

Componente de Interação Multimodal no Ginga

Diogo Pedrosa, José Augusto C. Martins
Jr., Maria da Graça C. Pimentel
Universidade de São Paulo
São Carlos-SP
Brazil
{diogo,joseacm,mgp}@icmc.usp.br

Erick Melo
Universidade Federal de São Carlos
São Carlos-SP
Brazil
erick_melo@dc.ufscar.br

ABSTRACT

Currently, in most existing DTV applications, the user interaction takes place by pressing keys on a remote control. For simple applications this type of interaction is sufficient, however, as interactive applications become more popular new input devices will be desired. This paper specifies an architecture that offers to Ginga applications the possibility of receiving multimodal data (audio, video, image, ink, accelerometer data, text, voice and customized data) from multiple devices (such as mobile phones, PDAs, netbooks, tablet PCs, notebooks, and desktops) and presents the Multimodal Interaction Component developed.

RESUMO

Atualmente, na grande maioria das aplicações de TVD existentes, a interação dos usuários se dá através do pressionamento de teclas de um controle remoto. Para aplicações simples esse tipo de interação é suficiente, contudo, à medida que as aplicações interativas se tornem mais populares novos dispositivos de entrada serão desejados. Este trabalho especifica uma arquitetura que oferece a aplicações Ginga a possibilidade de receber dados multimodais de entrada (tinta eletrônica, dados de acelerômetros, texto, voz, etc.) provenientes de múltiplos dispositivos (tais como, celulares, PDAs, netbooks, tablet PCs, notebooks e desktops) e apresenta o Componente de Interação Multimodal desenvolvido.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures—
Data abstraction, Domain-specific architectures

General Terms

Design, Documentation, Human Factors

Keywords

Entrada Multimodal, TV Digital Interativa

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WebMedia'10, October 5–8, 2010, Belo Horizonte, MG, Brazil
Copyright 2010 ACM X-XXXXX-XX-X/XX/XX ...\$10.00.



Figure 1: Exemplo de controle remoto para TVDi

1. INTRODUÇÃO

O Brasil passa atualmente pela fase de implantação de seu sistema de televisão digital interativa (TVDi). Em diversas cidades a transmissão digital já está sendo realizada¹ e diversos modelos de decodificadores² já podem ser encontrados no mercado.

A recepção pode ser feita por dispositivos fixos, móveis ou portáteis, que contam com *hardware* e *software* necessários para o processamento do sinal digital recebido. Esses dispositivos podem possuir capacidade de execução de aplicações interativas, que são executadas sobre uma camada de software conhecida como *middleware*. O *middleware* é responsável por permitir uma abstração do *hardware* e sistema operacional presentes no receptor através do fornecimento de uma interface de programação (API). O *middleware* do Sistema Brasileiro de TV Digital (SBTVD) é o Ginga, que possui um ambiente de execução de aplicações imperativas (Ginga-J) e um ambiente de execução de aplicações declarativo (Ginga-NCL).

Atualmente, na grande maioria das aplicações de TVD existentes, a interação dos usuários se dá através do pressionamento de teclas de um controle remoto que dispõe ao menos das teclas direcionais, OK, voltar, coloridas e numéricas (Fig 1). Para aplicações que apenas apresentam conteúdos adicionais e permitem uma navegação por esse conteúdo, esse tipo de dispositivo de interação é suficiente. Contudo, à medida em que aplicações interativas se tornem mais populares, novos dispositivos e periféricos de entrada serão desejados – como microfones, câmeras, telas sensíveis ao toque, canetas eletrônicas e acelerômetros (Fig. 2).

São exemplos de aplicações para TVDi o guia eletrônico de programação ou EPG (*Electronic Program Guide*), aplicações

¹O site <http://www.forumsbtvd.org.br/materias.asp?id=55> lista as cidades brasileiras onde a TV digital está no ar. Acessado em 30/junho/2010.

