diferentes responsabilidades. O processamento no Logtell 2.0, de modo diferente da primeira versão, passa a distribuir o processamento entre um lado cliente e um lado servidor. O lado do cliente é responsável pela interação e dramatização das histórias. O lado do servidor é responsável pela simulação que gera os enredos das histórias. Os processos agora ocorrem em paralelo e devem ser coordenados entre si. Para cada história que está sendo apresentada, há uma aplicação em execução no servidor, enquanto que uma ou várias aplicações estão sendo utilizadas em diferentes clientes. Isso lida com o caso onde vários usuários estão assistindo à mesma história.

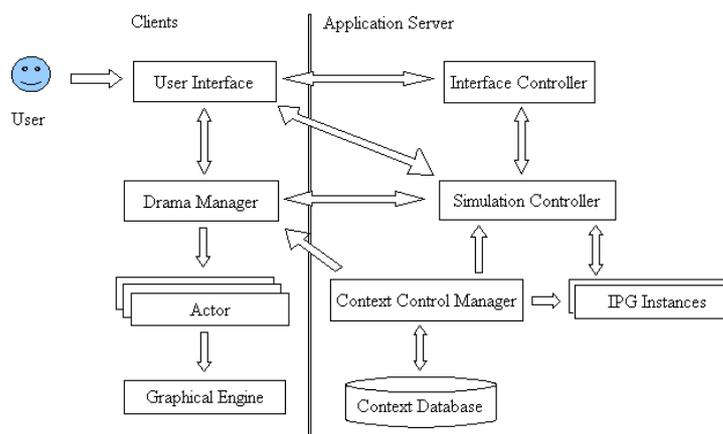


Figura 5.1: Nova Arquitetura do Logtell no Modelo Proposto

Pensando no caso da TV interativa digital, os set-top boxes certamente terão

recursos computacionais limitados, o que tornaria difícil realizar tarefas computacionalmente intensas, como planejamento automatizado. Ao concentrar a simulação em servidores de aplicações, é mais fácil alcançar uma melhor escalabilidade. Além disso, é mais fácil exercer controle quando uma única história é compartilhada por vários usuários.

Na versão anterior do Logtell (Figura 4.1), a interface principal do sistema concentrava o controle geral da história, através do *Plot Manager*, ativando uma única instância do IPG e acionando o *Drama Manager* conforme necessário. Além disso, mantinha-se sempre uma conexão com a instância do IPG enquanto a história era narrada. Na nova arquitetura, esta interface se decompôs em 3 módulos de controle: o *Simulation Controller*, o *Interface Controller* e a *User Interface*. O *Simulation Controller* e o *Interface Controller* são executados no servidor e a *User Interface* (interface gráfica do usuário) no cliente.

A *User Interface* corresponde, na nova arquitetura, à parte do *Plot Manager*, que, na primeira versão do Logtell, era responsável pela interação direta com o usuário. A *User Interface* implementa dois modos de interação diferentes: o modo a passo-a-passo, com as mesmas funcionalidades disponibilizadas na primeira versão do Logtell; e o modo contínuo, explicado em mais detalhes no item 5.3 adiante.

No lado do cliente, para utilizar as novas funcionalidades do modo contínuo, o usuário interage com o sistema através da *User Interface*, que informa as interações desejadas ao *Interface Controller* localizado no servidor. No modo passo-a-passo, a interface aciona diretamente o *Simulation Controller*. O *Drama Manager* requisita os eventos a serem dramatizados ao *Simulation Controller* e controla os atores 3D que representam cada personagem na dramatização gráfica 3D das histórias.

No lado do servidor, o *Interface Controller* recebe as sugestões às histórias fornecidas por vários clientes. Quando múltiplos usuários compartilham a mesma história, as interações são selecionadas de acordo com o número de clientes que as pediram. Em trabalhos futuros também poderão ser usados outros critérios para esta escolha, escolhendo não só as opções mais votadas, mas também fazendo um balanço com o quanto a sugestão escolhida ajuda a história a continuar, por exemplo. As sugestões são então enviadas ao *Simulation Controller*, que é responsável por:

- informar ao *Drama Manager*, no lado do cliente, os próximos eventos a serem dramatizados, quando requisitado;

- receber pedidos de interação e incorporá-los à história, utilizando instâncias do IPG;
- contabilizar sugestões quando no modo multi-usuário, e então usá-las, se possível;
- selecionar interações viáveis e possivelmente interessantes (de acordo com o seu potencial de gerar eventos na história), para serem disponibilizadas aos usuários;
- controlar um número de instâncias do *Interactive Plot Generator* para obter os próximos eventos da história; e
- controlar o tempo gasto na simulação e na dramatização.

As instâncias do IPG realizam simulações através da inferência de objetivos e usando planejamento automatizado para a realização desses objetivos. Para conseguir fazer isso, o contexto das histórias é consultado.

Para manter a responsividade, o fluxo de apresentação precisa ser contínuo. Desse modo, a simulação precisa estar sempre pelo menos um capítulo à frente da dramatização. Quando não há intervenção do usuário, os objetivos são inferidos, e os eventos são planejados de forma contínua. Quando os usuários intervêm na história, eles interagem de acordo com os eventos que estão sendo dramatizados. O *Simulation Controller* guarda “retratos” dos estados de cada simulação após cada ciclo, de forma que a simulação possa ser retomada a partir de um ponto anterior, possivelmente com outras alternativas de desencadeamento da trama, se isso for solicitado pelo usuário. A coerência lógica de uma intervenção requisitada é sempre checada antes de ser incorporada, e descartada se inconsistente. Quando uma intervenção é incorporada, a simulação tem que descartar ciclos de simulação que tenham sido previamente planejados e que não tenham levado a intervenção em conta. Para estar preparado para as intervenções, o sistema cria outras instâncias do IPG, simulando a incorporação dessas intervenções a serem sugeridas como opções aos usuários. As intervenções só são sugeridas se a instância do IPG confirmar sua consistência. Se aceitas, os próximos eventos já terão sido planejados e haverá pouco risco de interrupção no fluxo da narração.

O tempo gasto para a simulação é constantemente monitorado pelo *Simulation Controller*. Quando há risco de interrupção na dramatização devido à falta de eventos já

planejados, uma mensagem pode ser enviada ao *Drama Manager*, de forma que políticas para estender a dramatização dos eventos correspondentes possam ser aplicadas. Se, em uma dada circunstância, um capítulo for requisitado e ainda não tiver sido escrito, o *Drama Manager*, no cliente, pode dar um tratamento adequado para tentar evitar a interrupção do fluxo. Como é possível que, eventualmente, mais de um tipo de cliente seja implementado (por exemplo um para TV digital, outro para celular e outro para acesso via Internet), cada cliente poderá tratar atrasos de diferentes formas.

No modelo proposto, o *Drama Manager*, em princípio, é executado no cliente. Isso faz com que o tráfego de dados corresponda essencialmente à carga do contexto e às comunicações entre o cliente e o servidor, evitando-se a transmissão de vídeo. É razoável, porém, admitir que, em um contexto de TV digital, os *set-top boxes* não tenham capacidade computacional para cuidar da dramatização. Nesse caso, como arquitetura alternativa, o *Drama Manager* poderia ser deslocado para o servidor, transmitindo-se o vídeo resultante da dramatização para os clientes. Tal solução parece ser viável para o caso de compartilhamento das histórias por um número muito grande de usuários, no qual o vídeo poderia ser transmitido em *broadcast*. No caso de histórias individualizadas ou compartilhadas por pequenos grupos, os servidores teriam que produzir grande quantidade de vídeos diferentes em paralelo e transmiti-los para os respectivos clientes. Essa transmissão de vídeo *on demand* na TV digital, por exemplo, pode se tornar inviável, devido às limitações da largura de banda.

### **5.3 Interação Contínua**

Com o objetivo de dar suporte a *storytelling* interativo em ambientes que demandam alta responsividade, como na TV digital, o novo modelo do Logtell provê um modo contínuo de interação, no qual a história é gerada em tempo real conforme o usuário a assiste e interage com ela. Nesse modo de interação, os passos de simulação (combinados a intervenções do usuário) passam a ser capítulos gerados e apresentados continuamente.

Para funcionar bem nesse ambiente, sem atrapalhar a experiência de se assistir à dramatização, a interação não pode exigir muita atenção do usuário e os métodos de interação no modo contínuo precisam levar isso em conta. A interação no modo passo-a-passo foi mantida para se possibilitar também a criação e análise dos enredos de forma cuidadosa. No entanto, para fins de entretenimento, o uso do modo passo-a-passo tende a ser a exceção.

O *Simulation Controller* tem um papel importante na implementação de novas formas de interação no modo contínuo. Uma das suas tarefas mais importantes é criar os meios para a interação enquanto a geração dos enredos é feita e a sua dramatização ocorre no lado do cliente. Também no modo contínuo, as interações podem ser tanto fracas quanto fortes.

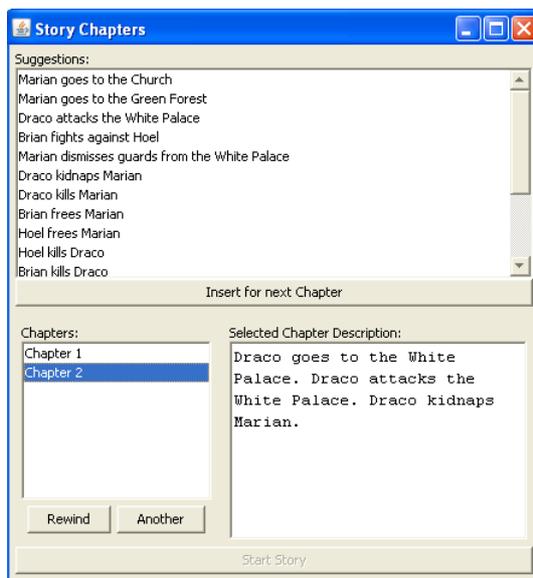


Figura 5.2: Interação Contínua

A Figura 5.2 demonstra a interface de interação no modo contínuo. Eventos e situações, descritos em linguagem natural, são sugeridos através de uma lista continuamente atualizada e sensível ao contexto que está sendo dramatizado. Sugestões podem ser escolhidas pelo usuário para que o Logtell tente inseri-las no próximo capítulo. Os capítulos também são listados e há uma janela onde os eventos do capítulo selecionado aparecem de forma resumida. Os comandos *Rewind* e *Another* permitem o retorno do processo de geração (e de dramatização) ao capítulo selecionado, diferindo apenas na forma como o processo é retomado, conforme descrito a seguir.

### 5.3.1 Intervenções Fracas: *Rewind* e *Another*

Interações Fracas no modo contínuo basicamente circundam o fluxo “normal” da história, como em geral se obtém com o uso dos comandos *Continue* e *Another* do modo passo-a-passo. O *Simulation Controller* direciona o fluxo da história escolhendo, automaticamente e de forma aleatória, alternativas de capítulos e a própria ordenação total dos eventos a serem dramatizados em cada capítulo (sendo essa última compatível

com a ordem parcial definida pelo IPG). Dessa forma, o usuário pode assistir à história sem ter que interferir. Além disso, em execuções diferentes, mesmo que mantida a configuração inicial, podem-se obter histórias diferentes.

No modo passo-a-passo, ao fim de uma fase de simulação, o usuário pode examinar os eventos ainda não incorporados. Pode também solicitar a dramatização repetidas vezes a partir de um certo ponto. No modo contínuo, como a história é apresentada sem interrupções, é preciso disponibilizar mecanismos para que o usuário possa voltar a pontos anteriores da história, seja para revê-los (comando *Rewind*), seja para solicitar a geração de alternativas diferentes, como seria feito com o comando *Another* no modo passo-a-passo (que gera versão diferente do capítulo). Para esse objetivo, diferentes retratos do processo de simulação ao fim de cada capítulo são guardados pelo *Simulation Controller*. De forma diferente do que ocorria na primeira versão do Logtell, pode-se agora retornar a um capítulo anterior para modificá-lo, mesmo que ele já tenha sido dramatizado. Dá-se ao usuário então a possibilidade de “mudar o passado da história” e verificar os resultados dessa mudança.

### 5.3.2 Intervenções Fortes

As Intervenções Fortes, como no modo passo-a-passo, correspondem tanto à inserção de eventos específicos no enredo quanto a diretivas de que certas situações devem ocorrer. Como o usuário está assistindo à história em paralelo, a descrição dos eventos ou situações através da digitação de texto ou com o auxílio de menus, como no modo passo-a-passo, demanda mais atenção do que o tolerável para quem precisa prestar atenção à história.

Para permitir intervenções fortes no modo contínuo, o *Simulation Controller* gera automaticamente sugestões de eventos e situações a serem inseridas na história se o usuário assim desejar. Essas sugestões devem, na medida do possível, ser coerentes e levar a desencadeamentos diferentes do enredo. Para garantir a coerência, o IPG pode ser acionado previamente, sugerindo-se então apenas o que já se sabe que pode ser incorporado ao enredo. Para obter sugestões coerentes com o capítulo que está sendo visualizado, alguns métodos foram pensados.

O primeiro método para obter uma sugestão seria usando uma biblioteca de planos típicos, organizados como uma hierarquia de eventos básicos e complexos. Planos típicos geralmente consistem de certas combinações de eventos onde os vários personagens perseguem seus objetivos, mas também podem corresponder aos motivos

(AARNE 1964) que são estruturas recorrentes compiladas através de estudos críticos sobre o gênero. O IPG contém um algoritmo para o reconhecimento de planos, baseado em um algoritmo especificado por Kautz (KAUTZ 1991). O procedimento é capaz de descobrir que alguns eventos são compatíveis com um motivo para o qual se tem um plano típico, deixando então que o *Simulation Controller* sugira a inclusão de eventos adicionais contidos nesse plano.

O segundo método seria através da análise das regras de inferência de objetivos em conjunto com a situação atual da narrativa. Através da análise, é possível detectar fatos a serem sugeridos que, uma vez que se configurem, podem levar à ativação de regras que provocam novos desencadeamentos para a história nos próximos capítulos.

A terceira maneira imaginada para a geração de sugestões para a história, seria através das regras de inferência de sugestões, que, de forma semelhante às regras de inferências de objetivos, fazem uma análise do estado do mundo atual e fornecem sugestões de eventos ou sugestões de situações a serem inseridas no próximo capítulo. Este método foi o único a ter sido implementado no protótipo, apresentado adiante.

#### **5.4 Interação com Múltiplos Usuários**

Tendo em mente o aspecto massivo da televisão interativa, é proveitoso analisar modos de permitir que mais de um usuário possa ter influência na evolução de uma história compartilhada. Em histórias compartilhadas, comandos como *Rewind* e *Another* tendem a não fazer muito sentido. A interação fica concentrada, portanto, nas intervenções fortes.

Para coordenar intervenções fortes de vários usuários, mecanismos de votação precisam ser implementados, de modo que as mais votadas sejam inseridas. É preciso também coordenar a apresentação simultânea da história nos diversos clientes, a atualização das listas de sugestão e o tempo para a recepção das escolhas dos usuários.

No novo modelo, o *Interface Controller* organiza a interação entre usuários e interage com o *Simulation Controller* como se houvesse apenas um usuário. Como regra, todas as sugestões devem ser enviadas pelo *Interface Controller* ao *Simulation Controller* quando alcançam um nível mínimo de aceitação configurado.

Apesar de o novo modelo prever inicialmente apenas os mecanismos de votação, outros mecanismos de interação com múltiplos usuários que compartilham a mesma história compensam ser analisados. Usuários poderiam, por exemplo, ser divididos em grupos com possibilidades de intervenção diferentes. Alternativamente, poderiam

também assumir o controle de personagens, mas isso implicaria em gerar enredos mais flexíveis, dando maior liberdade para a dramatização. De todo modo, o novo modelo cria uma plataforma útil para experimentos de interação entre múltiplos usuários.

### **5.5 Estratégias para Manutenção do Fluxo**

Para que o novo modelo seja bem sucedido, uma das principais questões é que, no modo contínuo, se evite a interrupção do fluxo de apresentação mesmo quando intervenções fortes ocorrem.

Para evitar a interrupção, a geração dos enredos precisa estar sempre à frente da dramatização. Com recursos computacionais de sobra, vários capítulos à frente poderiam ser gerados no servidor enquanto o capítulo corrente é apresentado. Se alternativas que prevêm intervenções fortes forem geradas em paralelo, diminui-se consideravelmente o risco de interrupção. No entanto, como os recursos dos servidores não são infinitos, um certo equilíbrio precisa ser achado. Uma alternativa básica é ter sempre um capítulo à frente gerado e tentar incorporar sugestões apenas se der tempo. Caso não haja, o capítulo sem incorporar as sugestões é usado.

A eficiência do processo de simulação também pode ser melhorada. Uma das idéias para alcançar performance maior no planejamento é o uso de *hierarchical task networks* (HTNs) (EROL *et al.* 1994; NAU *et al.* 2003). Esses algoritmos tendem a ser mais eficientes pois reduzem o espaço de busca, mas ao preço de restringir as soluções a padrões previamente estabelecidos. No entanto, deseja-se manter a flexibilidade na geração dos enredos. O uso de um planejador híbrido que dê preferência a soluções compatíveis com uma hierarquia de tarefas, mas que também possibilite a busca de soluções fora de padrões previamente pensados, pode levar a melhoria considerável na performance.

