

6 Arquitetura do Sistema

Nos capítulos anteriores são apresentados diversos aspectos relacionados com a geração das histórias (conteúdo, geração, níveis de interatividade, diversidade), que têm como apoio lógico o IPG (*Interactive Plot Generator* [33]). Para permitir que o usuário possa interativamente guiar e visualizar as histórias sendo geradas, neste capítulo faz-se a proposição de um sistema que integra os processos de geração, interação e visualização de histórias. Este sistema tem como intuito permitir que o usuário possa interagir com o conteúdo tanto na geração como na visualização das histórias, por meio de interfaces gráficas.

Como um dos objetivos desta tese é a proposição de conteúdo baseado em histórias interativas para ser usado em um ambiente de TV interativa, o sistema proposto contempla características que permitem fazer uso da tecnologia corrente de hardware e software para iTV como forma de disponibilizar histórias interativas personalizadas a cada usuário.

6.1 Arquitetura Geral

A Figura 6.1 apresenta um diagrama geral do sistema de histórias interativas em iTV. Ele é baseado em uma arquitetura distribuída cliente-servidor, onde o servidor age somente como o provedor de conteúdos (base de dados), enquanto o cliente age como receptor, gerador e visualizador do conteúdo interativo.

No servidor são definidas duas bases de dados, onde são armazenados conteúdos necessários tanto para a geração como para a visualização dos enredos. Na base Prolog são especificados os contextos lógicos dos enredos a serem disponibilizados aos usuários como forma de conteúdo interativo. Cada enredo é armazenado em um único arquivo texto, que contém os predicados (personagens, lugares, atributos), operações e regras

de inferência dos objetivos a serem perseguidos pelos personagens, como pode ser visto no Apêndice A.

Cada enredo está associado a elementos gráficos que são usados para transformar as representações simbólicas das operações em ações interpretadas por personagens habitando um mundo virtual 3D. Esses elementos gráficos são armazenados em uma base gráfica, que contém os modelos 3D de personagens e objetos do cenário, bem como texturas e definições do próprio ambiente, por meio da geometria do terreno.

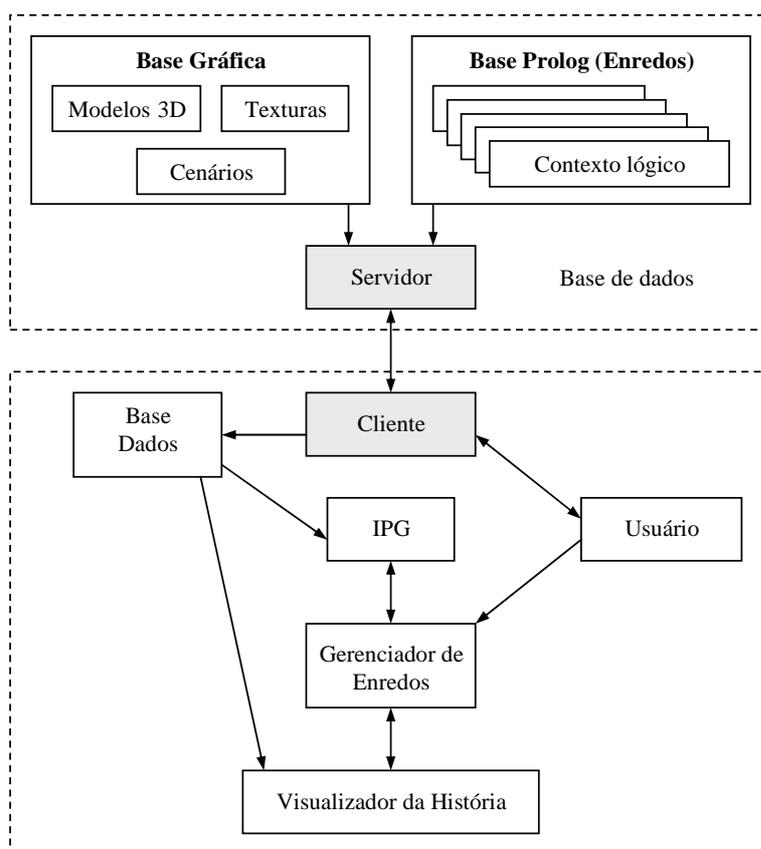


Figura 6.1: Estrutura para disponibilização de histórias interativas em iTV.

A Figura 6.2 apresenta um modelo de implantação do sistema dentro de um ambiente de iTV. No nível mais baixo encontra-se o provedor de conteúdos e no nível mais alto, o usuário. O Set-top Box está localizado no centro e é o intermediador de todo o processo. Ele realiza todo o processamento computacional referente à solicitação, ao recebimento e armazenamento de conteúdo interativo, bem como a geração e exibição gráfica das histórias, mediante interação com o usuário.