²Também conhecidos como receptor, terminal de acesso ou *set-top box* (STB)



Figure 2: Exemplos de dispositivos e periféricos que podem ser usados para entrada de dados

relacionadas com um programa de TV ou com cobertura de grandes eventos (olimpíadas e carnaval, por exemplo), jogos e aplicações de notícias [7, 10]. Essas aplicações oferecem poucas funcionalidades pois precisam ser fáceis de usar, mesmo tendo apenas o controle remoto como dispositivo de entrada. A restrição ao controle remoto, em comparação com um teclado completo e um dispositivo de manipulação direta como o mouse, comuns em computadores pessoais, limita a usabilidade de aplicações de TVDi [1]. Segundo Roibás et al. [14], interfaces que confiam toda navegação a um controle remoto convencional estão longe de expressar o potencial interativo de um sistema de TDVi. O desenvolvimento de aplicações mais ricas pode demandar interfaces complexas que limitariam sua popularização [6].

Este trabalho visa apresentar a arquitetura de um componente que permita que aplicações mais ricas possam ser construídas para TVDi, oferecendo a elas a possibilidade de recebimento de dados multimodais provenientes de diferentes dispositivos.

O restante deste documento está organizado da seguinte forma: a próxima seção apresenta a motivação e os cenários de uso para o componente; a Seção 3 apresenta trabalhos relacionados ao tema; a Seção 4 apresenta parte do Componente de Interação Multimodal (CIM) desenvolvido no contexto do projeto Ginga FrEvo e Ginga RAP [18], detalhando cada um dos módulos envolvidos e dando ênfase ao Gerenciador de Eventos, que desempenha um papel central na arquitetura; a Seção 5 apresenta uma aplicação teste que roda no Ginga e uma aplicação de transmissão que roda em um dispositivo externo, ambas desenvolvidas para permitir uma validação inicial do CIM durante seu desenvolvimento; e a Seção 6 apresenta nossas considerações finais.

2. MOTIVAÇÃO E CENÁRIOS DE USO

César e Chorianopoulos [5] consideram que um passo evolutivo está sendo dado com a substituição do modelo tradicional de produção-distribuição-consumo pelo modelo mais centrado no usuário, de criação-compartilhamento-controle. A criação de conteúdo, antes restrita a empresas de radiodifusão e produtoras, vem sendo cada vez mais realizada pelos usuários através de ferramentas de autoria para computadores pessoais, como Flash³, Director⁴, iMovie⁵ e Movie Maker.⁶ Porém, elas não são ferramentas voltadas para o paradigma da televisão.

³<http://www.adobe.com/products/flash>

⁴<http://www.adobe.com/products/director>

⁵<http://www.apple.com/ilife/imovie>

⁶<http://download.live.com/moviemaker>

Trabalhos como os apresentados por César et al. [3] e Catelan et al. [2] vêm sendo realizados no sentido de permitir que os usuários criem ou enriqueçam conteúdos multimídia de forma voltada para a televisão. Nesses trabalhos, o conteúdo base é oriundo da televisão e o resultado da autoria pode ser visto na televisão, mas a autoria em si é realizada em aplicações que executam em um Tablet PC. A interação com o Tablet PC possibilita a realização de anotações mais elaboradas, como as anotações feitas com tinta eletrônica e apresentadas em exemplos nesses trabalhos. Já no trabalho de Pimentel et al. [11], outra ferramenta para autoria de conteúdo multimídia é apresentada, mas desta vez a interação se dá através do próprio controle remoto da TV. Nele, as possibilidades de anotação são bem menores, ficando restritas a marcar cenas como favoritas, selecionar trechos do vídeo para serem pulados durante a reprodução em uma nova oportunidade e selecionar trechos do vídeo para serem reproduzidos ciclicamente. Percebe-se a necessidade de explorar os pontos fortes desses dois tipos de produção de conteúdo: o uso de dispositivos que possibilitem entradas mais enriquecedoras para aplicações e o benefício de fácil acesso ao conteúdo e do confortável ambiente para exibição trazido pela execução das aplicações diretamente nos decodificadores.

Programas educacionais, filmes, novelas e eventos esportivos são cenários que podem usufruir de mecanismos de entrada mais ricos que um controle remoto tradicional. Explorando o hábito de fazer comentários sobre um conteúdo televisivo, Teixeira et al. [20] apresentam um cenário em que o usuário realiza anotações de tinta sobre cenas de uma partida de futebol para indicar para um amigo a ocorrência de impedimento em um determinado lance. Algo semelhante pode ser imaginado no contexto de uma novela ou para diversos outros programas. Uma aplicação genérica para permitir anotações de tinta eletrônica sobre vídeo só seria possível através de mecanismos de entrada de dados multimodais.