Adicionalmente, algumas pesquisas recentes com planejadores neoclássicos, que fazem buscas no espaço de estados, mostram que bons resultados em termos de eficiência também podem ser obtidos, como, por exemplo, no planejamento baseado em grafos (BLUM AND FURST 1997; GEVERINI & SERINA 2002). O uso de heurísticas e estratégias de controle (DOHERTY 2001; HOFFMAN 2001) também são promissores. A incorporação dessas técnicas também merece ser levada em consideração.

Por fim, é interessante a incorporação ao novo modelo de estratégias para controle do tempo de dramatização dos eventos. Em (DORIA *et al.*, 2008) é apresentada

uma proposta de flexibilização da dramatização no Logtell, dando variedade tanto às formas de dramatizar cada evento quanto ao seu tempo de duração. Através de comandos para acelerar ou encurtar o evento corrente, é possível atrasar a dramatização de um capítulo até que se tenha o próximo capítulo disponível.

## **5.6 Mecanismos Avançados de Interação**

Além dos modos de interação básicos previamente citados, outros mecanismos poderão vir a ser incorporados:

- Uso de conceitos abstratos – A biblioteca de planos típicos com que o IPG trabalha admite dois tipos de relacionamentos entre eventos. Eventos podem ser parte de outros eventos ou especializações de eventos mais abstratos. O usuário poderia, por exemplo, indicar que a princesa, que faz o papel de vítima no contexto corrente, deve sofrer uma vilania, mas deixar a escolha do tipo de vilania a cargo do processo de simulação. De forma análoga, se uma hierarquia de predicados é definida, o usuário pode estabelecer que a princesa deva estar em perigo em certo momento mas sem especificar como. Um nível adequado de abstração encorajaria o usuário a intervir na história sem ter que se preocupar com detalhes.
- Controle das tensões narrativas – Escalas numéricas referindo-se a níveis de violência, romance, assim como à velocidade da narrativa poderiam ser manipulados pelo usuário enquanto assiste à dramatização. Essas configurações ajudariam a guiar o processo de simulação e a inferência de novos objetivos durante o processo de simulação. Isso requer, no entanto, um investimento maior na hora de se modelar o gênero (POLTI 1945).
- Sugestões em Linguagem Natural – É interessante que o usuário possa comunicar-se com o sistema através de frases num subconjunto restrito da linguagem natural. Depois de decodificadas, essas ordens seriam tratadas como as outras formas de interação já existentes, ou seja, seriam validadas de modo a verificar logicamente sua consistência. Num modelo multiusuário, sugestões expressas em linguagem natural podem ser mais facilmente entendidas e discutidas entre os usuários.

## 5.7 Conclusões

Neste capítulo, foi apresentado um novo modelo para *storytelling* interativo em ambiente de TV interativa. O modelo procura conciliar o foco na coerência com novos mecanismos capazes de prover a responsividade demandada por esse ambiente.

O modelo apresentado levou em consideração uma série de requisitos que incluem, além da diversidade e coerência dos enredos, o fluxo contínuo na apresentação das histórias, a facilidade e comodidade na interação, intervenções dos usuários nas histórias em diversos níveis e o compartilhamento de histórias entre múltiplos usuários.

Optou-se por utilizar e modificar o modelo do Logtell, por esse já ter foco na coerência dos enredos e já possibilitar tanto intervenções fracas quanto fortes por parte dos usuários. Para permitir que enredos continuassem podendo ser gerados e examinados com cuidado, decidiu-se manter o modo de interação passo-a-passo. O foco, no entanto, passou a ser um novo modo contínuo de interação, onde a geração de enredos ocorre em paralelo com a dramatização. Para o modo contínuo, procurou-se definir mecanismos de interação que dessem o mesmo poder de intervenção do modo passo-a-passo, mas provendo a comodidade necessária.

No novo modelo, uma nova arquitetura cliente-servidor é proposta, a qual contrasta com boa parte das arquiteturas adotadas nas pesquisas de *storytelling* interativo, onde o processamento tende a ficar concentrado em uma única máquina. A nova arquitetura apresenta uma série de vantagens funcionais e não funcionais, tais como uma maior escalabilidade e confiabilidade na disponibilização dos recursos computacionais necessários ao processo de simulação das histórias. Além disso, permite a coexistência de diferentes clientes em diferentes plataformas onde a TV interativa pode ser veiculada, tais como a TV digital e a telefonia móvel.

No capítulo 6 a seguir, é descrita a implementação de um protótipo que busca validar parte das idéias do modelo apresentado.

## 6. Implementação

Neste capítulo, são abordadas as principais questões relativas à implementação do protótipo para a validação do modelo de *storytelling* interativo apresentado no capítulo 5.

O protótipo foi feito sobre a implementação atual do Logtell e buscou atender aos requisitos básicos do novo modelo proposto, de forma a validar suas principais idéias.

### 6.1 Ambiente da Aplicação

Para a implementação do protótipo, um dos principais requisitos é a necessidade de um ambiente cliente-servidor que provenha, dentre outras facilidades, um meio onde existam confiabilidade e escalabilidade. Por isso, a arquitetura implementada faz uso de um servidor de aplicação.

Na primeira versão do Logtell, o *Plot Manager*, implementado em Java, faz o controle dos demais módulos, como o IPG e o *Drama Manager*. O IPG, por sua vez, embora seja implementado em Prolog, é acessado através de uma biblioteca Java que faz a interface com o processador de Prolog SICStus. Essa biblioteca utiliza uma DLL interna do SICStus, não sendo implementada em Java puro. O mesmo ocorre com o *Drama Manager*, que, embora se comunique com o Plot Manager via Java, é implementado em C++ através de uma DLL e de chamadas nativas.

Pelo fato do controle principal da execução do Logtell ser implementado em Java e para se ter uma maior independência de plataforma, foi escolhido um servidor de aplicações Java, o JBoss (MARRS e DAVIS, 2005). O JBoss foi construído usando uma série de ferramentas e bibliotecas de código aberto que já alcançaram um considerável grau de maturidade e que fornecem a um ambiente corporativo o núcleo necessário para sua infra-estrutura. Dentre outras tecnologias, o JBoss dá suporte, por exemplo, a JSPs, *Servlets*, EJBs, JMS, JNDI, *Web Services*, JavaMail, JDBC e Hibernate. Ao disponibilizar serviços distribuídos e facilidades de segurança, arquitetura para mensagens assíncronas, *proxies* remotos, acesso a bancos de dados, criação de *Web Services* e até servidores HTTP para Web, torna-se facilitada a construção de aplicações distribuídas entre vários servidores, as quais podem atender a uma grande quantidade de clientes simultâneos. Tal característica é bastante atraente para implementação do modelo de TV interativa proposto nesta dissertação.

Para a implementação, foi escolhida a versão 4.2.3.GA do JBoss. O novo protótipo do Logtell tem como plataforma Java alvo a versão 6 do JDK da Sun, fazendo uso de novos recursos da linguagem, como *Generics*, e usufruindo também das melhorias de performance das últimas versões da máquina virtual Java.

Uma das grandes vantagens de usar um servidor de aplicações como o JBoss é que, com a sua arquitetura, pode-se disponibilizar serviços de maneira mais independente da plataforma tecnológica, através do uso de *Web Services*. Para isso, basta que sejam feitas pequenas alterações nos serviços que foram codificados. No protótipo construído, essas alterações não foram feitas, porém sua implementação é totalmente compatível, dado que os serviços são disponibilizados através de *Stateless Session Beans*, ou seja, não é guardado um contexto de comunicação com cada cliente que faz chamadas remotas. Além disso, os serviços foram construídos como EJBs (*Enterprise Java Beans*) e seguindo o padrão EJB3. Com isso, só é necessária a adição de algumas anotações (*annotations*) no código para que um serviço do sistema se torne um Web Service. A possibilidade do uso de *Web Services*, que é um padrão de integração para sistemas heterogêneos, facilitará a extensão futura do sistema de *storytelling* interativo, permitindo o seu uso através de diversos tipos de clientes diferentes, como PCs, Televisão Digital, celulares, etc.

Na implementação foi utilizada uma série de padrões de projeto, os quais são bastante estudados na bibliografia e usados no ambiente de programação J2EE, ou seja, ambientes *Enterprise* de aplicações distribuídas feitas em Java. O acesso à base de dados é feito por classes que implementam o padrão *Data Access Object* (DAO) (ALUR *et al.*, 2001), que encapsula todo o acesso ao banco de dados e permite modificar a maneira de persistir as classes do modelo sem que sejam necessárias modificações no código do cliente. É importante notar que, por questões de tempo, o protótipo não usa de fato um banco de dados (os dados são guardados em memória), porém a sua arquitetura já está preparada para isso, dado que o acesso a banco de dados foi isolado, e poucas questões referentes ao banco de dados específico a ser escolhido devem surgir na hora de sua incorporação à nova arquitetura.

É importante também ressaltar que, na implementação da lógica dos serviços disponibilizados, foi adotada, como orientação geral, a utilização da arquitetura de POJOs (*Plain Old Java Objects*) descrita por (JOHNSON 2004). Os POJOs nada mais são do que classes Java que buscam uma simplicidade no seu código, sem seguir padrões arquiteturais como o EJB. Neste caso, como recomendado por (JOHNSON

2004), eles são utilizados para conter a lógica necessária para controlar o processo de escrever as histórias. Dessa forma, embora se utilizem EJBs, isto se deve principalmente por suas questões de arquitetura dos servidores de aplicação, que provêm o acesso a estes serviços e as questões funcionais e não funcionais que são fornecidas pela plataforma J2EE, porém estando a lógica principal dos serviços escrita nos POJOs, que são utilizados dentro da implementação dos EJBs. Desta forma, esta lógica pode ser facilmente reutilizada em diversos pontos do sistema, não tendo que ser sempre acessadas através do EJB, reduzindo o acoplamento do sistema à estrutura usada para estes serviços disponibilizados, ou seja, facilitando além do reuso potencial da lógica do sistema, futuras mudanças caso sejam necessárias novas modificações na arquitetura do sistema.

Para fazer a integração com o IPG na nova versão, alguns desafios foram encontrados. A estrutura da biblioteca Jasper, utilizada pelo Logtell para falar com o SICstus Prolog, onde o IPG é executado, criou dificuldades de alocação de memória e limitações no acesso simultâneo ao Prolog.

Um dos principais problemas é o fato de o Jasper utilizar um espaço de memória baixo (os primeiros 256 de MB do sistema), e falhar na sua inicialização caso outras aplicações ocupem este espaço de memória antes dele. Infelizmente, ao implementar a nova arquitetura, com o processo do Jasper dentro do ambiente JBoss, aconteceu exatamente a situação descrita e não era possível instanciar a conexão com o Prolog. O problema pôde ser remediado através de orientações descritas no site do fabricante do SICstus (SICSTUS, 2008), que recomendam o uso da ferramenta Rebase da Microsoft, em conjunto com a customização de um arquivo DLL da JVM a ser usada com o JBoss. Essa alteração não é tão intrusiva quanto parece, dado que ela apenas cria uma nova configuração de memória para a máquina virtual Java (semelhante às configurações para cliente ou servidor usadas no Java), que pode então ser usada.

Na implementação do novo protótipo, pretendia-se utilizar uma série de instâncias do IPG para atender a diversas demandas paralelas, da simulação de uma mesma história ou de histórias distintas. Entretanto, utilizando a biblioteca Jasper, não era possível instanciar a conexão com o Prolog mais de uma vez numa mesma máquina e, além disso, o Jasper exige o acesso ao Prolog através de uma única *thread*.

Na arquitetura anterior do Logtell, era sempre utilizada uma única instância do processo de conexão com o Prolog, que era mantido sempre aberto, de modo similar ao uso de um console Prolog que recebe comandos de um usuário. Essa instância do Prolog

recebia as requisições do *Plot Manager* com consultas Prolog e as atendia. A exigência de que o Prolog não seja acessado através de mais de uma *thread* impedia o envio direto de comandos da interface para o objeto Java que representa a instância. As chamadas resultantes do clique de um botão, por exemplo, viriam de uma *thread* separada, usada pelo Java para atender aos eventos que vêm de componentes visuais de interface. Para contornar essa limitação, era utilizado um mecanismo onde uma série de *flags* eram constantemente alterados e verificados por uma *thread*, que, sendo executada em paralelo com a interface, repassava os comandos necessários e resultados das consultas Prolog.

No novo ambiente cliente-servidor, não seria mais possível manter aberta apenas uma instância do Jasper, dado que o suporte a múltiplas requisições é exatamente uma das principais características do modelo proposto. Como as requisições que chegam ao servidor de aplicação podem vir de diversos clientes, o IPG pode ter que ser acionado para tratar diferentes contextos de histórias a cada momento. Surge aí a mesma limitação do modelo de *thread* única enfrentada já na implementação da primeira versão do Logtell: como atender as requisições vindas de diferentes *threads* e chamadas remotas?

Para atacar esse problema, a solução foi utilizar uma estratégia análoga à que foi utilizada na primeira versão do Logtell. Foi utilizado o padrão MBean, disponível pela arquitetura J2EE e suportado pelo JBoss. Com um MBean, é possível que se tenha uma instância de um objeto que é única por cada servidor de aplicação dentro do ambiente implantado. Dessa forma, foi criado o MBean chamado *PlannerMBean*, que, de maneira semelhante à arquitetura anterior, fornece o acesso ao IPG através de uma *thread* que centraliza o acesso à instância Prolog. Além disso, como só há uma instância Prolog por servidor, a cada chamada remota onde é necessário o uso do IPG, deve-se primeiro restaurar o *snapshot* que armazena o contexto da simulação referente à chamada. Os *snapshots* do contexto de cada história ao fim de cada fase de simulação são guardados e recuperados quando necessário.

Embora, por um lado, possa parecer uma limitação grande o fato de apenas uma instância de acesso ao IPG poder existir por servidor de aplicação, deve-se considerar que, na nova arquitetura criada, diversos servidores de aplicação podem ser ligados em um *cluster* com o JBoss. Dessa forma, diversas instâncias do IPG podem ser usadas em paralelo para atender às necessidades dos clientes conectados ao sistema. No protótipo, isso ainda não foi feito, e, antes que isso aconteça, deve-se primeiro integrar um banco

de dados verdadeiro, substituindo a implementação de persistência dos *snapshots* através dos DAOs. Isso é necessário porque não será suficiente manter os dados somente em memória, pois o contexto de simulação de uma história gerado em um servidor de aplicação pode ter que vir a ser estendido em outro servidor.

É importante ressaltar que algumas das limitações apresentadas podem eventualmente ser eliminadas utilizando novos métodos de acesso disponibilizados na versão mais atual do SICstus. No entanto, mudanças no código de acesso e no IPG seriam necessárias. Como há planos de migração do IPG para o SWI Prolog, o esforço necessário para essas mudanças não se justificava. No novo ambiente, os métodos de acesso deverão ser adaptados tendo em vista o aumento da performance. Como forma de dar suporte a melhorias futuras no acesso a outras implementações do IPG, o *PlannerMBean* acessa o IPG através da classe *PlannerFacade* que foi criada no novo protótipo, abstraindo de uma maneira mais modular, o acesso ao Prolog. Assim, a integração com o Prolog através do SICstus poderá ser facilmente substituída no futuro, sem fazer muitas mudanças estruturais na arquitetura implementada.

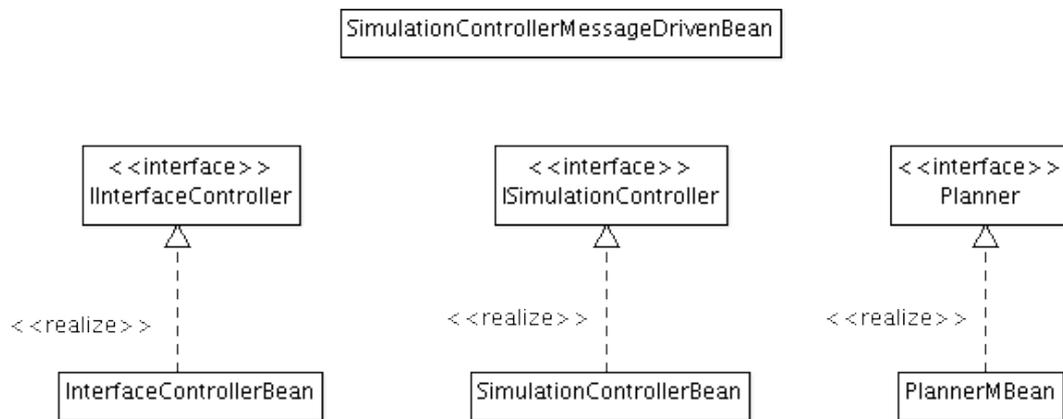


Figura 6.1: Beans J2EE do protótipo

A Figura 6.1 apresenta os módulos implementados no servidor sob a forma de *beans*. As seções a seguir detalham a forma como esses *beans* trabalham em conjunto com o código executado nos clientes.