A arquitetura apresentada na Figura 6.1 permite que o usuário possa selecionar o contexto de histórias com o qual deseja interagir. Por meio de uma interface com o cliente, ele pode saber quais contextos estão disponíveis

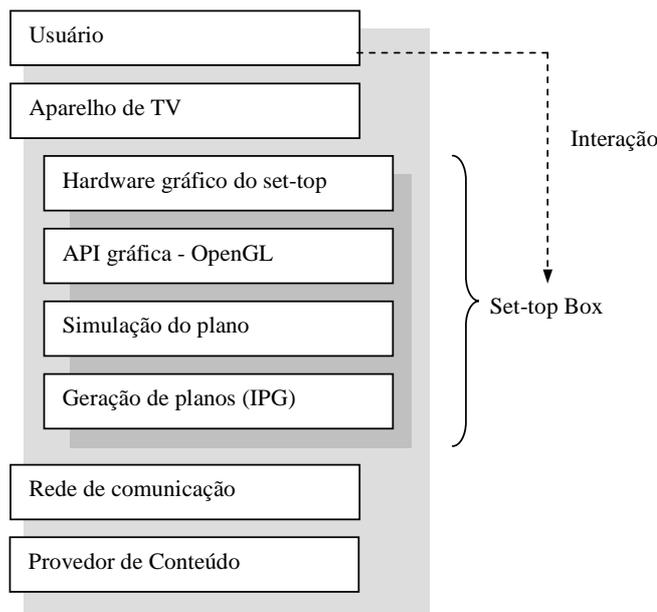


Figura 6.2: Modelo de implantação do sistema em um contexto real de iTV.

na base do provedor e selecionar o que mais lhe interessar. Uma vez selecionado o contexto, os dados referentes a ele são enviados pelo provedor ao cliente e então armazenados em uma base local. O **Gerenciador de Enredos** é o componente centralizador, que coordena tanto a geração como a visualização.

A colocação do IPG no lado do cliente tem dois objetivos: permitir que cada usuário tenha sua história personalizada e garantir maior eficiência, visto que a geração de planos é um processo computacionalmente caro. Deste modo, cada usuário faz uso de processamento local do set-top box, o que assegura menor tempo de resposta, bem como evita comunicações freqüentes com o provedor de conteúdos.

O uso de banda passante, geralmente um problema em diversos tipos de conteúdos interativos, não apresenta maiores impactos, visto que a comunicação com o servidor ocorre apenas num momento inicial, quando o usuário seleciona uma história a interagir. Neste momento, dados referentes à história, bem como modelos gráficos, devem ser levados ao cliente e então armazenados na base local para futura utilização. Durante todo o resto do processo de interação com a história, todo o processamento é realizado localmente. Antes de realizar qualquer requisição de dados ao provedor, pode-se fazer uma consulta ao cache local do set-top box, para verificar se o dado requisitado, ou parte dele, já se encontra disponível localmente.

Como mencionado no Capítulo 2, tanto o provedor de conteúdos como a camada de rede são abstraídos com o uso de um único computador, que

desempenha o papel de provedor e de set-top Box. A Figura 6.3 apresenta os módulos da arquitetura que atualmente estão implementados. A base de dados local já é inicializada com todos os dados necessários. Para o usuário, a interação ocorre de forma transparente. Nesta figura, o componente de visualização da história está apresentado de modo mais detalhado.

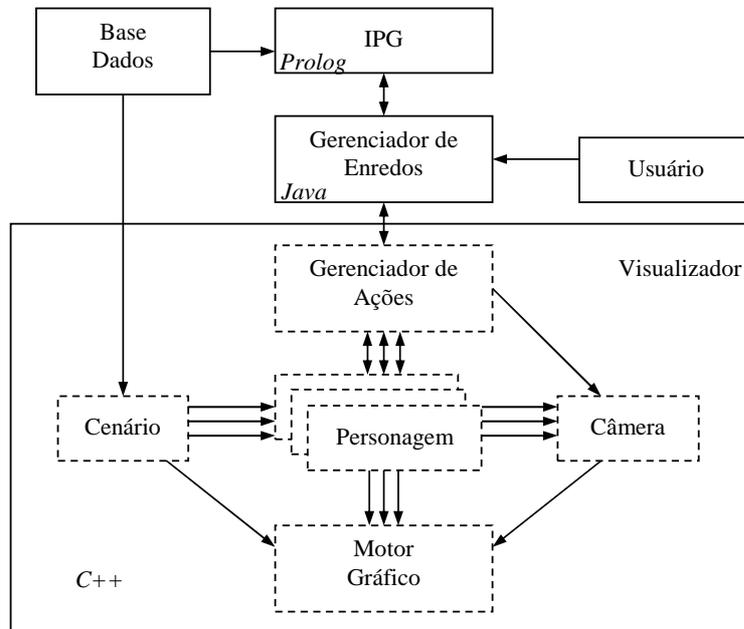


Figura 6.3: Módulos implementados do sistema. O módulo de visualização é apresentado em maiores detalhes.