Um requisito importante para realização de autoria de conteúdo multimídia por parte dos usuários é a possibilidade de escrever de textos, tarefa difícil de ser realizada com o controle remoto. O uso de outros dispositivos de entrada surge como uma possível solução para aplicações com essa necessidade. Exemplos de aplicações simples que requerem entrada de texto são aplicações de notícias que permitem comentários por parte dos usuários (como ocorre hoje na grande maioria dos sites de notícias e blogs), EPGs que permitem busca por palavra-chave na grade de programação, aplicações de comércio eletrônico que necessitam de preenchimento do nome do comprador e local de entrega e aplicações de comunicação, que permitem envio de mensagens entre telespectadores, e-mails, SMSs, ou participação em bate-papo [9]. Um exemplo de aplicação que permite busca por palavra-chave é apresentado no trabalho de Wittenburg et al. [21]. Os usuários pronunciam uma palavra utilizando um microfone como dispositivo de entrada e recebem como saída uma lista de programas possivelmente relacionados. Já na categoria de comunicação, um exemplo é o ChatTV⁷. A popularidade desse canal de bate-papo da NET⁸ mostra que num momento de descontração na frente da TV as pessoas sentem necessidade de se comunicar, mesmo que para isso

⁷<http://www.chatv.com.br>

⁸<http://nettv.globo.com>

tenham que utilizar um celular para mandar um SMS e pagar a taxa cobrada de R\$ 0,31 + impostos por mensagem, como é o caso nesse serviço. Nesse caso, o celular já está sendo utilizado de forma indireta como um dispositivo de entrada para uma aplicação de TV Digital.

Ao oferecer fácil integração com dispositivos de entrada externos, o decodificador acaba contribuindo para aumentar a acessibilidade de aplicações de TV Digital. Usuários portadores de necessidades especiais que já utilizam dispositivos específicos que os ajudam a superar suas limitações passam a poder utilizá-los também durante a interação com aplicações interativas de TV. A busca por oferecer a grupos minoritários as mesmas possibilidades de utilização de sistemas por meio de formas alternativas de interação, visando a democratização do acesso a informação e entretenimento, vem sendo realizada em diversas pesquisas, incluindo na área da TV Digital Interativa [1, 12, 14, 19]. Miranda et al. [6], preocupados especificamente com a necessidade criação de um novo dispositivo físico que permita interação com sistemas de TVDI, sugerem o uso da voz para atender a usuários portadores de deficiências motoras e visuais e analfabetos. Os autores ponderam, porém, que nem todos os ambientes permitem o uso de voz e que é necessário um treinamento para reconhecer a voz do usuário.

3. TRABALHOS RELACIONADOS

Considerando a televisão como um mero dispositivo de saída visual, Robertson et al. [13] apresentaram um sistema em que uma aplicação executando em um PDA podia fazer uso da televisão como uma tela secundária, e assim mostrar mais informações visuais para o usuário. César et al. [4] exploram a ideia da tela secundária na direção inversa. Em seu trabalho, os autores propõem 3 categorias de uso de um PDA que atua como uma tela secundária com capacidades de apresentação e interação: (1) permite ao usuário navegar pelos conteúdos e selecionar um determinado conteúdo para ser apresentado na televisão; (2) permite que o usuário continue assistindo, utilizando a tela do PDA, o conteúdo sendo exibido na TV de sua sala de estar, enquanto deixa o ambiente; (3) permite que o usuário faça recomendações de conteúdo para amigos e familiares, enviando trechos do conteúdo acompanhados de anotações ou conteúdos extras.

Seguindo na mesma linha, mas voltados para o contexto da televisão digital brasileira, Soares et al. [17] apresentam o modelo de controle hierárquico usado pelo Ginga-NCL para suporte a múltiplos dispositivos de exibição. Os autores argumentam que os produtores de conteúdo não desejam ter seu conteúdo sobreposto por informações adicionais e sugerem que a interface gráfica das aplicações seja exibida na tela do dispositivo secundário. Outro argumento dado no mesmo sentido é o de que o conteúdo sobreposto normalmente incomoda outros telespectadores que estão assistindo ao programa pelo mesmo aparelho. Uma aplicação de exemplo que faz uso dessa funcionalidade é apresentada. Nela, o usuário realiza a compra de uma chuteira utilizando a interface gráfica mostrada num dispositivo secundário, enquanto assiste à exibição de uma animação sobre Garrincha na tela da TV.