## 6.2 Simulation Controller

O *Simulation Controller* tem a responsabilidade principal de controlar a geração

das histórias. Para isso, foi criado como um *Stateless Session Bean*, chamado *SimulationControllerBean*, cujos métodos principais são apresentados na Figura 6.2.

logtell::ejb::SimulationControllerBean
<pre> getAllStoriesRunning() : List getAllContexts() : String[] requestStoryCreation(continuous : boolean, multiUser : boolean, context : String, timeToStart : long) : Long getStoryChapter(storyId : Long, chapterNumber : int) : Chapter updateStory(story : Story) : void startStory(story : Story) : void continueStory(storyId : Long, str_ev : String, str_ord : String, nextOpId : int) : PlanResult continueStory(storyId : Long, str_ev : String, str_ord : String, nextOpId : int, createNewSnap : boolean) : PlanResult doAnotherForStory(storyId : Long) : PlanResult ask(storyId : Long, ask : String, str_ord : String) : String getAllScheduledStories() : List getTimeToStart(storyId : Long) : long askMultiple(storyId : Long, ask : String, str_ord : String, anotherCount : int) : String askSuggestions(storyId : Long, string : String) : String </pre>

Figura 6.2: Métodos principais implementados no *SimulationControllerBean*

O *Simulation Controller* é capaz de fornecer aos clientes uma série de funções, a saber: o acesso aos contextos de histórias, a criação de novas histórias; a continuidade de uma simulação com novas fases de simulação; o fornecimento de novas alternativas para uma fase de simulação (através do comando *Another*); consultas sobre as histórias em execução; consultas sobre as histórias multiusuário agendadas; e o fornecimento de listas de sugestões de inserções de eventos e situações para interação forte em modo contínuo.

Como se pode observar, o *bean* criado busca atender às demandas de geração e simulação das narrativas, dando suporte tanto ao modo de geração de história passo-a-passo, que já existia, quanto ao novo modo contínuo – multiusuário ou não. Entretanto, este *bean* remoto, em vez de concentrar toda a lógica em um único bloco monolítico de código, é implementado, como já foi descrito na seção 6.1, através de POJOs, facilitando o reuso e modularização da lógica dos serviços criados.

A lógica de controle da criação de histórias contida no *SimulationControllerBean* se divide em alguns submódulos: serviços gerais para manipular a história ficam a cargo do *StoryManagerService* enquanto que a geração do enredo da história é controlada por subclasses da classe *AbstractStoryWriter*.

Através do uso de chamadas remotas ao *SimulationControllerBean*, pode-se requisitar a criação e geração iterativa das histórias. O modo de geração passo-a-passo funciona dessa maneira. Sempre que um novo ciclo de simulação é requisitado, o

serviço restaura o *snapshot* anterior e então requisita um novo ciclo de simulação.

A principal diferença entre o modo passo-a-passo e o modo contínuo, no que se refere à geração das histórias, ocorre principalmente nas subclasses da classe abstrata *AbstractStoryWriter*. O código que, na versão anterior do Logtell, manipulava o IPG, e que antes fazia parte do *PlotManager* foi completamente refatorado e distribuído.

As principais funções que eram utilizadas para solicitar ao Prolog consultas e que resultavam em novos eventos e suas dependências, agora se concentram na classe *PlannerFacade*, que é invocada pelo *PlannerMBean*, já descrito antes. O código que manipula essas chamadas no modo contínuo, porém, agora deve sempre ser capaz de organizar os ciclos de geração das narrativas em capítulos, onde a ordem total dos eventos é estabelecida automaticamente. Para suportar esse processo de geração da narrativa, existem duas subclasses: o *ContinuousStoryWriter* e o *StepByStepStoryWriter*.

O *StoryManagerService* é um grande POJO que concentra a lógica para a manipulação das histórias, dos *snapshots* da mesma e de seus capítulos. Essa classe manipula os DAOs necessários para acessar e armazenar os dados desejados, encapsulando dessa forma esse código e ajudando a abstrair essas questões de forma modular.

O *StepByStepStoryWriter* é encarregado basicamente de acionar o IPG, funcionando de forma similar à versão anterior do Logtell. Além do acionamento do IPG, contém método para ordenar os eventos usando uma ordenação topológica. Essa ordenação é feita no modo passo-a-passo apenas para facilitar a interação do usuário, não acrescentando restrições de ordem entre os eventos (ou seja, o usuário no modo passo-a-passo ainda deve determinar a ordem total dos eventos, fazendo a ligação entre eles na interface).

No modo contínuo, utiliza-se estratégia completamente diferente. Há uma instância do *ContinuousStoryWriter* para cada história apresentada. Existe um processo, denominado *SimulationControllerMessageDrivenBean*, que é executado continuamente no servidor, o qual é responsável por iniciar instâncias do *ContinuousStoryWriter* para cada história contínua com que se está trabalhando. O *ContinuousStoryWriter*, atua como um “roteirista ao vivo”. Para isso, manipula um *StepByStepStoryWriter*, mantendo seu próprio controle sobre a história que vai sendo gerada. O *ContinuousStoryWriter* implementa a estratégia do modelo: deve sempre estar à frente do que conteúdo a que o usuário assiste. Por isso, ao iniciar a história, o primeiro capítulo da história é logo escrito, mesmo antes de o usuário pedir o início da sua dramatização, aproveitando-se o

tempo de inicialização do ambiente gráfico. Quando, no modo contínuo, um capítulo é requisitado pelo cliente, é verificado, no lado servidor, se esse capítulo da história é o último disponível. Nesse caso, para evitar futura interrupção no fluxo, o *SimulationControllerBean* envia uma mensagem assíncrona (JMS) para o *ContinuousStoryWriter* responsável pela história, de modo que novo capítulo seja gerado.

O gerador de narrativas contínuas é também responsável por repassar as novas formas de interação com base em sugestões ao IPG. Sugestões inseridas pelo usuário só são incorporadas se o IPG verificar que elas são coerentes. Além disso, a incorporação das sugestões só ocorre se o capítulo a ser modificado já não está sendo assistido por algum cliente. Esse controle é feito tanto no modo contínuo em modo de único usuário quanto com as histórias compartilhadas.

O processo de geração de histórias multiusuário é muito semelhante ao das histórias contínuas com um único usuário. As principais diferenças estão no seu processo de início e nas interações. No modo multiusuário, quando um usuário faz uma interação forte (isto é, quando insere uma sugestão), esta na verdade é apenas um voto. A sugestão somente será escolhida para a história, se for a mais votada (em caso de empate, uma das mais votadas é escolhida randomicamente). No protótipo, apenas uma sugestão pode ser dada por capítulo, por interação.

Há uma questão que, no futuro, poderá ser melhorada e que é relativa à contagem de votos e ao critério para a utilização das sugestões dos usuários. Na implementação atual do modo multiusuário, ao ser requisitado o último capítulo escrito de uma história, o seguinte é escrito pelo gerador, da mesma forma que no modo contínuo de único usuário. Porém, no que se refere ao uso das sugestões inseridas no modo multiusuário, há um *threshold* configurado no código que define o quanto o processo aguarda até que faça a contagem das sugestões inseridas pelos usuários enquanto que no modo contínuo de único usuário, as sugestões serão incorporadas (se possível) em qualquer momento que o usuário desejar (salvo a exceção do caso aonde alguém já requisitou o capítulo futuro sendo modificado, aonde neste caso se descarta o capítulo modificado).

Outra diferença que também há no modo contínuo compartilhado, é que, nele, as histórias são agendadas pelos usuários, e então outros usuários podem selecionar as mesmas e aguardar o seu início. Além disso, os novos modos de interação fraca, através dos comandos *Rewind* e *Another* do modo contínuo não estão habilitados.



Figura 6.3: Submódulos do *Simulation Controller*.

### 6.3 User Interface

Na implementação do protótipo, o código da interface com o usuário teve que ser reescrito. O que antes era parte do *Plot Manager*, agora se divide em diferentes classes.

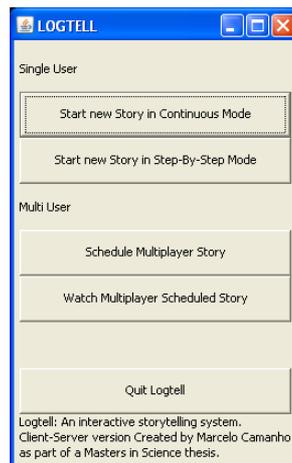


Figura 6.4: Interface Inicial do Logtell

Na nova interface do Logtell, agora existem múltiplas opções de uso, dada a natureza ambivalente do modelo e do protótipo. Agora não só é possível ao usuário o uso do modo passo-a-passo, mas também o modo contínuo, seja ele na modalidade de usuário único ou multiusuário. A Figura 6.4: apresenta a interface inicial através da qual o usuário escolhe a opção de utilização do sistema.

Outra diferença do novo protótipo é que, agora, também é possível escolher o gênero das histórias a serem geradas. As informações do gênero estão contidas em arquivo com o contexto inicial a ser utilizado. No futuro, deverá ser feita a junção do Logtell com o CCM, o módulo citado no capítulo 4, que gerencia os contextos modelados em objetos. Nesse caso, o contexto, em vez de ser carregado de um arquivo, será recuperado do banco de dados do sistema. A Figura 6.5 mostra a janela de escolha do gênero de história, que na verdade são os contextos disponíveis.

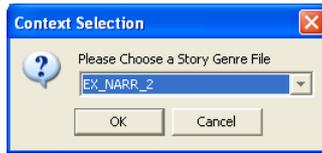


Figura 6.5: Escolha de Contexto

A classe principal do *Plot Manager* anteriormente era a classe *Plot*, que ainda existe na nova implementação, porém ela contém agora apenas as porções de código que são comuns a todos os modos de interação e foi transformada em uma classe abstrata.

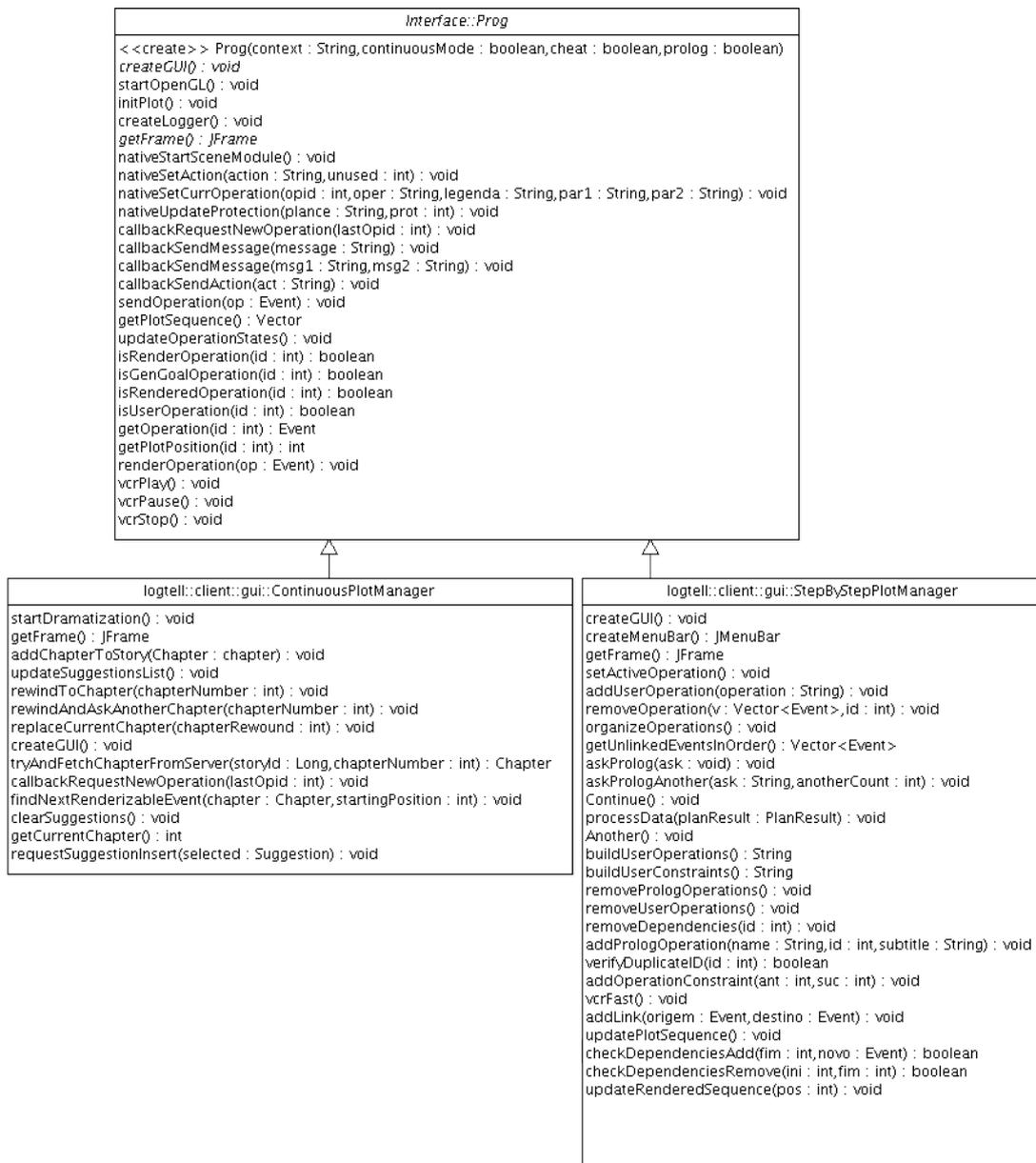


Figura 6.6: Novas classes para a interface com o usuário

Como se pode ver na Figura 6.6, a classe *Prog*, que era o núcleo do *Plot Manager*, agora é estendida pelas classes *ContinuousPlotManager* e *StepByStepPlotManager*. Seu código foi refatorado, reorganizado e em alguns trechos, reescrito, de forma a permitir essa nova arquitetura.

O *StepByStepPlotManager* implementa a interface passo-a-passo do Logtell, da mesma forma que a versão anterior do sistema, porém com a principal diferença de que, nessa nova implementação, ele não conversa diretamente com o IPG – agora o código faz chamadas remotas ao *SimulationControllerBean*, que é executado nos servidores de aplicação. O código do modo passo-a-passo utiliza somente este *bean* e tem como diferença com o modo contínuo, componentes gráficos para ordenar os eventos, além de opções para a inserção manual de eventos e situações, além dos botões para pedir novos ciclos de simulação, ou alternativas, como já demonstrado no capítulo 4. Há apenas uma pequena diferença visual, dado que foi aplicado um estilo de interface mais semelhante ao do Windows. Foi adicionada a opção “*Reset*”, dentro das opções, para que o usuário possa reiniciar sua história facilmente. A Figura 6.7 apresenta a interface no modo passo-a-passo (mantendo também os comandos para fazer perguntas sobre o estado da história, um botão auxiliar pra ordenar os eventos, além dos controles que aceleram/param a visualização da história).

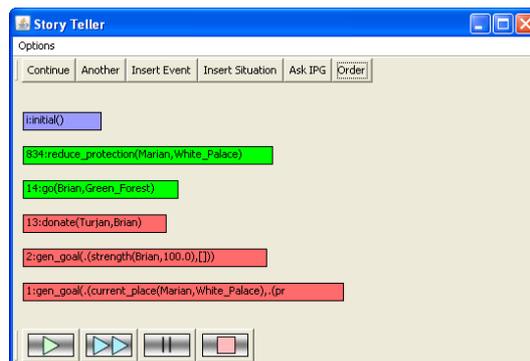


Figura 6.7: Interface do modo passo-a-passo.

O *ContinuousPlotManager* implementa o novo modo contínuo, tanto para o modo multiusuário quando para o de usuário único. Além de utilizar o *SimulationControllerBean* para solicitar os próximos capítulos a serem exibidos, ele também utiliza o *Interface Controller*, descrito na seção 6.4.

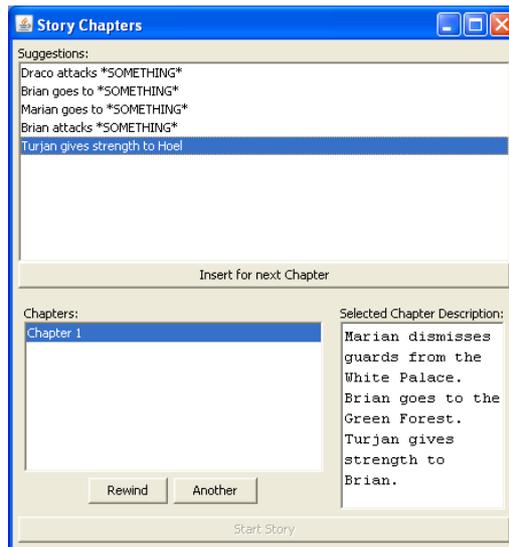


Figura 6.8: Interface do modo contínuo

A estratégia do modo contínuo difere bastante da estratégia do modo passo-a-passo. Em vez de se gerar trechos da história quando o usuário pede explicitamente, o início da história é solicitado automaticamente e, sempre que o motor gráfico 3D solicita novos capítulos para serem dramatizados, essa solicitação é repassada ao servidor. Dessa forma, o usuário está sempre assistindo à história que se desenrola, e pode interagir com a mesma, se assim o desejar.

A interação com o *Drama Manager* também difere um pouco, já que, em vez de se enviar uma seqüência total de eventos para serem dramatizados, os eventos são enviados aos poucos. Ao se iniciar a história, o primeiro capítulo é requisitado ao servidor que fornece o conteúdo desse capítulo. De posse do conteúdo do capítulo, o *ContinuousPlotManager* envia para o *Drama Manager* o primeiro evento a ser dramatizado. Conforme cada evento acaba, o próximo evento é requisitado pelo *Drama Manager*. Nesse momento, o *ContinuousPlotManager* busca, na sua cópia do capítulo atual, o próximo evento. Caso o capítulo tenha terminado, é feita uma chamada remota para receber o próximo capítulo do servidor, o qual já deverá estar pronto.