O **Gerenciador de Ações** tem a função de processar os eventos fornecidos pelo Gerenciador de Enredos, transformando-as em ações que são então delegadas aos respectivos personagens. Ele também realiza todo o sincronismo de visualização das ações de personagens para a representação dos eventos. Toda vez que um evento é processado, um novo é requisitado ao Gerenciador de Enredos.

6.2

Os Módulos do Sistema

Os três grandes módulos do sistema (IPG, Gerenciador de Enredos e Visualizador) foram implementados usando as linguagens mais adequadas para cada tipo de tarefa, como apresentado na Figura 6.4. A linguagem JAVA utilizada no Gerenciador de Enredos é utilizada como base para a comunicação entre os módulos. Cada módulo faz uso de linguagens/bibliotecas multiplataformas, um requisito fundamental para

viabilizar a implantação desta tecnologia nas diversas arquiteturas de set-top boxes e sistemas operacionais.

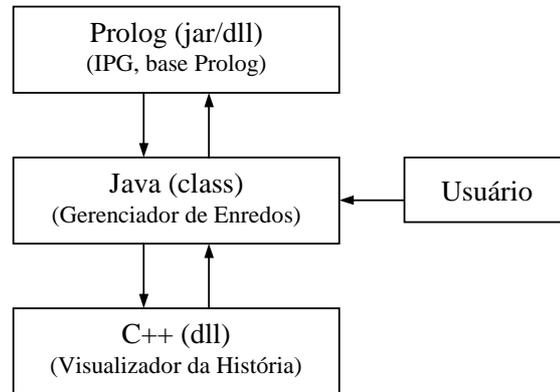


Figura 6.4: Linguagens usadas para implementar os módulos.

6.2.1

O Gerenciador de Enredos

Gerenciador de Enredos age como centralizador e integrador da arquitetura, servindo como ponte de comunicação entre o IPG (implementado em Prolog) e o **Visualizador de Histórias** (implementado em C++). Totalmente implementado em Java, ele concentra todos os recursos de interação com o usuário. A funcionalidade do Gerenciador de Enredos engloba os seguintes itens:

- Interação com o IPG para guiar a geração da história, segundo comandos do usuário. Pode requisitar ao IPG novas alternativas de enredos parciais, a continuação de um enredo, ou a adição de novos eventos e ordens temporais a um enredo que está sendo processado;
- Interface com o usuário para o gerenciamento das operações geradas pelo IPG, ou inseridas pelo próprio usuário. Por meio de uma interface gráfica, o usuário pode requisitar a renderização gráfica, fazer a inserção/remoção de eventos e forçar a ocorrência de situações específicas no enredo, bem como definir a ordem total dos eventos;
- Interação com o módulo gráfico para gerenciar a visualização da história. Gerencia também o envio de eventos a serem representados, o que pode ser resultado de uma ação explícita do usuário, ou por uma requisição do módulo gráfico.

6.2.2 O IPG

Para executar o IPG, faz-se uso do SICStus Prolog [125], um interpretador Prolog multiplataforma¹ desenvolvido pelo Instituto de Computação da Suécia (*Swedish Institute of Computer Science*). O SICStus é executado em código nativo, em formato de biblioteca de ligação dinâmica, podendo assim ser incorporado a outros aplicativos.

O SICStus disponibiliza uma interface em Java chamada Jasper que permite que uma aplicação Java possa acessar o SICStus de forma transparente (Isso é realizado via chamadas JNI - *Java Native Interface* [84]).

Sobre o Jasper, implementou-se em Java o **Componente de Consulta**. Ele tem a função de interagir com o Jasper no intuito de converter os dados gerados pelo IPG em um formato que possa ser facilmente interpretado pelo Gerenciador de Enredos (em Java). O **Componente de Consulta** também tem como função fazer a inicialização do IPG, com a história a ser processada, bem como fornecer acesso simplificado a eventuais consultas feitas sobre a base Prolog.

Na Figura 6.5 é apresentada a hierarquia de componentes que permite a comunicação entre o Gerenciador de Enredos e o IPG.