Os cenários apresentados por esses trabalhos, apesar de fazerem uso de dispositivos secundários, exploram principalmente a tela adicional que os dispositivos oferecem. Ao contrário da arquitetura aqui apresentada, a objetivo principal deles não é oferecer às aplicações de TV Digital a possi-

bilidade de receberem dados multimodais. Os cenários apresentados por César et al. [4] e Soares et al. [17] permitem que o dispositivo secundário ofereça entrada para a aplicação da televisão, mas não possibilitam a combinação de dois ou mais modos de entrada nem oferecem todos os tipos de dados cobertos pelo CIM.

Trabalhos que envolvem o uso de múltiplos dispositivos no ambiente imperativo do *middleware* brasileiro são apresentados por Silva et al. [15][16]. Esses trabalhos apresentam uma API em Java, exclusiva para aplicações do sistema brasileiro, que permitem que as aplicações utilizem recursos multimídia oferecidos pelos dispositivos, como câmera para captura de vídeo e foto, microfone para captura de áudio, rede telefônica, tela, e auto-falantes. Apesar de permitir a entrada de dados multimodais, a API apresentada possui duas grandes diferenças em relação ao que é oferecido pelo componente descrito neste trabalho: (i) A iniciativa para a transmissão de dados parte de lados diferentes. Com a entrada de dados via controle remoto e com a entrada de dados multimodais via CIM, o dado é enviado para o *middleware*. Já no caso dos trabalhos de Silva et al. é o *middleware* que busca os dados do dispositivo. Como consequência, usando o CIM, a aplicação está o tempo todo pronta para receber dados, enquanto que com a API Java de utilização de recursos, a aplicação tem que ter conhecimento da presença do dispositivo antes de poder receber qualquer dado proveniente dele. (ii) O CIM não limita os tipos de dados que podem ser recebidos. Além do áudio, vídeo e imagem já suportados pelo trabalho de Silva et al., o CIM prevê dados de tinta eletrônica, acelerômetros, voz ou puramente textuais, e ainda permite entrada de dados genéricos para casos não previstos.

4. ARQUITETURA DO COMPONENTE DE INTERAÇÃO MULTIMODAL

Para permitir que aplicações tenham a possibilidade de receber dados multimodais de diferentes dispositivos sem a necessidade de estabelecer explicitamente uma conexão com os mesmos, é desejado o uso de protocolos para configuração automática e descoberta de serviços em redes IP, como o *Zero Configuration Networking* (ZeroConf⁹) e o *Universal Plug and Play* (UPnP¹⁰).

A Figura 3 ilustra a arquitetura e o funcionamento do CIM. Os **Dispositivos** enviam os dados em um formato XML (explicado na Seção 4.1) para o componente. Esses dados são recebidos pelos **Módulos de Comunicação** (Seção 4.2) e repassados para o **Gerenciador de Eventos** (Seção 4.3), que faz uso de um *parser* para transformar o documento XML recebido num objeto do tipo **IMultimodalInputEvent**. Em seguida, o **Gerenciador de Eventos** notifica as aplicações que registraram interesse em receber esses dados.

A conversão dos dados recebidos em um evento multimodal simplifica o desenvolvimento de aplicações, pois permite que ela faça uso do mecanismo já bastante familiar de recebimento de eventos através de observadores e, principalmente, oferece uma forma padronizada de acesso aos dados. A localização do CIM no núcleo comum do Ginga permite que ele seja usado tanto pelo ambiente imperativo (Ginga-J) quanto pelo declarativo (Ginga-NCL), ou seja, os dados podem ser

⁹<http://www.zeroconf.org/>

¹⁰<http://www.upnp.org/>

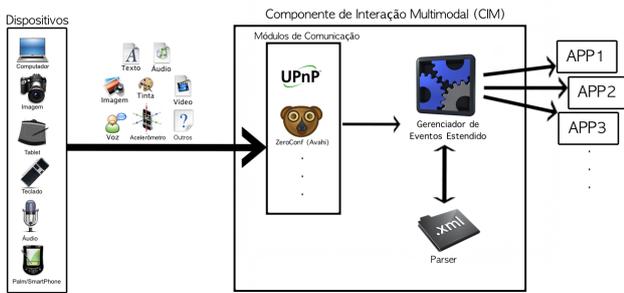


Figure 3: Arquitetura do CIM

acessados por métodos Java, no caso de *xlets*, ou por tabelas com campos pré-determinados, no caso de aplicações NCLua.