Atualmente, as cenas dramatizáveis pelo módulo 3D são simples e não demoram muito tempo. Por outro lado, os ciclos de geração, ao usarem a inferência de objetivos combinada com planejamento automático, podem demandar um tempo razoável, que depende da complexidade do contexto. Além disso, dependendo da carga a que o servidor esteja atendendo ou por problemas na rede, sempre é possível que existam atrasos da geração do enredo em relação à sua dramatização. Por isso, o modelo prevê o

uso de mecanismos para estender a dramatização de eventos correntes ou usar *fillers*, como, por exemplo, comerciais. No protótipo, esse mecanismo ainda não está implementado. Optou-se por simplesmente repetir requisições um certo número de vezes, desistindo-se da apresentação da história se o novo capítulo não for fornecido.

## 6.4 Interface Controller

Para lidar com os novos mecanismos de interação no modo contínuo propostos no capítulo 5, agora existe o *Interface Controller*. A Figura 6.9 mostra os seus métodos.

logtell::ejb::InterfaceControllerBean
makeSuggestion(desiredChapterNumber : int, storyID : Long, suggestion : Suggestion) : void
rewindStory(desiredChapterNumber : int, storyID : Long) : Chapter
rewindAndAskAnother(desiredChapterNumber : int, storyID : Long) : Chapter
getAvailableSuggestions(currentChapter : int, storyId : Long) : List

Figura 6.9: *InterfaceControllerBean*

As responsabilidades do *Interface Controller* incluem receber e tratar os comandos enviados pelos clientes no modo contínuo, incluindo os comandos *Rewind* e *Another*, e as solicitações de inclusão de sugestões enviadas pelos usuários. O comando *Rewind*, conforme descrito anteriormente, faz com que geração da história retroceda até um capítulo anterior já recebido no cliente. A execução do comando *Another* também faz um retrocesso, mas fornece, se possível, uma versão alternativa para o capítulo indicado. Além de tratar esses comandos, o *Interface Controller* também tem a responsabilidade de buscar para o cliente a lista das sugestões de intervenções fortes a serem disponibilizadas a cada momento.

Quando o usuário faz uma sugestão, o *InterfaceControllerBean* publica uma mensagem com essa sugestão, que será interceptada pelo processo que cuida da geração da narrativa com que se deseja interagir.

Para descobrir a lista de sugestões disponíveis, o *InterfaceControllerBean* faz uma chamada ao *SimulationControllerBean*, que então consulta o estado atual da história no capítulo que está sendo apresentado. Usando as estratégias disponíveis, descritas no capítulo 5, é gerada uma lista de sugestões de interferências a serem disponibilizadas para o(s) usuário(s).

Os novos mecanismos de interação fraca do modo contínuo, correspondentes aos comandos *Rewind* e *Another*, também funcionam de maneira similar à inserção de

sugestões. Uma mensagem é enviada para o processo da história em questão, para que o *ContinuousStoryWriter* prepare-se para reescrever os capítulos futuros, conforme necessário, e só então o capítulo desejado é retornado. A execução de um comando *Rewind* é praticamente instantânea, pois demanda apenas o tempo de recuperar o *snapshot* do capítulo para o qual se deseja voltar no banco de dados. No caso do comando *Another* é de se esperar que exista algum atraso, dado que é necessária uma consulta ao IPG para obter a alternativa ao capítulo que havia sido apresentado. Deve-se ressaltar que tanto em um caso quanto em outro, a geração do enredo é trazida para um ponto anterior e, dependendo de novas interações, histórias completamente diferentes podem ser obtidas.

## 6.5. Conclusões

Neste capítulo foi apresentada uma visão geral do protótipo que foi implementado com base modelo apresentado no capítulo 5. Essa implementação serviu para demonstrar a aplicabilidade de parte das idéias propostas.

No novo protótipo, uma série de melhorias foram feitas na própria estrutura e organização do Logtell. O código, no que tange ao controle da dramatização, foi praticamente todo refatorado de modo a atender aos requisitos do novo modelo. Além disso, o ambiente de *storytelling* interativo funciona agora em uma arquitetura cliente-servidor, utilizando *frameworks* que são usados e aceitos no mercado, com forte grau de maturidade tecnológica. Uma das principais qualidades do protótipo é exatamente a conformidade com essas estratégias que são atualmente utilizadas em aplicações comerciais e empresariais.

O uso de um ambiente *Enterprise*, implementado em um servidor de aplicação como o JBoss, facilitou a adequação do sistema de *storytelling* interativo a novos requisitos funcionais e não funcionais. Dentre os requisitos funcionais que puderam ser mais facilmente implementados, podem ser destacados a criação de histórias em ambiente cliente-servidor e a possibilidade de interoperabilidade com diferentes clientes. Ao se implementar o sistema em uma plataforma onde podem ser conectados diferentes servidores, tornou-se bem mais fácil atender ao importante requisito não funcional de escalabilidade.

Não foi feita ainda a integração com um banco de dados para armazenar os *snapshots* das histórias. Os *snapshots*, por enquanto, são guardados em memória principal, mas o armazenamento em banco de dados poderá ser facilmente realizado,

dado que o código foi escrito de forma organizada, usando padrões como DAOs. Esses padrões isolam o acesso aos *snapshots* das demais partes do sistema, sendo necessárias poucas mudanças no código para alterar a forma de armazenamento.

Os mecanismos de criação de sugestões para os usuários, por enquanto, estão limitados a regras de inferência de sugestões a serem oferecidas aos usuários. Esse mecanismo depende bastante de um esforço autoral. Para gerar sugestões com menor esforço, outros mecanismos, também propostos no capítulo 5, os quais são baseados no reconhecimento de planos típicos e na análise de regras de inferência de objetivos poderão ser bastante úteis.

Os mecanismos de votação e inserção das sugestões também podem ser aperfeiçoados. Aceita-se, por enquanto, uma única sugestão por capítulo. A conciliação de duas intervenções fortes em um mesmo capítulo poderá levar a resultados interessantes, mas o mecanismo de controle precisará balancear a complexidade das interações permitidas com a demanda de manutenção de um fluxo contínuo na apresentação.

Existem ainda aperfeiçoamentos a serem feitos na forma como os clientes tratam atrasos dos servidores no fornecimento de novos capítulos. Procurou-se desenvolver mecanismos para evitar os atrasos, mas, dependendo da demanda, eles podem ser inevitáveis. Nesse caso, a incorporação ao protótipo de mecanismos de alongamento do capítulo corrente ou de inserção de *fillers* tornam-se necessários.

Outras melhorias futuras incluem a incorporação dos mecanismos avançados de interação previstos no capítulo 5. Tais mecanismos incluem o uso de linguagem natural e de conceitos abstratos para a inserção de sugestões nas histórias e a manipulação, por parte dos usuários, de escalas numéricas correspondentes a tensões narrativas que influenciam a evolução das histórias.

Apesar da existência de aperfeiçoamentos a serem feitos, procurou-se atender, na versão atual do protótipo, aos requisitos básicos do modelo proposto no capítulo 5, de modo a permitir a sua validação. No capítulo 7, é apresentada a aplicação do protótipo em um contexto exemplo com a análise dos resultados obtidos.

## 7. Aplicação do Protótipo

Para fins de avaliação do modelo proposto no capítulo 5, optou-se por utilizar o protótipo descrito no capítulo 6 com o mesmo contexto exemplo do gênero “Espadas e Dragões”, descrito no capítulo 4, o qual serviu de base para a validação da primeira versão do Logtell. Como o objetivo principal do trabalho é propor um modelo viável de *storytelling* interativo para o ambiente de TV interativa e não a obtenção de um produto pronto para uso com entretenimento, o contexto exemplo foi modificado apenas nos pontos necessários para testar o atendimento aos requisitos do modelo.

No capítulo 4, pode-se encontrar a descrição do contexto exemplo, com os predicados que descrevem os personagens, lugares e as relações entre eles nas histórias, as operações que especificam os eventos que podem ocorrer e também as regras de inferência de objetivos.

Para os testes apresentados neste capítulo, foi definida uma configuração inicial a partir da qual as histórias são desenvolvidas. Nessa configuração *Brian* e *Hoel* são cavaleiros, enquanto *Marian* é uma Princesa, *Turjan* um Feiticeiro e *Draco* um Dragão. *Brian*, *Hoel* e *Marian* são de natureza boa, enquanto que *Turjan* é neutro e *Draco* é mau. Todos estes personagens estão vivos no começo da história. *Brian* e *Hoel* são mais fracos do que o Dragão, *Marian* é mais fraca ainda, e *Turjan* é tão forte quanto *Draco*. *Brian* e *Hoel* têm forte afeto por *Marian*, mas esta não retribui. Além disso, cada um dos personagens tem um papel, informação que é utilizada na geração das histórias: *Brian* e *Hoel* são heróis, *Draco* é um vilão, *Marian* é uma vítima e *Turjan* é um “doador”. O doador é um papel especial que, neste gênero, é alguém capaz de doar poder ou então de enfeitiçar este outro personagem, trocando sua natureza.

Além dos personagens e seus papéis e relações, também são definidas localidades. Cada localidade pode ser de natureza boa, má ou neutra e pode estar protegida ou desprotegida. Locais protegidos contêm guardas que protegem personagens que são da mesma natureza do local. Existem 5 localidades: os castelos Branco (*White Palace*), Cinza (*Gray Palace*) e Vermelho (*Red Palace*), além da Floresta Verde (*Green Forest*) e da Igreja (*Church*). O Castelo Cinza é de natureza boa e é desprotegido. Além disso, é a residência de *Brian* e *Hoel*. O Castelo Branco é a residência de *Marian*. É um local de natureza boa e está fortemente protegido. *Turjan* vive na Floresta Verde, que é neutra como ele e tem um certo nível de proteção. *Draco* vive em sua fortaleza, o Castelo Vermelho, também com certo nível de proteção. Finalmente, a Igreja é um local

de natureza boa, mas que está desprotegido. O nível de proteção de um local é uma informação fundamental na hora da construção da história, dado que um personagem só poderá, por exemplo, seqüestrar outro personagem ou lutar com ele, se esse outro personagem estiver em um local suficientemente desprotegido.

As operações que especificam os eventos possíveis e as regras de inferência de objetivos estão listadas nas tabelas do capítulo 4.

Para fazer os testes, foi utilizado um servidor Athlon 64 3000+ (2 Ghz) com 1.5 GB de RAM e Windows XP SP3. De forma ideal, mais testes deveriam ter sido realizados com mais clientes e um servidor, mas os testes aqui relatados foram feitos utilizando apenas o computador citado. Para os testes das seções 7.1 a 7.3, foi utilizada somente esta máquina como principal ambiente de testes, e servidor e cliente eram executados em conjunto. Mesmo assim, foram feitos outros testes não monitorados com o ambiente em rede com outros computadores mais potentes e o sistema também funcionou, usando uma máquina cliente e outra como servidor.

Nas seções seguintes, são descritos testes realizados e resultados obtidos que visaram avaliar a adequação do modelo e do protótipo aos principais requisitos para a utilização de *storytelling* interativo em um ambiente de TV interativa.

## **7.1 Continuidade do Fluxo**

Um dos principais requisitos do modelo proposto é a capacidade de se manter uma continuidade do fluxo da história, sempre que possível. Isto é, deve ser possível assistir à história conforme ela está sendo gerada, evitando-se, tanto quanto possível, as interrupções.

Foi feita, inicialmente, a verificação de interrupções quando a história é gerada e dramatizada sem interferência do usuário. Nesse caso, verificou-se que não houve nenhuma interrupção perceptível. Capítulos eram gerados e apresentados no modo contínuo sem que se ocorressem interrupções perceptíveis entre um capítulo e outro. Os tempos de geração de um novo capítulo e de seu envio aos clientes mostraram-se compatíveis com o tempo de dramatização do capítulo corrente. Em situações onde pode haver problemas na rede, isso poderia não ser verdade, dado que problemas na velocidade da conexão poderiam influenciar o resultado final e a performance do sistema. Nos testes realizados em ambiente de rede local, isso não ocorreu. Por outro lado, apesar de os testes terem sido feitos em ambiente local, verificou-se que, mesmo com o servidor de aplicações sendo executado no mesmo computador do cliente, o que

poderia provocar uma sobrecarga, não ocorreram interrupções. A Tabela 7.1 lista os tempos medidos para a recepção dos capítulos em uma utilização do sistema sem interferência do usuário. Pelo fato de os intervalos serem mínimos, demonstrou-se que a continuidade não foi comprometida nesse caso mais simples.

<b>Capítulo</b>	<b>Descrição dos eventos</b>	<b>Tempo até cliente receber o capítulo (em milissegundos)</b>
1	Marian dismisses guards from the White Palace. Brian goes to the Green Forest. Turjan gives strength to Brian.	156
2	Draco goes to the White Palace. Draco attacks the White Palace. Draco kidnaps Marian.	31
3	Brian goes to the Red Castle. Brian attacks the Red Castle. Draco fights against Brian. Brian kills Draco. Brian frees Marian.	47
4	Brian goes to the Church. Marian goes to the Church. Brian and Marian get married.	94

Tabela 7.1: Exemplo de narrativa sem interferências.

Para evitar problemas quando ocorrem falhas de comunicação, ainda há espaço para otimizar a estratégia de recuperação dos capítulos futuros. Atualmente, um novo capítulo a ser exibido só é recuperado quando o último evento do capítulo atual acaba de ser apresentado. Como uma possível estratégia, a solicitação de um novo capítulo ao servidor poderia ser antecipada, ocorrendo antes da apresentação de alguns eventos do capítulo corrente. Consegue-se, dessa forma, reduzir o impacto que uma conexão de rede mais lenta poderia ter no processo. Entretanto, é importante ressaltar que, no modo contínuo, as informações correspondentes aos capítulos são estruturas bem simples. Em condições normais, o simples envio desses dados para os clientes não deve causar atrasos. A questão principal seria então apenas a compatibilização entre o tempo de geração de um novo capítulo no servidor enquanto o capítulo anterior é apresentado no cliente. No caso da geração de histórias sem interferência, verificou-se que o processo funciona sem maiores problemas.

Foram realizados testes com o uso das novas formas de interação fraca, que foram implementadas no novo modo contínuo. Com o comando *Rewind*, que tem a função de voltar o processo de geração até o momento em que um certo capítulo passado foi gerado, o comportamento esperado do sistema é de pouca demora, dado que

os capítulos anteriores e seus respectivos retratos (*snapshots*) da história já estão gravados.

<b>Capítulo sendo dramatizado</b>	<b>Capítulo requisitado pelo Rewind</b>	<b>Tempo em milissegundos até capítulo chegar na interface (em milissegundos)</b>
1	1	235
2	1	360
3	2	266

Tabela 7.2: Alguns tempos para o *Rewind*

Como a tabela 7.2 demonstra, os resultados são compatíveis com o esperado: o tempo que demora após um comando de *Rewind* é desprezível, no ambiente em que foi testado. Como descrito antes, ao fazer um *Rewind*, o sistema apenas recupera as informações do retrato da história no passado e as envia para o cliente, levando um tempo desprezível a mais porque interrompe a dramatização brevemente. É importante ressaltar que, no protótipo atual, além de fazer isso, os capítulos seguintes que já haviam sido gerados e possivelmente dramatizados são descartados, pois, com interações dos usuários em pontos anteriores a eles, eles se tornariam inconsistentes. Uma otimização possível seria mantê-los enquanto não ocorresse alguma interação que os invalidasse.

Os resultados obtidos com o uso da funcionalidade de *Rewind* mostram que o tempo para o retorno a um capítulo anterior é compatível com o tempo para retorno a um capítulo anterior em um DVD (em torno de 1 segundo), podendo ser até menor, dependendo do aparelho.

<b>Capítulo sendo dramatizado</b>	<b>Capítulo requisitado pelo Another</b>	<b>Tempo em milissegundos até capítulo chegar na interface (em milissegundos)</b>
1	1	5219
2	2	5250
3	2	5203

Tabela 7.3: Alguns tempos para o *Another*

A execução de um comando *Another* é bem semelhante à execução de um

comando *Rewind*, com a diferença de que, além de se solicitar a volta do processo de geração a um ponto anterior da história, pede-se que seja fornecida uma versão alternativa do capítulo selecionado. Como essa operação envolve, além da recuperação de um *snapshot*, a solicitação de uma alternativa do capítulo ao IPG, é razoável que esta operação não seja instantânea. Como a tabela 7.3 demonstra, a operação demora algum tempo, porém ainda muito pequeno (na maioria dos capítulos gerados), dado que, para o IPG o custo de pedir uma nova alternativa é menor do que o tempo normal de inferência e planejamento de objetivos numa fase de simulação de um novo capítulo.