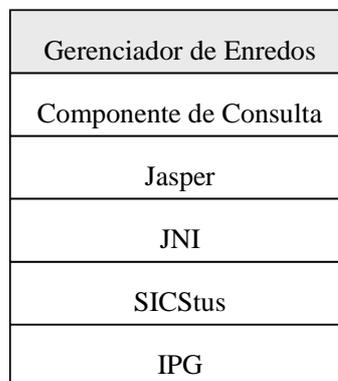


Figura 6.5: Camadas entre o Gerenciador de Enredos e o IPG.

¹Atualmente, as seguintes plataformas e sistemas operacionais são disponíveis: Intel/x86 (Windows 2000/XP, Linux, Solaris 7), Compaq/Alpha (Tru64 Unix 5.1), HP9000/7XX (HP-UX 11.x), IBM RS/6000(AIX 4.3.3, AIX 5L 5.1 (32/64 bits)), Silicon Graphics (IRIX 6.5), Sun Sparc (Solaris), Macintosh (MacOS).

6.2.3 O Visualizador da História

O módulo Visualizador da História, implementado em C++, cuida de todas as funcionalidades necessárias à visualização gráfica das histórias. Como mostrado na Figura 6.3, ele incorpora o **Gerenciador de Ações**, o cenário, os personagens, a câmera e o motor gráfico. O módulo faz uso tanto de processos de Inteligência Artificial como de Computação Gráfica.

Quando o usuário requisita a visualização da história, o **Visualizador da História** recebe do Gerenciador de Enredos um conjunto de eventos a serem representados graficamente. Cada operação está associada a um evento, que por sua vez está associado a personagens. O Gerenciador de Ações faz o gerenciamento de todos os personagens. Ele tem como função avaliar cada evento e invocar os respectivos personagens a realizar o evento. Os personagens, implementados como agentes reativos, devem realizar as ações associadas e enviar seu estado geométrico (representado pela malha de polígonos corrente) ao motor gráfico para ser então desenhado na tela. A câmera virtual tem como papel exibir a ação que está sendo processada pelo personagem encarregado de representar a operação corrente.

A Figura 6.6 ilustra as camadas para a implementação do Visualizador de Histórias e a comunicação deste com o Gerenciador de Enredos. O produto final do Visualizador são malhas poligonais que representam a geometria da cena no dado momento. Estas malhas são renderizadas pela biblioteca OpenGL [147]. Todos os componentes deste módulo são definidos em C++ e encapsulados em uma dll, que é carregada pelo Gerenciador de Enredos. Para que este módulo fale com o Gerenciador de Enredos, disponibilizam-se, via interface JNI, um conjunto de funções.

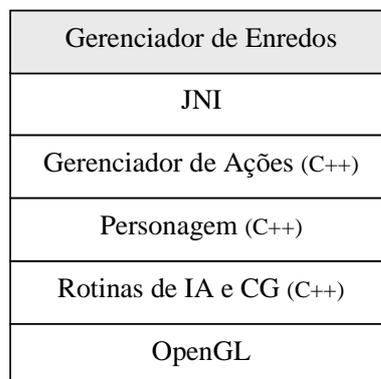


Figura 6.6: Hierarquia em camadas do módulo C++.

6.3 Conclusões e Observações

Por fazer uso de diversas tecnologias de software, o sistema utiliza três linguagens diferentes de programação, bem como uso de várias outras bibliotecas. Procurou-se fazer escolhas que permitissem o seu uso em múltiplas plataformas.

A integração de toda esta estrutura em um único sistema é uma tarefa complexa que exigiu longo período de estudos, planejamento e implementação. A linguagem Java, por disponibilizar uma API para comunicação com aplicações escritas em código nativo (JNI), é o elemento centralizador, que também inclui interface com o usuário, que é o meio por onde o usuário controla tanto a geração, organização e visualização das histórias.

Neste processo de integração, o módulo que apresentou os maiores desafios foi a integração com o IPG, não somente sob o aspecto de linguagem de programação, mas principalmente sob o aspecto de formas de interação com o usuário. No que se refere à implementação, as versões mais novas do SICStus Prolog disponibilizam APIs tanto em C como em Java, permitindo assim que as aplicações conversem diretamente entre si. Mesmo com a existência desta API, foi necessário desenvolver mais uma camada em Java, que além de traduzir mensagens trocadas entre os dois módulos, também incorpora recursos que permitem a sincronização da história que está sendo processada pelo IPG com a história sendo manipulada pelo usuário na interface gráfica. A integração dos módulos Java e C++ é mais simples pois ambos os módulos compartilham dados muito similares, uma vez que Java e C são linguagem procedimentais e de sintaxe muito semelhantes.

No próximo capítulo apresentam-se detalhes da implementação dos módulos, como ocorre a comunicação entre eles, bem como detalhes das bases de dados usadas para a geração e a visualização das histórias.