Este trabalho aborda apenas o caso em que um módulo de comunicação recebe os dados de entrada por meio de redes de computadores, porém a arquitetura não limita o recebimento de dados a esse meio. Podem ser considerados também casos em que periféricos são conectados diretamente ao decodificador, como, por exemplo, via interfaces USB.

O Componente de Interação Multimodal foi desenvolvido utilizando a implementação de referência do Ginga¹¹ disponível como uma máquina virtual.

4.1 Eventos multimodais

Do mesmo modo que eventos de pressionamento de teclas são representados pela interface `IInputEvent` já existente, eventos multimodais são representados pela interface `IMultimodalInputEvent` criada. Um único evento multimodal pode transportar todos os tipos de dados previstos ao mesmo tempo e é composto por:

- **id** – String que identifica o evento.
- **strings** – Vetor de strings, útil para que textos já prontos nos dispositivos possam ser enviados para aplicações.
- **ink** – Objeto de uma classe que guarda dados de tinta eletrônica, resultantes da interação do usuário com um dispositivo sensível ao toque ou munido de uma caneta eletrônica, por exemplo.
- **accel** – Objeto de uma classe que guarda dados resultantes da interação com um acelerômetro.
- **binaries** – Vetor de objetos que guardam dados (nome, mimetype e conteúdo) de arquivos binários, tais como arquivos de áudio, vídeo e imagem em qualquer formato.
- **voice** – Objeto de uma classe que guarda dados representando fala, que podem ser resultantes de sistemas de reconhecimento de fala ou servirem de entrada para sistemas sintetizadores de voz.
- **valuesMap** – Uma tabela de dispersão que pode transportar algum tipo de dado diferente dos anteriores. Possibilita maior flexibilidade por permitir que aplicações façam uso de tipos não previstos na API. Outro

¹¹<http://www.softwarepublico.gov.br/>

possível uso é transportar mais informações sobre dados do evento, como por exemplo um evento multimodal que transporte uma imagem jpg no vetor binários e os pares ("largura", 720) e ("altura", 480) na tabela.

Um protocolo em formato XML foi definido para permitir que os dispositivos enviem dados multimodais para o Ginga. O protocolo possui um cabeçalho contendo a identificação do dispositivo e usuário que geraram o evento e o instante inicial e final da geração. Os dados em si aparecem no corpo do evento. No trecho de documento XML apresentado a seguir é mostrado um exemplo de evento multimodal.

```
<?xml version="1.0"?>
<multimodal ... id="idDoEventoDeTeste">
  <head>
    <device id="DEADBEEF-DEAF-BABA-FEED-BABE00000006"
           model="iPhone 3GS"/>
    <user id="59616261-6461-6261-4E50-472050325033"/>
    <timestamp begin="2010-05-19T09:30:10.5"
              end="2010-05-19T09:30:17.8"/>
  </head>
  <body>
    <value id="largura">40</value>
    <value id="altura">30</value>
    <text>Foto da praia</text>
    <text>Que lugar lindo!</text>
    <inkml:ink>
      <inkml:trace>10 0,9 14,7 42,6 56,8 84,8 98,8 112
    </inkml:trace>
      <inkml:trace>130 155, 158 160, 170 154, 179 129
    </inkml:trace>
    </inkml:ink>
    <accel xValue="3" yValue="-2" zValue="1"/>
    <voice>aqui entra um voicexml</voice>
    <binary filename="praia.jpg" mimetype="image/jpeg">
      PD94bWwgdMvYc21...</binary>
  </body>
</multimodal>
```

Optou-se por utilizar o InkML¹² para representar os dados de tinta eletrônica, por tratar-se de um padrão W3C. O *parser* desenvolvido utiliza código da biblioteca `inkML-Libcpp`¹³, que foi adaptado para utilizar a biblioteca `Xerces`,¹⁴ já utilizada em outras partes do Ginga, em vez da `TinyXml`,¹⁵ utilizada originalmente pela `inkMLLibcpp`. Pretende-se também utilizar `VoiceXML`,¹⁶ porém a atual implementação do componente ainda não oferece suporte para dados de voz.