De um modo geral, verificou-se que, em condições ideais, obteve-se resultado plenamente satisfatório no que tange à continuidade do fluxo quando ocorrem intervenções fracas. Por condições ideais entenda-se a situação onde as requisições decorrentes de comandos *Rewind* e *Another* são atendidas por servidores que não estão sobrecarregados. Como já explicado antes no capítulo 6, devido a limitações no protótipo atual, apenas uma instância do IPG pode existir por servidor de aplicação, o que significa que, se essa instância estiver ocupada, por exemplo, em escrever o capítulo futuro de uma história, o tempo necessário para atender a uma requisição de *Another* para outra história será logicamente maior. Nesse caso, foram obtidos tempos que variavam de 15 segundos até 30 segundos, dado que as chamadas ao IPG são enfileiradas.

Quanto aos casos em que se fazem inserções na história no modo contínuo, ou seja, quando se utiliza a funcionalidade de inserção de sugestões na história, no ambiente ideal descrito antes, não ocorrem atrasos. Isso acontece porque as inserções de sugestões, dentro do novo mecanismo, ocorrem de forma assíncrona para os usuários, ou seja, o cliente envia uma sugestão e o servidor então a redistribui para o processo que está escrevendo a história. Uma das questões principais no que concerne ao fluxo contínuo e às inserções de sugestões é justamente o tempo que se leva para processar as mesmas. Nos casos em que a sugestão é enviada em um momento muito próximo do fim da dramatização de um determinado capítulo na interface do cliente, o que ocorre é que o tempo para processar a sugestão pode demorar demais. Além disso, quanto mais inapropriada for a sugestão, potencialmente mais tempo se levará para que o IPG tente utilizá-la. Deve-se ressaltar, por isso, a necessidade futura da implementação de mecanismos que criem sugestões tanto promissoras para as histórias quanto o possível.

Quando a inserção de uma sugestão é feita e o IPG não tem tempo suficiente para incorporá-la antes que o primeiro cliente solicite o próximo capítulo, é fornecido

um capítulo que não a incorpora e a versão que incorpora a sugestão é descartada. Isso ocorre porque não faz sentido modificar um capítulo que já está sendo dramatizado em algum cliente. De fato, como algumas sugestões podem levar um tempo grande para a geração do capítulo que as incorpora, estratégias precisam ser adotadas de modo a reduzir o número de situações em que sugestões coerentes são descartadas.

Para se obter melhores resultados em situações críticas, uma alternativa é a geração prévia de capítulos alternativos, a qual não foi ainda implementada. Quando a solicitação de alternativas para um capítulo fosse feita através de um comando *Another* ou da inserção de sugestões, a alternativa já estaria pronta para ser fornecida ao cliente. No caso das sugestões, elas poderiam até mesmo ser testadas antes de serem disponibilizadas aos usuários. Para que isso não criasse um gargalo no sistema, seria importante ter vários servidores com cada um deles executando várias instâncias do IPG. Conforme foi mencionado antes, no capítulo 5, o melhoramento do algoritmo de planejamento e o controle do tempo de duração dos eventos também podem ser determinantes para a obtenção de resultados bons em situações críticas.

## **7.2 Diversidade e Coerência das Histórias**

Como forma da verificação da diversidade e coerência das histórias que o protótipo é capaz de gerar, tem-se o seguinte critério em mente: se o novo modelo for capaz de incorporar todos os mecanismos de intervenção do Logtell original, subentende-se que é capaz então de herdar a diversidade e coerência das histórias geradas.

Na versão anterior, que implementava apenas o modo de interação passo-a-passo, o usuário era capaz de influenciar a história basicamente de 3 maneiras: inserindo “sugestões” (apesar de não terem esse nome, as interações fortes de inserção de eventos e situações, de certo modo, sempre foram “sugestões”, dado que o Logtell as rejeita quando não são coerentes com a história); pedindo a geração dos próximos eventos da história e pedindo alternativas.

De fato, no modo contínuo que foi implementado, o usuário é capaz de alcançar basicamente as mesmas histórias obtidas no modo passo-a-passo. O mecanismo de “continuar” a história já é implícito, dado que, no modo contínuo, a história está sempre sendo escrita à frente. Quanto às versões alternativas, elas também podem ser obtidas de forma até mais flexível, dado que, anteriormente, o usuário somente poderia pedir alternativas para o último trecho da história que foi criado pelo IPG e, agora,

alternativas podem ser pedidas para qualquer capítulo anterior. Finalmente, a inserção de eventos e situações agora ocorre por meio da seleção de sugestões a serem incorporadas. O único mecanismo de interação, disponível para o usuário no modo passo-a-passo, que não é possível no modo contínuo é o estabelecimento da ordem total dos eventos, dado que essa ordenação ocorre automaticamente. No entanto, esse mecanismo tende a gerar pouca variedade no resultado final das histórias. Dessa forma, é possível dizer que, no modo contínuo, obtém-se o mesmo nível de coerência do modo passo-a-passo e, aproximadamente, o mesmo nível de diversidade.

Contudo, há de se admitir que, no modo passo-a-passo, existe uma liberdade maior na inserção de eventos e sugestões, liberdade essa que foi cerceada no modo contínuo. Isso foi necessário porque, em uma interação em modo contínuo, é natural que exista menos tempo para tomar decisões a respeito do que se deseja modificar na história. Se as sugestões cobrirem as principais possibilidades, tende-se a obter o mesmo tipo de resultado no modo contínuo e no modo passo-a-passo. Ressalta-se, por isso, a importância de se dispor de mecanismos que forneçam sugestões potencialmente interessantes e coerentes. De qualquer maneira, como já foi explicado antes, o modo passo-a-passo ainda é permitido na nova arquitetura, sendo útil, em particular, para a análise das histórias que são geradas.

Com o uso dos modos de interação fraca disponíveis no modo contínuo, representados pelos comandos *Rewind* e *Another*, é possível obter praticamente todos os enredos geráveis sem intervenções fortes no modo passo-a-passo.

Apenas com o uso do comando *Another*, foi possível gerar histórias diferentes. Abaixo, temos um exemplo de história gerada, com os seguintes capítulos:

1. *Draco goes to the White Palace. Draco attacks the White Palace. Brian goes to the Green Forest. Turjan gives strength to Brian.*
2. *Draco attacks the White Palace. Draco kidnaps Marian.*
3. *Brian goes to the Red Castle. Brian attacks the Red Castle. Draco fights against Brian. Brian kills Draco. Brian frees Marian.*
4. *Brian goes to the Church. Marian goes to the Church. Brian and Marian get married.*

Nesse exemplo, obtido com aplicação de um comando *Another* para o primeiro capítulo, nota-se que a história foi diferente da apresentada na Tabela 7.1. *Draco* ataca o

castelo e reduz sua proteção, enquanto que, na história original, *Marian* “colaborava” com *Draco*, dispensando guardas. Outro exemplo, obtido com um *Another* para o terceiro capítulo é o seguinte:

1. *Marian dismisses guards from the White Palace. Brian goes to the Green Forest. Turjan gives strength to Brian.*
2. *Draco goes to the White Palace. Draco attacks the White Palace. Draco kidnaps Marian.*
3. *Brian goes to the Red Castle. Hoel goes to the Red Castle. Hoel attacks the Red Castle. Draco fights against Brian. Brian kills Draco. Hoel frees Marian.*
4. *Hoel goes to the Church. Marian goes to the Church. Hoel and Marian get married.*

Como se pode notar, neste exemplo uma história diferente foi criada, e possui uma peculiaridade: embora os dois heróis tenham atacado o castelo juntos, a princesa é grata a quem a liberta, como definido nas regras do contexto; portanto, embora os dois tenham mérito, é com *Hoel* que ela se casa, mesmo tendo sido *Brian* quem matou o dragão *Draco*.

Também com o uso das intervenções fortes, é possível obter praticamente todos os enredos possíveis do contexto exemplo, desde que existam sugestões que explorem essas alternativas, e os mecanismos do modelo para a geração destas.

Para os testes com o protótipo, como alternativa inicial para permitir intervenções fortes, foi feito um esforço autoral que teve como resultado a criação de uma série de sugestões, que para o contexto mapeado, costumam ser, mais ou menos promissoras, dependendo do momento em que são inseridas na história. Essas regras foram adicionadas ao contexto inicial das histórias e foram usados para testar o protótipo.

Foi mapeado um conjunto de regras, cada uma com um momento mais adequado para serem inseridos nas histórias, que são:

- No início da história:
  - *go(marian, church).*
  - *go(marian, green\_forest).*
  - *attack(draco, white\_palace).*
  - *fight(brian, hoel).*

- *reduce\_protection(marian,white\_palace).*
- Com a vítima frágil:
  - *kidnap(draco,marian).*
  - *kill(draco,marian).*
- Com a vítima raptada:
  - *free(brian, marian).*
  - *free(hoel,marian).*
  - *kill(hoel,draco).*
  - *kill(brian,draco).*
- Com a vítima morta:
  - *kill(brian,draco).*
  - *kill(hoel,draco).*
- Com a vítima libertada:
  - *marry(brian, marian).*
  - *marry(hoel,marian).*
  - *kill(brian,hoel).*
  - *kill(hoel,brian).*

Com o uso das sugestões, foi possível criar histórias diferentes das que seriam criadas naturalmente pelo IPG. Dessa forma, podem-se explorar alternativas coerentes segundo o gênero.

Por exemplo, ao inserir a sugestão “*Draco kills Marian*” no primeiro capítulo, obteve-se a seguinte história:

- *Marian dismisses guards from the White Palace. Marian goes to the Red Castle. Brian goes to the Green Forest. Turjan gives strength to Brian. Draco kills Marian.*
- *Brian goes to the Red Castle. Brian attacks the Red Castle. Draco fights against Brian. Brian kills Draco.*

Nessa história, como foi inserido o evento do assassinato de uma vítima, a regra de inferência da vingança foi ativada, e, portanto, *Brian* terminou por vingar o assassinato de *Marian*. Em outra história, ao inserir “*Hoel kills Draco*” no segundo

capítulo, obteve-se:

- *Marian dismisses guards from the White Palace. Brian goes to the Green Forest. Turjan gives strength to Brian.*
- *Hoel goes to the Red Castle. Hoel attacks the Red Castle. Brian goes to the Red Castle. Draco goes to the White Palace. Draco attacks the White Palace. Draco kidnaps Marian. Brian fights against Draco. Hoel kills Draco.*
- *Hoel frees Marian.*
- *Marian goes to the Church. Hoel goes to the Church. Hoel and Marian get married.*

Nessa história, acontece uma estrutura diferente: os dois heróis de fato lutam contra o dragão *Draco* juntos, mas é *Hoel* quem dá o golpe final. Logo após, ele liberta *Marian*, ficando então mais justa a gratidão de *Marian* por *Hoel*.

### **7.3 Facilidade e Comodidade da Utilização**

O ambiente de TV interativa impôs ao modelo a necessidade de fornecer mecanismos fáceis e cómodos para a interação, pois o usuário não tem muito tempo para se concentrar na interação enquanto assiste à história.

A inserção de sugestões e a execução de comandos de *Rewind* e *Another* pode ser solicitada com apenas dois cliques. Uma única interferência dessas pode ser suficiente para a obtenção de uma história completamente diferente. Acredita-se, então, que esses mecanismos são condizentes com a possibilidade de interação preguiçosa (*lazy interaction*) típica para ambientes de TV interativa. Note-se que a execução desses comandos, que hoje ocorre através do *mouse*, poderia ter que ser adaptada quando o cliente estivesse interagindo através do celular ou de sua TV.

Além dos mecanismos básicos, o modelo propôs outros mais avançados e que não chegaram a ser implementados ainda. O uso de linguagem natural, especialmente se usado em conjunto com reconhecimento de voz, pode ser bastante útil em contexto de TV digital. O uso de conceitos abstratos em sugestões é dramaticamente interessante pelo fato de permitir sugestões vagas cuja incorporação de uma forma concreta nos enredos pode surpreender o usuário. Outro mecanismo proposto é o controle, por parte do usuário, de variáveis numéricas que interfiram na simulação, determinando, por

exemplo, níveis de suspense, romance, violência, etc. Mecanismos como esse podem ser implementados de forma análoga ao controle de volume e permitem uma forma de interação indireta que tem bom potencial para surpreender o usuário.

Um fato que demonstra o nível de facilidade e comodidade da utilização dos mecanismos implementados é o número de interações necessário para se obter uma determinada história. A primeira história citada na seção 7.2 foi obtida usando o comando *Another* 2 vezes no capítulo 1; a segunda foi um pouco mais trabalhosa, mas mesmo assim fácil de ser feita, usando o comando *Another* 4 vezes no capítulo 3 (o único esforço do usuário foi o de ver o capítulo gerado, e então, solicitar outra opção, sem ter que reiniciar todo o processo); na terceira e na quarta histórias, foram necessárias somente inserções de sugestões pontuais: na terceira, foi inserido um evento no capítulo 1, e, na quarta, um evento no capítulo 2.

Os mecanismos previstos no modelo não esgotam o assunto. Dentro do ambiente de *storytelling* interativo, existem muitas outras possibilidades que ainda não foram exploradas. Mecanismos de realidade aumentada, por exemplo, podem ser especialmente úteis para usuários com dificuldades motoras e para crianças.

#### **7.4 Escalabilidade**

O protótipo utiliza o JBoss, que é uma plataforma que procura fornecer escalabilidade e confiabilidade na distribuição de recursos em rede. Entretanto, como foi explicado, o protótipo ainda não está pronto para trabalhar com vários servidores simultaneamente. Para tal, seria preciso usar um banco de dados na implementação dos DAOs e também configurar o ambiente para executar em *cluster*.

O Logtell já funciona agora em um ambiente J2EE cliente-servidor, só que com um único servidor. As mudanças na configuração do sistema e na camada que faz acesso ao banco de dados (que, por hora, é virtualmente implementado em memória) não afetariam o código do restante do sistema. Dessa forma, é possível afirmar que a base necessária para se obter escalabilidade já foi criada.

Uma questão pertinente à escalabilidade já foi elucidada antes: o fato do acesso ao IPG ser feito através de uma biblioteca que não suporta bem múltiplas instâncias na mesma máquina virtual. Entretanto, isso deverá ser resolvido em futuro próximo. Como o código que faz acesso ao IPG é bem encapsulado, o esforço de alteração do *Simulation Controller* não será grande.

Mesmo assim, existe uma verificação que pôde ser feita: o uso do modo

multiusuário. Utilizando o modo multiusuário, foi possível assistir a uma história em modo contínuo compartilhado. Testes feitos em rede local mostraram que não houve atrasos no tempo de transmissão dos capítulos para os clientes. Isso ocorre porque, na verdade, em uma história compartilhada, a geração é feita por um único processo, independente do número de clientes que a assistem. As únicas diferenças em termos de carga do servidor ocorrem na comunicação com os clientes. Como a comunicação com os clientes é bem mais leve que a geração do enredo, não é de se estranhar que o custo de processamento de uma história contínua multiusuário seja aproximadamente equivalente ao custo de uma história contínua para apenas um usuário. A Figura 7.1, a seguir mostra um teste onde o servidor e dois clientes são executados em uma mesma máquina e os dois clientes compartilham uma mesma história. Nesse teste, a mesma história é dramatizada em duas janelas diferentes, sem interrupção de fluxo e permitindo, inclusive, a incorporação de intervenções fortes.

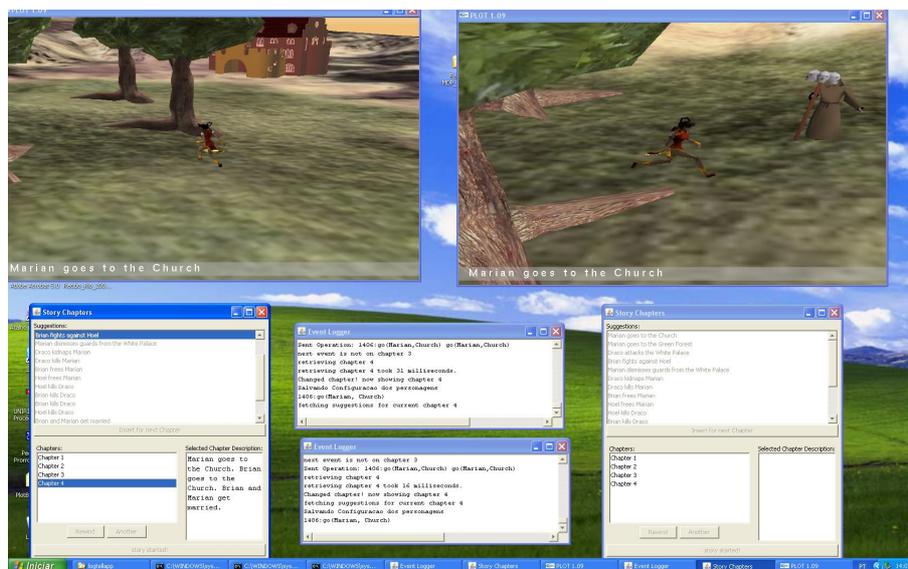


Figura 7.1 Modo multiusuário com dois clientes assistindo à mesma história

## 8. Conclusões e Trabalhos Futuros

### 8.1 Considerações Gerais

Nesta pesquisa, foi apresentado um novo modelo de *storytelling* interativo, que busca atender aos requisitos de alta responsividade da TV interativa. O modelo estende o que foi originalmente definido no sistema Logtell, tendo como principal preocupação a conciliação entre a coerência das histórias e a continuidade do fluxo de apresentação. Outras características como a diversidade das histórias, a definição de mecanismos de interação próprios para o meio e a escalabilidade também foram considerados.