4.2 Módulos de comunicação

Como já comentado, a comunicação entre os dispositivos e o decodificador deve ser realizada através de protocolos para configuração automática e descoberta de serviços em redes IP, como o UPnP e o ZeroConf. A atual versão do CIM dá suporte ao ZeroConf, com auxílio da biblioteca `Avahi`,¹⁷ e, ainda, um módulo de comunicação que dê suporte ao UPnP está em implementação para que assim o CIM possa oferecer maior flexibilidade aos dispositivos que desejem fornecer dados para aplicações de TV digital.

¹²<http://www.w3.org/2002/mmi/ink>

¹³<http://sourceforge.net/apps/trac/inkmltk/wiki/InkMLLib>

¹⁴<http://xerces.apache.org/xerces-c/>

¹⁵<http://www.grinninglizard.com/tinyxml/>

¹⁶<http://www.w3.org/TR/voicexml21/>

¹⁷<http://avahi.org/>

Para enviar dados multimodais, uma aplicação que executa no dispositivo deve buscar o serviço ZeroConf oferecido pelo módulo de comunicação e se conectar a um socket através do endereço IP obtido do serviço. Várias conexões podem ser abertas ao mesmo tempo.

Os módulos de comunicação executam em *threads* separadas, que são iniciadas pelo construtor do gerenciador de eventos e só terminam sua execução quando o Ginga é encerrado. Garante-se assim que o decodificador estará sempre apto a receber eventos multimodais.

4.3 Gerenciador de Eventos

O Gerenciador de Eventos é o principal módulo do componente. Trata-se de uma extensão do gerenciador de eventos já existente na implementação de referência do Ginga. Ele foi implementando de forma a assumir o papel do gerenciador atual e ficar responsável por receber os dados multimodais provenientes dos serviços de comunicação e encapsulá-los em eventos que serão despachados para as aplicações interessadas.

Devido à forma componentizada com que a implementação de referência do Ginga tem sido desenvolvida, a integração do Gerenciador de Eventos estendido com o restante do Ginga se deu de forma fácil. Bastou editar o arquivo de configuração do Ginga Common Core Component Manager, indicando que o novo Gerenciador de Eventos, o **EnhancedInputManager**, deveria ser usado no lugar do **InputManager** presente no componente Ginga Common Core System.

Uma de suas responsabilidades é manter atualizada uma nova lista de observadores (*listeners*), específica para eventos multimodais, permitindo que aplicações interessadas possam se registrar ou se remover da lista. Todas as vezes que dados de entrada forem gerados por algum dispositivo e repassados para o Gerenciador de Eventos através do serviço de comunicação, o gerenciador cria um objeto que representa um evento multimodal de entrada, preenche-o com os dados recebidos, percorre a lista de observadores que registraram interesse e notifica-os da ocorrência do evento passando o objeto criado.

5. DISCUSSÃO

Para permitir uma validação inicial do CIM, duas aplicações foram desenvolvidas. A primeira é uma aplicação de linha de comando, desenvolvida em C, que pode ser executada em qualquer dispositivo localizado na mesma rede do decodificador e que conte com o suporte ao protocolo ZeroConf. Seu objetivo é gerar eventos multimodais para o Ginga. Assim que é iniciada, ela busca o serviço ZeroConf de recebimento de eventos disponibilizado pelo Módulo de Comunicação. Quando encontra, ela entra num laço em que envia a cada pressionamento da tecla ENTER o conteúdo de um xml de exemplo localizado no próprio diretório, bastante semelhante ao apresentado na Seção 4.1.

A segunda aplicação foi desenvolvida em C++ como uma aplicação residente que é iniciada pela função principal do Ginga. Seu objetivo é receber eventos multimodais e mostrar seu conteúdo através de mensagens de registro (*log*) em um terminal. As aplicações multimodais devem implementar a interface **IMultimodalInputEventListener**, registrar um observador de eventos multimodais através do método **addMultimodalInputEventListener** do **EnhancedInputManager** e tratar os eventos recebidos dentro do método **userMultimodalEventReceived**. Além de mostrar todos os

dados do evento em mensagens de registro, essa aplicação de teste também desenha os traces contidos na tag `ink` do evento, como mostrado na Figura 4.

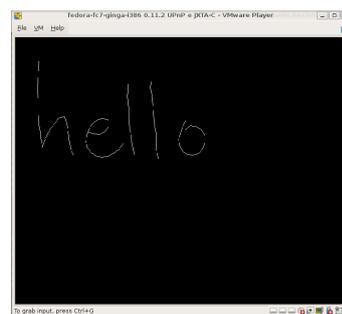


Figure 4: Saída da aplicação de teste do CIM

O componente implementado atualmente permite o recebimento de eventos multimodais por parte de aplicações residentes em C++. Para permitir um uso mais amplo do componente, APIs que permitam o recebimento dos eventos por parte de aplicações Java e Lua serão implementadas.

Depois dessa etapa, a implementação de uma aplicação mais robusta poderá ser implementada. Pretende-se estender o protótipo da aplicação de edição de documentos interativos apresentada por Pimentel et al. [11] para que a mesma passe a contar também com entradas de dados multimídia, como as oferecidas pelo Componente de Interação Multimodal. Atualmente, existe uma separação clara entre o tempo de autoria, realizada por autores especializados, e o tempo de consumo, realizado por telespectadores. Com o CIM, o próprio usuário final pode atuar como desenvolvedor, criando sua própria aplicação durante a interação com o conteúdo multimídia e utilizando bem mais recursos que o controle remoto.

Para flexibilizar ainda mais o componente, o protocolo será estendido para permitir o uso de formatos diferentes para o mesmo tipo de dado. Assim, caso se deseje utilizar um outro padrão de representação dos dados de tinta, por exemplo, seria necessário adicionar uma extensão no CIM que permita a interpretação desse novo formato.

Em relação ao Gerenciador de Eventos, pretende-se acrescentar 3 novas funcionalidades: 1) Extrair dados adicionais sobre o conteúdo recebido pelo protocolo e incluir esses metadados no evento multimodal criado, para assim oferecer informações mais completas às aplicações. Exemplos de metadados são altura e largura de uma imagem. 2) Permitir que aplicações se registrem para tipos de eventos específicos em vez de eventos multimodais de forma geral. 3) Realizar a captura de todas as interações ocorridas, ou seja, armazenar de forma persistente cada um dos eventos, para permitir a autoria automática de documentos multimídia correspondentes ou a extração de conhecimento relativamente às ações dos usuários.

6. CONSIDERAÇÕES FINAIS

O componente apresentado amplia o leque de opções dos desenvolvedores Ginga, possibilitando o desenvolvimento de aplicações muito mais inovadoras, já que quebra uma das principais restrições do desenvolvimento de aplicação para a televisão, que é contar apenas com entradas provenientes do controle remoto. Um dos tipos de aplicações que mais

poderá se beneficiar das funcionalidades introduzidas com o componente é o que de alguma forma permite criação de conteúdos por parte do usuário final. Aplicações dependentes de entrada de texto também serão bastante beneficiadas. No que se refere ao domínio da aplicação, as contribuições também são abrangentes. Um dos pontos que também merece ser reforçado é que possibilidade de integração do decodificador com dispositivos de entrada externos acaba contribuindo para aumentar a acessibilidade de aplicações de TV Digital.

Durante a definição da arquitetura e especificação do protocolo, procurou-se adotar padrões já consagrados, como o InkML, VoiceXML, ZeroConf e UPnP, além de exigir o mínimo de alterações possíveis nos demais componentes do Ginga. A arquitetura componentizada do Ginga facilitou atingir essa segunda meta.

A carência de suporte a aplicações que utilizem interação multimodal foi a grande motivação para a realização desse trabalho. Embora a norma do Ginga não preveja interação multimodal, o bom desempenho do componente em implementações de aplicações mais robustas pode resultar em uma proposta de alteração das normas no futuro.

Agradecimentos: RNP/MCT, CAPES, CNPq, FAPESP e FINEP.