Apesar de inspiradas pelas demandas de responsividade de ambientes como o da TV digital, as soluções propostas são de natureza interoperáveis, e compatíveis com outros ambientes tais como a Web.

Para a validação das idéias, foi implementado um novo protótipo do Logtell que passou a ser executado agora em rede, segundo uma arquitetura cliente-servidor. O protótipo foi testado com base no mesmo contexto exemplo de histórias usado para validar a primeira versão do Logtell, correspondente ao gênero “Espadas e Dragões”.

Os resultados obtidos com este trabalho mostram, de forma geral, a viabilidade da utilização de *storytelling* interativo para TV, em um tipo de aplicação onde a coerência dos enredos é valorizada. A seção 8.2 faz um relato mais detalhado das principais contribuições. Uma grande quantidade de trabalhos futuros potenciais poderão ser feitos usando como base o modelo proposto e o protótipo implementado. A seção 8.3 destaca as principais possibilidades.

### 8.2 Principais Contribuições

#### **Estudo da Conciliação entre Coerência e Continuidade do Fluxo**

Devido às características do meio, a coerência e a diversidade das histórias ganham maior importância em um ambiente de TV interativa. Para garanti-las, escolheu-se utilizar a simulação baseada em lógica formal adotada no Logtell. Obtém-se assim a flexibilidade para a geração de um conjunto grande de alternativas coerentes. No entanto, o tempo de simulação pode não ser irrelevante e, na TV, as histórias precisam ser apresentadas sem interrupção, para não comprometer a responsividade. Para conciliar esses requisitos, foram definidas estratégias específicas de distribuição do processamento, de controle do fluxo e de coordenação entre a produção de enredos no servidor e a sua apresentação nos clientes. Os testes realizados com o protótipo mostram

que o modelo viabiliza essa conciliação entre coerência e responsividade, que foi o principal foco deste trabalho. Além disso, o modelo prevê estratégias que poderão vir a ser implementadas para garantir essa conciliação em condições mais críticas, tal como a utilização paralela de várias instâncias do módulo gerador de enredos.

### **Estudo de Métodos de Interação**

Para permitir a apresentação contínua do conteúdo em paralelo com a interação com o usuário, foram propostos alguns métodos básicos de interação, os quais incorporam basicamente todas as possibilidades oferecidas pelo Logtell para interação fraca e forte com as histórias. Os novos métodos criados para o modo de interação contínuo, demandam pouco esforço do usuário e, virtualmente, permitem a obtenção de qualquer história que pode ser obtida no modo passo-a-passo. Além disso, foram propostos, no modelo, métodos mais avançados que possibilitariam interações ainda mais cômodas. Foram também estudados e implementados os mecanismos para que usuários pudessem interagir de forma conjunta com uma mesma história que estejam compartilhando.

### **Proposta de Arquitetura**

A adoção de uma arquitetura cliente-servidor é uma solução natural dentro de um ambiente distribuído como a TV interativa. Foi feita uma distribuição de carga de processamento entre clientes e servidores, na qual o controle do processo e a geração dos enredos são executados no servidor e a interface com o usuário e a dramatização ficam no cliente. A geração dos enredos no servidor faz sentido em virtude da maior demanda de recursos computacionais. Essa solução permite que o tráfego na rede possa ser bastante reduzido, mas exige que os clientes tenham poder de processamento suficiente para executar o módulo de dramatização.

Uma alternativa com centralização da dramatização no servidor e envio de vídeo em *broadcast* pode, eventualmente, ser conveniente em virtude de limitações computacionais das *set-top boxes*. No entanto, para isso, algumas modificações teriam que ser introduzidas no protótipo.

A adoção de um servidor de aplicação que permite a ligação de vários servidores em *cluster* possibilitará que, no futuro, o protótipo seja adaptado sem muito esforço para atender a um grande conjunto de clientes, como é esperado em um ambiente de TV interativa. Além disso, o uso de técnicas que facilitam a interoperabilidade foi

importante para permitir que o sistema de *storytelling* interativo seja usado em diferentes ambientes, com transmissão terrestre digital, por IPTV; pela Web, etc.

### **Base para Pesquisa em *Storytelling* Interativo**

O protótipo, que foi implementado de forma modular, servirá como base para estudar a incorporação de diversos mecanismos através do desenvolvimento de novas versões. Além disso, com a colocação do sistema em rede, será muito mais fácil disponibilizá-lo a um grande conjunto de usuários, que poderão testá-lo e dar retorno importante para o direcionamento de pesquisas futuras.

### **Publicações**

Ao longo da pesquisa, trabalhos submetidos foram aceitos para serem publicados (CAMANHO *et al.*, 2008; CIARLINI *et al.*, 2008). Desta forma, isto serviu como um indicativo de que os propósitos da pesquisa são válidos, dados que foram aceitos pela comunidade científica, inclusive em eventos internacionais.

## **8.3 Trabalhos Futuros**

### **Aperfeiçoamentos no Protótipo**

Há um conjunto de aperfeiçoamentos no protótipo que deverão ser feitos para permitir a implementação completa do modelo de *storytelling* interativo proposto no capítulo 5.

Atualmente, as sugestões de interação forte são obtidas apenas através do uso de regras de inferência de sugestão, as quais demandam um esforço autoral considerável. O modelo prevê também alternativas mais automáticas, como o uso de reconhecimento de planos associados a estruturas recorrentes nos gêneros narrativos (motivos). Em (KARLSSON *et al.*, 2006) são descritos mecanismos que permitem a incorporação ao enredo de planos típicos reconhecidos no modo passo-a-passo do Logtell. A idéia básica é adaptar esses mecanismos para obter as sugestões de interação forte no modo contínuo. Com esses novos mecanismos, espera-se obter, com mais facilidade, sugestões que levem a enredos surpreendentes e diversos.

Para permitir a utilização simultânea do protótipo por um grande número de usuários, algumas modificações no protótipo serão necessárias. Dentre estas, inclui-se a integração com um banco de dados para o armazenamento dos *snapshots* das histórias, de

modo que vários servidores possam trabalhar em paralelo, controlando um mesmo grupo de histórias. Outra limitação do protótipo atual é o fato de que o acesso ao IPG ainda é feito através de uma biblioteca que obriga a existência de apenas uma instância da conexão com o Prolog por máquina virtual. Futura versão do sistema deverá eliminar esse problema com mudanças nos métodos de acesso e, eventualmente, do ambiente Prolog usado pelo IPG.

Com a possibilidade de trabalho de servidores em *cluster* e com várias instâncias do IPG por servidor, será possível obter melhores resultados em situações críticas onde o servidor único de instância única fica sobrecarregado.

### **Novas Implementações e Extensões do Modelo de Storytelling Interativo**

Existem outras pesquisas em andamento que visam a extensão do Logtell e que podem ser integradas com o novo modelo proposto neste trabalho.

A pesquisa descrita em (DORIA *et al.*, 2008) é uma das mais relacionadas ao trabalho desta dissertação. Nessa pesquisa, usa-se um modelo não-determinístico para controlar em alto nível a dramatização dos eventos, de modo a obter: duração variável dos eventos conforme a conveniência; dramatização variada, mesmo que o enredo seja o mesmo; e oportunidades de interação no nível de dramatização. O controle do tempo de duração dos eventos pode ser uma ajuda importante para garantir a continuidade do fluxo em situações críticas. Por isso, um dos primeiros trabalhos futuros a serem realizados é a obtenção de uma versão do Logtell que seja capaz de combinar os resultados dos dois trabalhos (CIARLINI *et al.*, 2008).

A necessidade de uso do IPG em tempo real faz com que a sua eficiência seja crucial para o funcionamento do Logtell. No contexto de exemplo, o IPG consegue, na maior parte das vezes, gerar capítulos em velocidade suficiente para não haver interrupção de fluxo. No entanto, em contextos mais complexos, há um risco maior de ocorrência de interrupções. Para evitar isso, pretende-se investir no desenvolvimento de uma nova versão do IPG. Essa versão deverá incorporar mecanismos de planejamento onde se mistura o conhecimento do domínio, sob a forma de redes hierárquicas de tarefas (HTNs), com planejamento independente de domínio. A idéia é ter um ganho de eficiência decorrente do uso de HTNs, mas sem perder a flexibilidade atual para a obtenção de soluções novas. Além da preocupação com a eficiência, deseja-se dar mais flexibilidade para a dramatização. Seria desejável que algum nível de não-determinismo fosse possível na geração dos enredos. Isso permitiria maior variação para a

dramatização e mais opções de interação com o usuário.

Uma questão fundamental para o sucesso de sistemas de *storytelling* interativo é o desenvolvimento de mecanismos que facilitem o esforço autoral de especificação dos contextos das histórias, em especial quando se deseja trabalhar com gêneros mais complexos do que contos de fadas. Para o Logtell, um esforço inicial já foi feito com a implementação de uma versão preliminar do *Context Control Module* (CCM), que facilita a especificação, através de uma interface gráfica, das operações, personagens, lugares e regras de inferência de objetivos de um contexto. Essa versão do CCM armazena os dados em um banco de dados e é capaz de gerar um arquivo de contexto no formato usado pelo IPG para gerar os enredos. Pretende-se aperfeiçoar o CCM para permitir o armazenamento unificado e sem redundâncias das informações referentes à lógica de geração de enredos, ao controle da dramatização e à animação gráfica. Além disso, o CCM deverá se comunicar diretamente com os outros módulos do sistema, facilitar o reaproveitamento de conteúdo e prover os meios para a especificação rápida e flexível dos contextos das histórias.

Novos mecanismos de dramatização deverão ser implementados. Planeja-se migrar a dramatização para um motor de jogos customizável, o que permitirá a obtenção de animações gráficas mais realistas. Além disso, vêm sendo estudados tópicos ligados à direção, tais como o posicionamento de câmera e o controle dos atores virtuais nas cenas.

Além das pesquisas atualmente em andamento, há diversas possibilidades de incorporação futura de recursos ao modelo de *storytelling* interativo do Logtell. O tratamento de emoções e *drives* (motivações) dos personagens tanto no nível de autoria dos enredos quanto de dramatização, o uso de música, a geração de textos para diálogos e a narração das histórias são possibilidades interessantes.

Por fim, deve-se ressaltar que boa parte das técnicas aqui citadas não se limita apenas a entretenimento. Podem ser utilizadas em simulações em geral, em particular com propósitos de educação, treinamento e suporte à decisão. A interação contínua, incorporada agora ao sistema, é de fundamental importância para que bons resultados sejam obtidos também com esses propósitos. Em muitos casos, a dramatização através da animação de atores virtuais pode ser substituída por outras formas de representação das “histórias”, tal como a composição automática de (partes) de vídeos.

## Referências Bibliográficas

AARNE, A., 1964. *The Types of the Folktale: A Classification and Bibliography*. Traduzido e editado por Stith Thompson, FF Communications, 184. Helsinki: Suomalainen Tiedeakatemia.

ALUR D., CRUPI & J., MALKS, D., 2001. *Core J2EE Patterns: Best Practices and Design Strategies*. Publisher: Prentice Hall / Sun Microsystems Press.

APPLE, 2009. "What is Apple TV?" Disponível em <<http://www.apple.com/appletv/whatis.html>>. Acesso em Janeiro de 2009.

BLUM, A. L. AND FURST, M. L., 1997. Fast Planning through planning graph analysis. *Artificial Intelligence*, v. 90, n. 1-2, pp 281-300.

BECKER, V., VARGAS, R., FILHO, G. H., MONTEZ, C., 2008. *Júri Virtual I2TV: Uma Aplicação para TV Digital Interativa baseada em JavaTV e HyperProp*. Disponível em <<http://www.tvdi.inf.br/index.php?s=artigos>>. Acesso em Dezembro de 2008.

CAMANHO, M. M. ; CIARLINI, A. E. M. ; FURTADO, A. L. ; POZZER, C.T. ; FEIJÓ, B., 2008. Conciliating Coherence and High Responsiveness in Interactive Storytelling. In: *Proceedings of the 3rd ACM International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008)*, Atenas.

CAVAZZA, M., CHARLES, F., MEAD, S., 2002. Character-based interactive storytelling. *IEEE Intelligent Systems, special issue on AI in Interactive Entertainment*, v. 17, n. 4, pp. 17-24.

CHARLES, F.; CAVAZZA, M., MEAD, S., 2001. *Character-driven story generation in interactive storytelling*. Relatório Técnico, VSMM, Berkeley.

CIARLINI, A. E. M. ; CAMANHO, M. M. ; DORIA, T. R. ; FURTADO, A. L. ; POZZER, C.T. ; FEIJÓ, B., 2008. Planning and Interaction Levels for TV Storytelling. *1st. Joint International Conference on Interactive Digital Storytelling*, Erfurt, Alemanha,.

CIARLINI, A.E.M., POZZER, C. T., FURTADO, A.L., FEIJO, B. A., 2005. Logic-Based Tool for Interactive Generation and Dramatization of Stories. In: *Proc. ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2005)*, Valencia.

CIARLINI, A.; FURTADO, A., 2002. Understanding and Simulating Narratives in the Context of Information Systems. In: *Proc. ER'2002 – 21st. International Conference on Conceptual Modelling*, Tampere, Finlândia, Out. 2002..

CIARLINI, A.; VELOSO, P. ; FURTADO, A., 2000. A formal framework for modelling at the behavioural level. In: *Proc. of the 10<sup>th</sup> European-Japanese Conference on Information Modelling and Knowledge Bases*, Saariselkä, Finlândia.

CIARLINI, A., 1999. *Geração interativa de enredos*. Tese de Doutorado, Departamento de Informática, PUC-Rio, Rio de Janeiro.

CPqD, 2005. *Política Regulatória: Panorama Brasileiro Atual – Projeto Sistema Brasileiro de Televisão Digital: Modelo de Implantação*. Versão PD.30.12.36A.0002A/RT-05-AA. Campinas, CPqD, 2005, (Relatório Técnico, Cliente: Funttel, atividade 1236, OS:40539).

CPqD, 2001. *Relatório Integrador dos Aspectos Técnicos e Mercadológicos da Televisão Digital*. Anatel. Março de 2001.

CPqD, 2008. Modelo de Referência – Sistema Brasileiro de Televisão Digital Terrestre. Versão PD.30.12.36A.0002A/RT-08-AB. 2006. Disponível em <<http://www.tvdi.inf.br/index.php?s=artigos>>. Acesso em Dezembro de 2008.

CRAWFORD, C., 1999. Assumptions underlying the Erasmatron storytelling system. In *Working Notes of the 1999 AAI Spring Symposium on Narrative Intelligence*. AAI Press.

CRAWFORD, C., 2005. *Chris Crawford on Interactive Storytelling*. Indianápolis, Estados Unidos: New Riders Games, 2005. 384 p. ISBN 0-321-27890-9

DAVIC, 2008. “DAVIC 1.4 Part 2 – DAVIC Specification Reference Models and Scenarios”, 1998. Disponível em <<http://www.davic.org>>. Acesso em Dezembro de

2008.

DOHERTY, P., KVARNSTRÖM, 2001. J. TALplanner: A temporal logic based planner. *AI Magazine*, v. 22, n. 3, pp. 95-102, 2001.

DORIA, T. R. ; CIARLINI, A. E. M. ; ANDREATTA, A., 2008. A Nondeterministic Model for Controlling the Dramatization of Interactive Stories. In: *Proceedings of the ACM MM2008 - 2nd ACM Workshop on Story Representation, Mechanism and Context - SRMC08*.

DRISCOLL, G., 2000. *The essential guide to digital set-top boxes and interactive TV*. Prentice Hall, Upper Saddle River, NJ, 1st edition, Novembro 2000.

EROL, K.; HENDLER, J.; NAU, D. S. UMCP, 1994.: A sound and complete procedure for hierachical task-network planning. In *Proceedings of the International Conference on AI Planning Systems (AIPS)*, pp. 249-254.

ETSI, 2003. “TS 102 812 V1.2.1: Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP) Specification 1.1.1”. ETSI Standard.

ETSI, 2005. “TS 102 819 V1.3.1: Digital Video Broadcasting (DVB) Globally Executable MHP version 1.0.2 (GEM 1.0.2)”. ETSI Standard.

FRANZ, M. L. V., 1980. *O significado psicológico dos motivos de redenção nos contos de fadas*. Editora Cultrix, São Paulo.

FURHT, B., 1996. Interactive television systems. In: *Proceedings of the 1996 ACM Symposium on Applied Computing*, pp. 7-11, Philadelphia, Pennsylvania, February 1996. ACM Press.

FURTADO, A.; CIARLINI, A., 2000. Generating narratives from plots using schema information. In: *Proceedings of the 5th International Conference on Applications of Natural Language to Information Systems-Revised Papers*, Versalhes, França, Junho 2000.

FURTADO, A.; CIARLINI, A., 2001. “Constructing Libraries of Typical Plans”. In *Proc. CAiSE’01, The Thirteenth International Conference on Computer Advanced Information System Engineering*, Interlaken, Suíça.

GEVERINI, A., SERINA, I., 2002. LPG: A planner based on local search for planning graphs. In *Proceedings of the International Conference on AI Planning Systems (AIPS)*, pp. 968-973.

GLASSNER, A., 2004. *Interactive Storytelling: Techniques for 21st Century Fiction*. AK Peters Ltd.