7. REFERENCES

- [1] A. Carmichael, M. Rice, D. Sloan, and P. Gregor. Digital switchover or digital divide: a prognosis for usable and accessible interactive digital television in the uk. *Univers. Access Inf. Soc.*, 4(4):400–416, 2006.
- [2] R. G. Cattelan, C. Teixeira, R. Goularte, and M. D. G. C. Pimentel. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM TOMCCAP*, 4(4):1–24, 2008.
- [3] P. César, D. Bulterman, and A. Jansen. An Architecture for End-User TV Content Enrichment. *Journal of Virtual Reality and Broadcasting*, 3(9), Dec. 2006. urn:nbn:de:0009-6-7594.
- [4] P. César, D. C. Bulterman, and A. J. Jansen. Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content. In *EuroITV '2008*, pages 168–177, 2008.
- [5] P. César and K. Chorianopoulos. Interactivity and user participation in the television lifecycle: creating, sharing, and controlling content. In *UXTV '2008*, pages 125–128, 2008.
- [6] L. C. de Miranda, L. S. G. Piccolo, and M. C. C. Baranauskas. Artefatos físicos de interação com a tvdi: desafios e diretrizes para o cenário brasileiro. In *Brazilian IHC 2008*, pages 60–69, 2008.
- [7] S. Morris and A. Smith-Chaigneau. *Interactive TV Standards*. Focal Press, 2005.
- [8] L. S. Oliveira, M. da Graça Pimentel, and J. P. Queiroz-Neto. Work in progress - new challenges to educate in isolated areas in the amazon. In *FIE'09: Proc. IEEE Intl. Conf. Frontiers In education*, pages 1229–1230, 2009.
- [9] D. Pedrosa, D. A. Vega, M. d. G. C. P. Pimentel, and R. P. M. Fortes. Text input in digital television: a component prototype. In *EuroITV '2010*, pages 75–78, 2010.
- [10] C. Peng. *Digital Television Applications*. PhD thesis, Helsinki University of Technology, November 2002.
- [11] M. d. G. C. Pimentel, R. G. Cattelan, E. L. Melo, A. F. Prado, and C. A. C. Teixeira. End-user live editing of itv programmes. *Int. J. Adv. Media Commun.*, 4(1):78–103, 2010.
- [12] M. Rice and N. Alm. Designing new interfaces for digital interactive television usable by older adults. *Comput. Entertain.*, 6(1):1–20, 2008.
- [13] S. Robertson, C. Wharton, C. Ashworth, and M. Franzke. Dual device user interface design: Pdas and interactive television. In *ACM CHI '1996*, pages 79–86, 1996.
- [14] A. Roibás, R. Sala, S. Ahmad, and M. Radman. Beyond the remote control: Going the extra mile to enhance itv access via mobile devices & humanizing navigation experience for those with special needs. In *EuroITV '2005*, pages 133–141, 2005.
- [15] L. D. N. Silva, C. E. C. F. Batista, L. E. C. Leite, and G. L. Souza Filho. Suporte para desenvolvimento de aplicações multiusuário e multidispositivo para tv digital com ginga. *T&C Amazônia*, 12:75–84, 2007.
- [16] L. D. N. Silva, T. A. Tavares, and G. L. Souza. Desenvolvimento de programas de tvdi explorando as funções inovadoras do ginga-j. In *WebMedia '2008*, pages 26–29, 2008.
- [17] L. F. G. Soares, R. M. Costa, M. F. Moreno, and M. F. Moreno. Multiple exhibition devices in DTV systems. In *ACM MM'2009*, pages 281–290, 2009.
- [18] L. F. G. Soares, G. L. Souza Filho, C. A. C. Teixeira, C. S. Soares Neto, T. V. Batista, J. Gonçalves, S. D. J. Barbosa, M. K. Zuffo, M. B. d. Oliveira, F. d. C. Gomes, V. Roesler, L. M. G. Gonçalves, A. M. F. Burlamaqui, J. B. d. Santos Júnior, C. A. S. Santos, L. V. Agostini, F. M. Costa, M. d. G. C. Pimentel, and M. Cristo. GingaFrEvo & GingaRAP evolução do middleware Ginga para múltiplas plataformas (componentização) & ferramentas para desenvolvimento e distribuição de aplicações declarativas. Projeto CITIC/MCT, 2008.
- [19] M. V. Springett and R. N. Griffiths. Innovation for inclusive design: an approach to exploring the idtv design space. In *UXTV '08: Proceeding of the 1st international conference on Designing interactive user experiences for TV and video*, pages 49–58, 2008.
- [20] C. A. C. Teixeira, E. L. Melo, R. G. Cattelan, and M. d. G. C. Pimentel. User-media interaction with interactive tv. In *ACM SAC '2009*, pages 1829–1833, 2009.
- [21] K. Wittenburg, T. Lanning, D. Schwenke, H. Shubin, and A. Vetro. The prospects for unrestricted speech input for tv content search. In *In AVI'06*, pages 352–359, 2006.