GRACIOSA H. M. M. G., 2008. TV Digital no Brasil. Disponível em <<http://www.teleco.com.br/tutoriais/tutorialtvd2/default.asp>>. Acesso em Dezembro de 2008.

GRASBON, D. and BRAUN, N., 2001. A morphological approach to interactive storytelling. In *Proc. CAST01, Living in Mixed Realities. Special issue of Netzspannung.org/journal, the Magazine for Media Production and Inter-media Research*, pp. 337-340, Sankt Augustin, Germany..

HOFFMAN, J., 2001. FF: The Fast-Forward planning system. *AI Magazine*, v. 22, n. 3, pp. 57-62.

HAVi, 2008. “*HAVi Level 2 Graphical User-Interface -Specification of the Home Audio/Video Interoperability (HAVi) Architecture*”. HAVi, Inc. Disponível em <<http://www.havi.org>>. Acesso em Dezembro de 2008.

IBGE, 2008. “*IBGE - Pesquisa Nacional por Amostra de Domicílios*”. Disponível em <[http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2007/tab\\_sintese.shtm](http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2007/tab_sintese.shtm)>. Acessado em Dezembro, 2008.

ISO, 1996. “*ISO/IEC 13818-1 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 1: Systems (MPEG-2 Systems)*”.

ITU, 2001. ITU Recommendation J.200:2001, Worldwide common core – Application environment for digital interactive television services.

ITU, 2003. ITU Recommendation J.202:2003, Harmonization of procedural content formats for interactive television applications.

ITU, 2004. ITU Recommendation J.201:2004, Harmonization of declarative content format for interactive television applications.

JOHNSON, R., 2004. *Expert One-on-One J2EE Development without EJB*. Wiley Publishing Inc., 2004.

KAUTZ, H. A., 1991. A Formal Theory of Plan Recognition and its Implementation. In: *Allen, J. F. et al (eds.): Reasoning about Plans*. Morgan Kaufmann, San Mateo..

KARLSSON, B.; POZZER, C. T.; CIARLINI, A. E. M.; FURTADO, A. L.; FEIJÓ, B., 2006. Improving the Scene: Extending LOGTELL to Support a Plan-recognition / Plan-generation Paradigm. In: *V Brazilian Symposium on Computer Games and Digital Entertainment..*

MATEAS, M., STERN, A., 2000. Towards integrating plot and character for interactive drama. In: *Working notes of the Socially Intelligent Agents: The Human in the Loop*, AAAI Fall Symposium. Technical report, pp. 113-118, Menlo Park, CA, 2000. AAAI Press.

MATEAS, M., STERN, A., 2004. A Behavior Language: Joint action and behavioral idioms. In: *Life-like Characters. Tools, Affective Functions and Applications*: Springer Verlag.

MATEAS, M., STERN, A., 2005. Structuring content in the Facade interactive drama architecture. In *Proc. Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2005)*.

MARRS, T. and DAVIS, S., 2005. *JBoss at Work: A Practical Guide*. O'Reilly Media, Inc.

MCKEE, Robert, 2006. *Story: Substância, Estrutura, Estilo e os princípios da escrita de roteiro*; tradução de Chico Marés. Ed. Arte e Letra, Curitiba, 2006. ISBN: 978-85-60499-00-7.

MONTEZ.C, BECKER.V., 2005. *TV DIGITAL INTERATIVA*; 2ª Edição. Florianópolis: Editora da UFSC.

MSN TV2, 2008. Disponível em <<http://www.webtv.com/pc>>. Acesso em Dezembro de 2008.

NAU, D. S., AU, T.-C., ILGHAMI, O., KUTER, U., MURDOCK, W., WU, D.,

YAMAN, F., 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, 20:379-404.

PAIVA, A., MACHADO, I., PRADA, R., 2001. Heroes, villains, magicians, ... Dramatis personae in a virtual story creation environment. In *Proc. Intelligent User Interfaces*.

POLTI, G., 1945. *Thirty-Six Dramatic Situations*. Whitefish, MT: Kessinger Publishing.

POZZER, C. T., FEIJO, B., CIARLINI, A. *et al.*, 2004. Managing Actions and Movements of Non-Player Characters in Computer Games. In *Proc. of the Brazilian Symposium on Computer Games and Digital Entertainment*.

POZZER, C.T., 2005. *Um Sistema para Geração, Interação e Visualização Tridimensional de Histórias para TV Interativa*. Tese de Doutorado, Departamento de Informática, PUC-Rio.

PRADA R., MACHADO I., PAIVA A., 2000. TEATRIX: Virtual Environment for Story Creation. In *Lecture Notes in Computer Science*. Vol. 1839/2000. pp. 464-473.

PROPP, V., 1968 *Morphology of the Folktale*, Laurence Scott (trans.), Austin: University of Texas Press.

RIEDL, M. and YOUNG, R. M., 2004. An intent-driven planner for multi-agent story generation. In the *Proceedings of the 3rd International Conference on Autonomous Agents and Multi Agent Systems*, Julho 2004.

RIEDL, M. and YOUNG, R. M., 2006. "From Linear Story Generation to Branching Story Graphs". *IEEE Computer Graphics and Applications*, v. 26, n. 3, pp. 23-31, May/June 2006, doi:10.1109/MCG.2006.56

RIEDL, M.O., STERN, A., DINI, D., ALDERMAN, J., 2008. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on Systems Science and Applications*, Special Issue on Agent Based Systems for Human Learning, v. 4, n. 2.

RODRIGUES, P. S. L.; FEIJÓ, B., VELHO, L.; POZZER, C. T.; CIARLINI, A. E. M.; FURTADO, A. L., 2006. Narrating Stories in Participatory Games. In: *V Brazilian Symposium on Computer Games and Digital Entertainment*.

- MARRIOT; STUCKEY, P., 1998. *Programming with Constraints*. MIT Press.
- MORRIS, S., SMITH-CHAIGNEAU, A., 2005. *Interactive TV Standards – A Guide to MHP, OCAP and JavaTV*. ISBN-13 978-0-240-80666-2. Elsevier, Focal Press..
- SANCRINI, M., 2005. O Uso da Televisão Digital no Contexto Educativo, *Educação Temática Digital*, Campinas, v. 7, n. 1, ISSN 1676-2592, pp. 31-44..
- SANTOS, D. T., SILVA, M.R.C, MELONI, L.G.P., 2005. Ferramentas de Apoio ao Ensino a Distância via TV Digital Interativa. In: *Taller Internacional de Software Educativo*, Santiago-Chile.
- SCHWALB, E. M., 2003. *iTV Handbook: Technologies and Standards*, Prentice Hall PTR, July 2003.
- SGOUROS, N.M., 1999. Dynamic generation, managing and resolution of interactive plots. *Artificial Intelligence*, v. 107, n. 1, pp. 29-62.
- SI, M., MARSELLA, S.C., RIEDL, M., 2008. Integrating Plot-Centric and Character-Centric Designs: An Mixed-Initiative Framework for Interactive Drama. *Proceedings of the 4th Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, Palo Alto, California.
- SI, M., MARSELLA, S.C., PYNADATH, D.V., 2005. THESPIAN: An Architecture for Interactive Pedagogical Drama. In: *International Conference on Artificial Intelligence in Education, Amsterdam*.
- SICSTUS, 2008. *Rebaseing JVM for SICStus*. Disponível em <[http://www.sics.se/~perm/sicstus/rebase\\_jvm.html](http://www.sics.se/~perm/sicstus/rebase_jvm.html)>. Acesso em Dezembro de 2008.
- SOARES, L. F. G., RODRIGUES, R. F., MORENO, M. F., 2007. GINGA-NCL: the Declarative Environment of the Brazilian Digital TV System. *Journal of the Brazilian Computer Society*, Revista n. 4; v. 12; Mar. 2007 - ISSN 0104-6500.
- SOUZA, G. L. F., L. E. C. LEITE, BATISTA, C. E. C. F., 2007. GINGA-J: The Procedural Middleware for the Brazilian Digital TV System. *Journal of the Brazilian Computer Society*. Revista n. 1; v. 13; Mar. 2007 - ISSN 0104-6500

SPIERLING, U.; BRAUN, N.; IURGEL, I. ; GRASBON, D., 2002. Setting the scene: playing digital director in interactive storytelling and creation. *Computers & Graphics*, v. 26, n.1, pp. 31-44, 2002.

SUN, 2008a. Sun Microsystems, Java TV API, Disponível em <<http://java.sun.com/products/javatv/>>. Acesso em Dezembro de 2008.

SUN, 2008b. Sun Microsystems. “Java Media Framework API (JMF)”. Disponível em <<http://java.sun.com/products/javamedia/jmf/index.jsp>>. Acesso em Dezembro de 2008.

SWEDLOW, T. 2000: *Interactive Enhanced Television: A Historical and Critical Perspective*. Disponível em <<http://www.itvt.com/etvwhitepaper.html>>. Acessado em Dezembro, 2008.

SZILAS, N., 2008. The mutiny: an interactive drama on IDtension . In: *Proceedings of the 3rd ACM International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008)*. New York, NY, USA : ACM, 2008. pp. 539-540.

SZILAS, N., 2003. IDtension: The simulation of Narrative. In *Proc. of the 3rd International Conference on Computational Semiotics for Games and New Media*, Middlesbrough (Royaume-Uni), pp. 10-12.

TANENBAUM, A., 2003. *Redes de Computadores*. 4a edição. Rio de Janeiro: Editora Campus.

TOME, T.; PESSOA, A.C.F.; RIOS, J.M.M.; LOURAL, C.A.; DALL'ANTONIA, J.C., 2001. *Relatório Integrador dos Aspectos Técnicos e Mercadológicos da Televisão Digital*. Versão AB PD.33.SV.E5A.0005A/RT-02-AB. Campinas: CPqD, 2001, 84p.

URSU, M. F., THOMAS, M., KEGEL, I., *et al.*, 2008. Interactive TV Narratives: Opportunities, Progress, and Challenges. In: *ACM Transactions on Multimedia Computing, Communications, and Applications*, v. 4, n. 4 (Outubro 2008). ISSN:1551-6857

YANG, Q., TENENBERG, J. and WOODS, S., 1996. On the Implementation and Evaluation of Abtweak. In: *Computational Intelligence Journal*, v. 12, n. 2, Blackwell Publishers (1996) pp. 295-318.

YOUNG, R., 2000. Creating interactive narrative structures: The potential for AI approaches. In: *AAAI Spring Symposium in Artificial Intelligence and Entertainment*, Palo Alto, California, 2000. AAAI Press.

YOUNG, R., 2001. An overview of the mimesis architecture: Integrating narrative control into a gaming environment. In: *Working notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, pp. 78-81, Stanford, CA, Março 2001. AAAI Press.

YOUNG, R.M., RIEDL, M.O., BRANLY, M., JHALA, A., MARTIN, R.J., SARETTO, C.J., 2004. An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments. *Journal of Game Development*, v. 1, pp. 51-70. 2004.

## Apêndice A – Arquiteturas e Padrões Para a TV Digital Interativa

### Transmissão de Dados

Uma das principais mudanças dentro do novo meio digital da iTV é o próprio canal de transmissão, fundamental para a construção da DTV, ou *Digital Television*. O modelo de referência para a DTV na maioria dos sistemas, se faz através do uso de algoritmos de compressão, em especial o MPEG-2 para imagem e MP3 pra áudio, reduzindo o uso da banda enquanto se preserva a qualidade.

O MPEG-2 tem uma série de particularidades, como o fato de ser um algoritmo assimétrico, tendo portanto um custo de codificação muito maior que o da decodificação, algo conveniente para a DTV. Além disso, é um algoritmo flexível, capaz de codificar imagens em diversos níveis de qualidade, sendo ainda escalonável, o que possibilita arranjos compostos de sinal de áudio e vídeo (x vídeos com y áudios). Por ser tão adaptável, ele também permite que até mesmo um decodificador de baixa capacidade de processamento, adapte as informações de acordo com seu potencial, sem perder a oportunidade de exibir o conteúdo recebido. É importante ressaltar que o MPEG-2 é apenas uma descrição genérica da forma de codificação e multiplexação dos sinais, sem impor especificações concretas na implementação.

É importante ressaltar que há uma grande diferença entre o padrão DTV e o conhecido por HDTV – *High Definition TV*. O consumo de banda da HDTV é muito maior, e só é possível atualmente através de transmissão digital. Sua resolução varia em tamanho, dependendo do padrão: 1080x1920 (USA), 1000x1778 (Europa) e 1080x1920 (Japão). Trata-se de resolução muito maior que a da TV convencional, também chamada de SDTV – *Standard Definition TV*: 484x720 (Padrão NTSC, USA) e 575x767 (Padrão PAL, Europa). Existem ainda padrões ainda mais impressionantes, devido à suas grande resoluções de imagem como o padrão japonês *Ultra High Definition Video* (UHDV) da NHK (*Japan Broadcasting Corporation*). De todo modo é relevante lembrar que quanto maior a qualidade da imagem e som, maior será a banda necessária, o que aumenta as exigências na infra-estrutura, ainda mais quando se deseja adicionar recursos de interatividade, por exemplo, na transmissão do sinal.

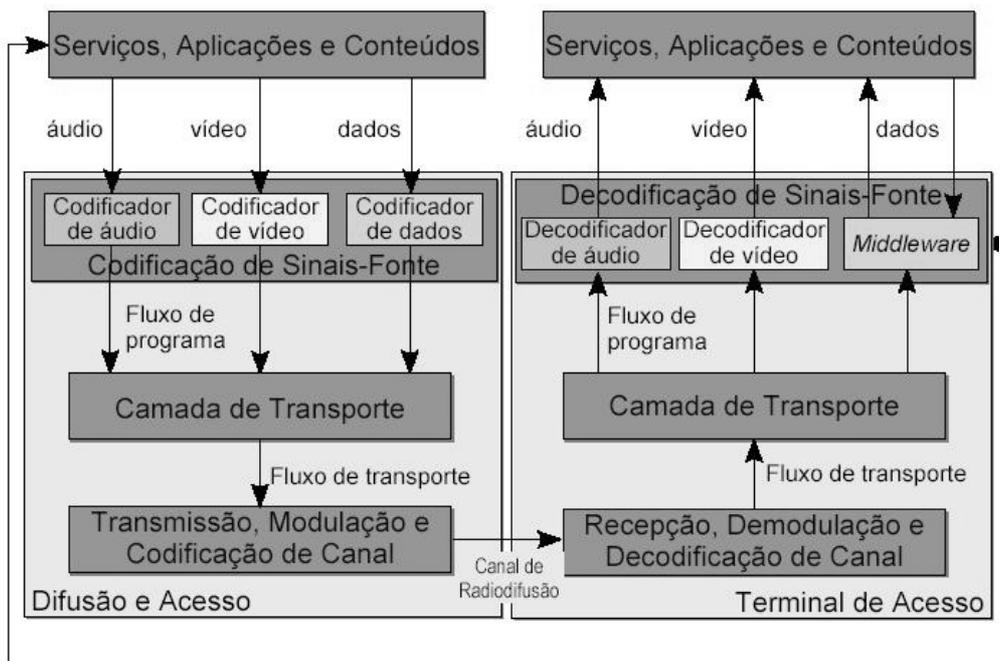


Figura A.1: Representação esquemática de TV Digital

Para existir interatividade em um ambiente de TV Digital, como esquematizado na Figura A.1, não é estritamente necessária a aquisição de novos televisores digitais. Até televisões analógicas podem ser utilizadas em conjunto com *set-top boxes* ou URDs - “Unidades Receptoras-decodificadoras” que contenham conversores de sinais digitais para sinais analógicos. *Set-top boxes* (SCHWALB, 2003) têm a capacidade de processamento de sinais de vídeo e áudio, além de poderem executar programas. São capazes de recepção, demodulação, decodificação e remodulação do sinal digital, gerando sinal de áudio e vídeo compatível com televisores analógicos, além de dar suporte às formas de interação denominadas "Pseudo" e "Real".

Na "pseudo" interatividade, o *set-top box* se comunica com a central de produções do canal desejado e processa os fluxos de dados multiplexados, sendo capaz de exibir na televisão uma interface para o usuário, que pode então interagir com o programa de TV através do controle remoto, ou teclado. A interatividade "real" é viabilizada através de uma conexão com um canal de retorno, geralmente por internet ou no caso de TV a cabo por exemplo, transmitindo pelo próprio meio por onde recebe o sinal de TV; e possibilita uma maior gama de oportunidades de interação. É possível a instalação dinâmica no *set-top box* de uma cópia de um sistema de arquivos produzido no estúdio de dados. No *set-top box* podem então ser exibidos textos transmitidos e recebidas aplicações para serem executadas.

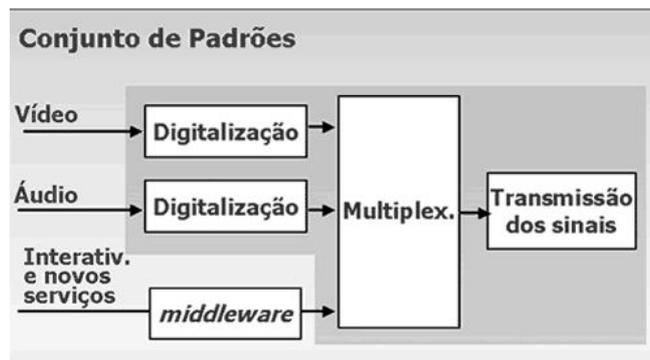


Figura A.2: Esquema Geral Simplificado de Padrões para TV Digital (adaptado de GRACIOSA, 2008)

No mundo, existem diferentes padrões de transmissão TV digital, como apresentado na Figura A.2, cada um com vantagens e desvantagens:

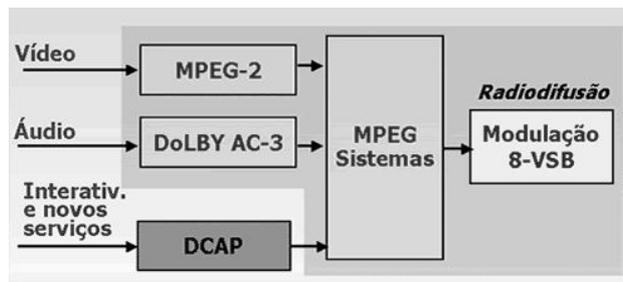


Figura A.3: Esquema Simplificado de Padrões do ATSC (adaptado de GRACIOSA, 2008)

- Americano – ATSC (*Advanced Television Systems Committee*) - É o mais antigo, representado na Figura A.3, começou a ser utilizado em 1998, e é focado na transmissão de vídeo em alta definição, o HDTV.

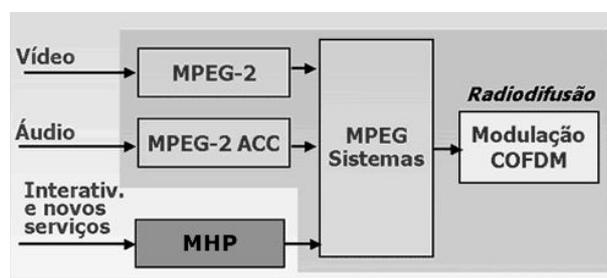


Figura A.4: Esquema Simplificado de Padrões do DVB (adaptado de GRACIOSA, 2008)

- Europeu – DVB (*Digital Video Broadcasting*) - Implantado em 1993, representado na Figura A.4, tem a possibilidade de multiprogramação, interatividade e novos serviços. Além disso, também privilegia a HDTV, e

também a recepção móvel e portátil.

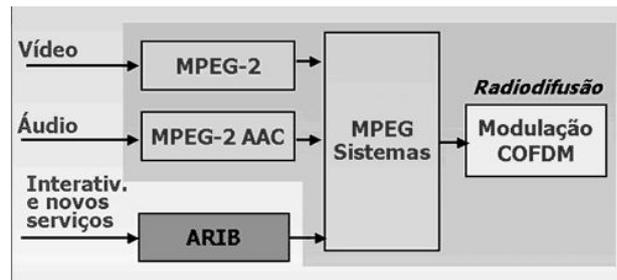


Figura A.5: Esquema Simplificado de Padrões do ISDB (adaptado de GRACIOSA, 2008)

- Japonês – ISDB (*Integrated Service Digital Broadcasting*) - Implantado em 1999, representado na Figura A.5, tem como principal vantagem a facilidade e qualidade na transmissão para variados dispositivos, como TV, celulares, etc, aliando mobilidade à alta definição. Privilegia multiprogramação, interatividade e novos serviços.

Além disso, existem as possibilidades de transmissão para a TV Digital através de cabo, satélite e por via terrestre. As características geográficas do Brasil privilegiam a transmissão por via terrestre. Por outro lado, a TV por assinatura obedece seus próprios padrões, que já utiliza transmissão digital há alguns anos, muito antes da definição do sistema de televisão digital brasileiro pelo governo.

A largura da banda disponível por banda terrestre é de 6MHz, o equivalente a 29,162 Mbps. Com o uso de MPEG-2, uma emissora de TV pode transmitir um único programa em qualidade de HDTV, ou então diferentes combinações, podendo usar a banda para transmitir 6 canais SDTV simultâneos, ou até outras opções (CPqD, 2001).



Figura A.6: Algumas opções de distribuição da largura de banda para TV digital

A Figura A.6 ilustra algumas das opções de modelos de negócio para a utilização da banda disponível. Como se evidencia pela quantidade de diferentes padrões, e pelos exemplos de distribuição de banda na figura, não há ainda consenso no que concerne à

melhor opção para uso da televisão Digital. Uma opção é priorizar uma melhor resolução, utilizando a banda toda para o a HDTV. Outra opção é simplesmente manter o padrão SD (lembrando que sua qualidade já é superior ao sinal analógico recebido por antenas convencionais) e acrescentar interatividade, o que permitiria integração com serviços interativos e internet. Pode-se considerar ainda a opção de variar a resolução da imagem de acordo com o conteúdo, aproveitando a flexibilidade do meio. De todo modo, é de se esperar que se reserve sempre uma parte da banda para interatividade, de acordo com todo o seu potencial. Para dar suporte à comunicação entre o espectador e o conteúdo que está sendo gerado de forma interativa, serão necessárias linhas de comunicação rápidas e confiáveis, ainda mais num contexto de vídeo sobre demanda.

No Brasil, o sistema de transmissão digital foi definido após longo processo de comparação com os outros sistemas existentes (o americano, o japonês e o europeu) e já está em implantação, em caráter livre e gratuito. Decidiu-se por um sistema baseado no modelo japonês (*ISDB-T: Integrated Services Digital Broadcasting – Terrestrial*), por ser mais novo e mais facilmente adaptável. Dentre as principais facilidades incluídas nele, estão:

- Modulação COFDM-BST (*Coded Orthogonal Frequency Division Multiplexing - Band Segmented Transmission*). O BST divide o canal em treze segmentos e cada segmento pode levar um conteúdo/programa diferente. O segmento do meio, sétimo, é usado para transmitir para os celulares e equipamentos portáteis, e de forma gratuita.
- Possibilidade de transmitir mais de um programa no mesmo canal; por exemplo, um programa em alta definição e outro para o celular, podendo assim transmitir o mesmo programa de diferentes formas.
- Possibilidade de incorporar novas tecnologias tal como o uso de MPEG4 no lugar do MPEG2, usado pelos outros sistemas.
- Possibilidade do uso de *middleware* para prover interatividade – o Ginga, criação nacional desenvolvida pela academia, através da PUC-RIO e da Universidade Federal da Paraíba.
- Possibilidade de criar, no mesmo município, uma rede de transmissores na mesma frequência para cobrir áreas de sombra - onde a imagem não pode ser vista - e permitir que toda a população possa ver os programas de todas as emissoras.

O sistema final definido para Brasil foi o ISDB-TB (ISDB-T do Brasil), também denominado SBTVD (Sistema Brasileiro de Televisão Digital) (CPqD, 2006). Ele incorporou atualizações ao modelo japonês nas partes de áudio, vídeo e interatividade e traz uma série de vantagens para a realidade brasileira, como, por exemplo, a redução no pagamento de *royalties* e o preço do *set-top box* mais acessível, o que possibilita uma maior inclusão digital. Ao utilizar *set-top boxes* para a conversão do sinal digital para analógico, o parque instalado é aproveitado, permitindo uma implantação gradual, que leve em conta as diferentes condições sócio-econômicas do Brasil. Com o uso do *set-top box* ligado à internet, é possível ainda utilizar a televisão como um *browser* (contudo, ainda que normalmente mais limitado que o de um PC).

Alguns projetos têm servido como modelo para a TV Digital. O Projeto I2TV (Infra-estrutura Internet2 para Desenvolvimento e Teste de Programas e Ferramentas para TV Interativa), desenvolvido no Brasil (BECKER *et al.*, 2008) por um conjunto de entidades que integram universidades (UFSC, USP, PUC-Rio, UFPB e UFRN) e emissoras de TV de seus estados, usou a internet para a transmissão de conteúdo digital. Nesse projeto, foram estudadas e analisadas camadas de hardware, de *middleware*, de exibição e de geração de conteúdo, sob uma abordagem que busca integrar *broadcast* de TV Digital com a Internet. Esse projeto serviu para demonstrar a possibilidade da criação de protótipos funcionais para o ambiente desejado.

### ***Middleware***

Para implementar aplicativos para televisão digital existem duas abordagens principais: o uso de aplicações declarativas ou de aplicações procedurais. Define-se por aplicação declarativa aquela cujo conteúdo é projetado de forma declarativa, enquanto que, nas aplicações procedurais, seu conteúdo é projetado de maneira procedural. Esta diferença de padrões define a própria filosofia e utilidade para cada aplicação feita.

Ao projetar um conteúdo declarativo, deve-se então utilizar uma linguagem declarativa, ou seja, uma linguagem que se baseie em modelos lógicos declarativos, em vez de algoritmos seqüenciais, mais comuns à maioria das linguagens de programação. Por outro lado, quando um conteúdo é procedural, deve-se usar uma linguagem não declarativa. Podem ser adotadas diferentes abordagens ao se utilizar linguagens não declarativas, incluindo entre elas a programação orientada a objetos, o desenvolvimento baseado em componentes, e outros padrões de implementação e projeto. Porém, na bibliografia comum sobre televisão digital, o termo “procedural” é usado para

representar todas as linguagens não declarativas. Na programação procedural, utiliza-se uma abordagem baseada em chamadas a procedimentos, também conhecidos como funções, métodos etc., que são utilizados em algoritmos seqüenciais estruturados, o que dá, de certa forma, um controle maior sobre o comportamento do código, e sobre o controle fluxo de execução dos programas. Dentre as linguagens procedurais, incluem-se a maioria, como C, C++, Java, etc., sendo que, usualmente, a linguagem procedural mais utilizada na área, dada a popularidade e grande quantidade de projetos e iniciativas da Sun Microsystems, tais como a JavaTV (SUN, 2008), é o Java.

No Brasil, através de uma pesquisa conjunta de diversas instituições foi criado o *middleware* Ginga (Figura A.7), para a implementação de aplicativos interativos para a TV Digital. Em concordância e com a filosofia do Java, o Ginga busca o desenvolvimento independente da plataforma de hardware dos diferentes *set-top boxes*. O Ginga é aplicável aos receptores para sistemas de transmissão terrestre, e busca cumprir completamente uma série de especificações para serem usadas em aparelhos de TV Digital integrados, computadores multimídia e *clusters* locais de aparelhos em redes domésticas (HAN). O núcleo do Ginga, a Ginga-Core (Figura A.8) é composta por diversos decodificadores de conteúdos comuns, como imagens JPG, PNG, etc. e procedimentos para obter conteúdos transportados em canais (*Streams*) de transporte do MPEG-2 e através do canal de retorno.

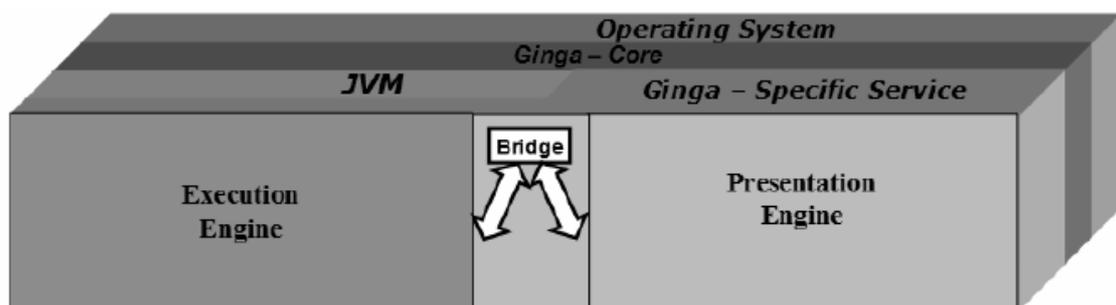


Figura A.7: Arquitetura do Ginga, compatível com (ITU, 2001).

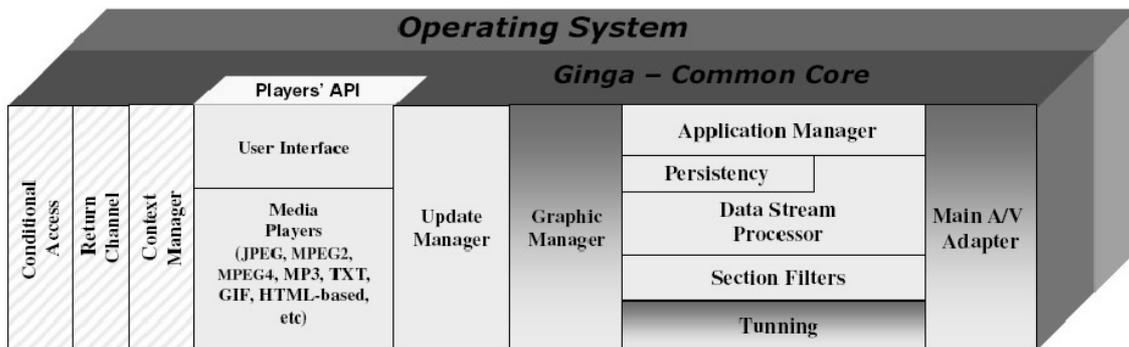


Figura A.8: Componentes da Ginga Common Core.

O padrão Ginga é compatível com várias definições internacionais ITU (ITU 2001, 2003, 2004), e suporta tanto aplicativos declarativos quanto procedurais. É possível ainda a criação de aplicativos em que as duas formas de conteúdo existem simultaneamente e uma referencia a outra.

O Ginga-NCL (SOARES *et al.*, 2007) é a porção declarativa do Ginga. Ele provê uma estrutura para a apresentação de aplicações declarativas escritas na linguagem NCL. Através de um decodificador, também conhecido como *NCL formatter*, o conteúdo pode ser processado. Outros módulos importantes do Ginga-NCL incluem o agente de usuário, baseado em XHTML, que inclui folhas de estilo (CSS) e um interpretador ECMAScript, além da *engine* LUA, capaz de interpretar *scripts* LUA, muito utilizados em jogos e outros produtos interativos. Além disso, é possível o uso de outros padrões declarativos através de implementações diferentes baseadas em XHTML.

O aspecto procedural do Ginga é implementado através do Ginga-J (SOUZA *et al.*, 2007), o qual inclui diferentes APIs baseadas em padrões JavaTV. O Ginga-J é projetado para ser capaz de suprir as funcionalidades necessárias para a criação de aplicações para a TV Digital, com funcionalidades para protocolos de acesso, interação, manipulação de objetos multimídia, etc. O Ginga-J utiliza aplicativos procedurais, também conhecidos como Xlets Java. Como toda implementação Java, utiliza uma máquina virtual Java (JVM), que estabelece uma camada de abstração sobre o sistema operacional subjacente, o que facilita a compatibilidade e portabilidade das diferentes implementações em diferentes hardwares, dado que se siga corretamente a sua especificação. Uma aplicação Ginga-J deve-se basear nas definições GEM 1.1 (ETSI, 2005) (*Globally Executable MHP*), que são especificações unificadas para a criação de *niddlewares* para TV Digital. Uma aplicação compatível com este padrão será, em princípio, compatível com o Ginga-J.

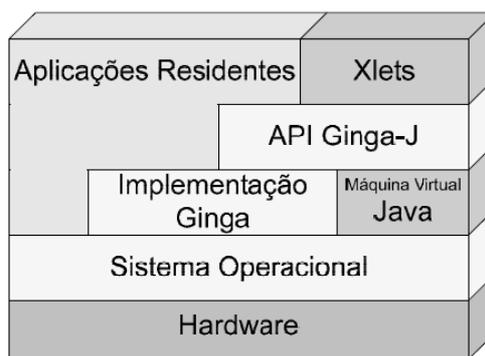


Figura A.9: Arquitetura do Ginga-J

O Ginga se baseia em 3 conjuntos de APIs chamadas *Green*, *Yellow* e *Blue*. As APIs *Green* são compatíveis com as especificações GEM. As APIs *Yellow* são extensões feitas para alcançar requisitos específicos do sistema brasileiro, e que podem ser implementadas através de um software adaptador usando as APIs *Green*. As APIs *Blue* são as que não são compatíveis com as API GEM, ou seja, só podem ser executadas nos ambientes *middleware* Ginga.

A API *Green* do Ginga-J é composta pelas APIs Sun JavaTV (SUN, 2008a), DAVIC (DAVIC, 2008), HAVi (HAVI, 2008) e DVB (MORRIS, 2005; ETSI, 2003), todas incluídas na especificação do *framework* GEM. As APIs *Yellow* do Ginga-J são compostas pela API JMF 2.1 (SUN, 2008b), que é necessária para o desenvolvimento de aplicações avançadas, com captura de som por exemplo, e outras funcionalidades ligadas ao processamento multimídia. Trata-se de uma extensão à API de apresentação do GEM, com funcionalidades para suporte de *stream* de vídeo no padrão Ginga-J, uma extensão à API do canal de retorno presente no GEM, que permite o envio de mensagens assíncronas, e uma extensão à API de *Service Information* do padrão B.23 do ISDB ARIB. O conjunto de APIs *Blue* do Ginga-J contém uma API de integração de dispositivos, que permite a comunicação do receptor de TV Digital com qualquer dispositivo de entrada ou saída que use uma interface compatível (como *Bluetooth*, infra-vermelho, rede *Ethernet*, etc.). Contém também uma API multiusuário que usa a API de integração de dispositivos para permitir que vários usuários possam interagir simultaneamente com aplicativos de televisão digital. Além disso, contém uma API para fazer uma ponte com o NCL, para a criação de aplicativos Java que contenham aplicativos NCL.