

Plataforma Extensível para Interfaces Multimodais

Tiago Guerreiro¹, Vitor Fernandes², Bruno Araújo³, Joaquim Jorge⁴, João Pereira⁵
IST-UTL / INESC-ID

R. Alves Redol, 9, 1000-029 Lisbon

{tjvg¹, vmnf², brar³}@immi.inesc.pt, jaj@acm.org⁴, jap@inesc.pt⁵

Abstract

Esta comunicação apresenta uma plataforma para a gestão de aplicações multimodais de forma reutilizável e extensível. Para atingir este objectivo focamos a arquitectura nas Capacidades dos dispositivos e nas Intenções dos utilizadores. Abordamos, em particular, ambientes colaborativos onde a integração de várias modalidades e aplicações obriga à existência de um sistema extensível que combine, em tempo real, diversos componentes provenientes de plataformas heterogéneas. Este artigo descreve o enquadramento proposto e demonstra como interligar aplicações e utilizadores em torno de um ambiente imersivo (tiled display wall), incluindo diferentes modalidades de interacção não convencionais e aplicações.

Keywords

Framework, Interfaces Multimodais, Extensibilidade, Reutilização, Capacidades, Intenções, Colaboração.

1. INTRODUÇÃO

Os dispositivos de interacção tradicionais, tais como o rato e o teclado, são ainda esmagadores perante o uso de modos avançados de interacção. No entanto, grandes esforços foram realizados nas últimas décadas com resultados prometedores, para apresentar alternativas na interacção entre pessoas e qualquer tipo de computador. Adicionalmente, existem diversos projectos para interligar modalidades de interacção e construir aplicações multimodais de utilização natural [4]. Apesar da existência de varias modalidades não convencionais, muito poucas aplicações tiram vantagens destas tecnologias devido ao seu custo de implementação [2]. Num ambiente que integre diversas aplicações multimodais, o esforço da equipa de desenvolvimento é frequentemente desperdiçado devido ao foco numa determinada aplicação. A nossa *plataforma* permite ao utilizador gerir tanto as modalidades de interacção como as capacidades das entradas. Com este propósito, focamo-nos na reutilização das capacidades das entradas e na integração de novas modalidades de interacção em aplicações com um esforço reduzido. Por outro lado, permitimos a sua utilização em ambientes colaborativos, onde cada entrada é controlada de acordo com as

preferências da aplicação e com as capacidades disponíveis num determinado momento. Finalmente, para demonstrar a plataforma num contexto colaborativo, apresentamos um cenário imersivo usando uma *display wall* em que várias modalidades de entrada são utilizadas.

2. PLATAFORMA

O principal objectivo da nossa plataforma é gerir modalidades de entrada e aplicações separadamente permitindo assim, para cada componente, a sua reutilização e extensão. A plataforma foca nas capacidades das entradas e nos interesses dos utilizadores de forma a oferecer em cada momento a modalidade de entrada mais adequada a uma determinada tarefa, no contexto de uma aplicação. As capacidades são um conjunto de *tokens* ou dados de entrada que podem ser enviados a qualquer aplicação, isto é, comandos do rato, *tokens* de gesto, coordenadas de posicionamento ou comandos por voz. Por outro lado, um determinado *token* pode ser enviado por diversos módulos de entrada, por exemplo, “cima” poderia ser fornecido por um dispositivo tipo teclado ou por um reconhecedor suportando gestos corporais. A *plataforma* é responsável pelo redireccionamento dos *tokens* dos módulos de entrada para as aplicações de acordo com as suas preferências. As capacidades e as preferências das entradas e aplicações são definidas *a priori* pelo programador/utilizador. Desta forma, diferentes modalidades podem ser integradas e partilhadas entre aplicações. Além disso, uma nova entrada oferecendo um *token* já previamente conhecido por uma aplicação, pode ser facilmente integrada estendendo as suas modalidades sem alterações no lado da aplicação.

2.1 Visão Geral do Sistema

Para atingir os objectivos de extensibilidade utilizámos

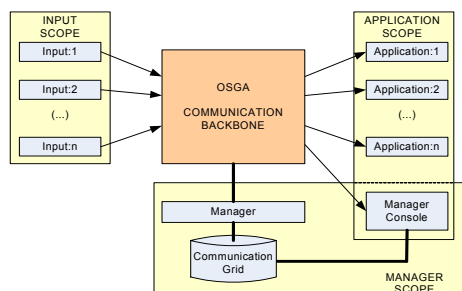


Figure 1 – Arquitectura do Sistema

uma abordagem baseada em mensagens. A arquitectura do nosso sistema, apresentada na Figura 1, está organizada em quatro entidades diferentes: Módulos de entradas, Aplicações, um Gestor e uma infra-estrutura de comunicação denominada OSGA[5]. Os módulos de entrada são as interfaces de comunicação para dispositivos simples tais como o teclado e rato mas também sistemas de reconhecimento de voz ou sistemas baseados em seguimento de câmaras que conseguem entregar tokens multimodais correspondentes a comandos por fala, gestuais ou outro tipo de dados tais como ficheiros. As aplicações estão ligadas à infra-estrutura de comunicação através de uma interface que permite que a interacção tire partido da informação transmitida pelos módulos de entrada. A infra-estrutura de comunicação é responsável pelo redireccionamento de todas as mensagens, permitindo a inter-comunicação entre vários módulos de entradas e múltiplas aplicações. A comunicação e a descoberta de novos módulos de entrada é coordenada por uma quarta entidade que tem a função de gestor. Este módulo de Gestão suporta a configuração e o estabelecimento de comunicações entre módulos de entrada e aplicações. A coordenação consiste num mecanismo de ligação matricial que guarda as capacidades disponibilizadas por cada módulo de entrada activa e as necessidades ou preferências de cada aplicação. Finalmente, o Gestor oferece uma interface que permite ao utilizador seleccionar módulos de entrada alternativos em relação à preferência da própria Aplicação. Esta interface, tal como as aplicações, beneficia de todos os módulos de entrada disponíveis num dado momento por parte da plataforma.

2.2 OSGA: Open Source Group Architecture

Existindo vários protótipos resultantes dos trabalhos de investigação em curso, originou a necessidade de planear trabalhos futuros de integração que permitam juntar pequenos projectos e criar aplicações integradas de larga escala. A OSGA (Open Software Group Architecture)[5] é uma plataforma de integração distribuída baseada em mensagens XML desenvolvida no nosso grupo de investigação para ultrapassar o problema da integração. Esta plataforma pode ser facilmente utilizada em aplicações futuras bem como integrada com protótipos actuais/passados. Foi desenvolvida sobre o módulo XmlBlaster[6] e providencia a existência de vários clientes a receber mensagens de acordo com mecanismos de subscrição e publicação. Basicamente, todas as mensagens enviadas para o sistema são re-direccionadas para todos os clientes que subscreveram a um dado tópico de mensagem. O sistema suporta igualmente, filtrar mensagens de acordo com propriedades adicionais tais como a identificação do remetente, ou indícios sobre o conteúdo da mensagem.

2.3 Protocolo de Comunicação

Para suportar a interacção entre módulos de entrada, aplicações e o gestor, organizamos o nosso protocolo de mensagens XML (Figura 2) em quatro fases distintas. A primeira fase é a inicialização das entradas, que permite

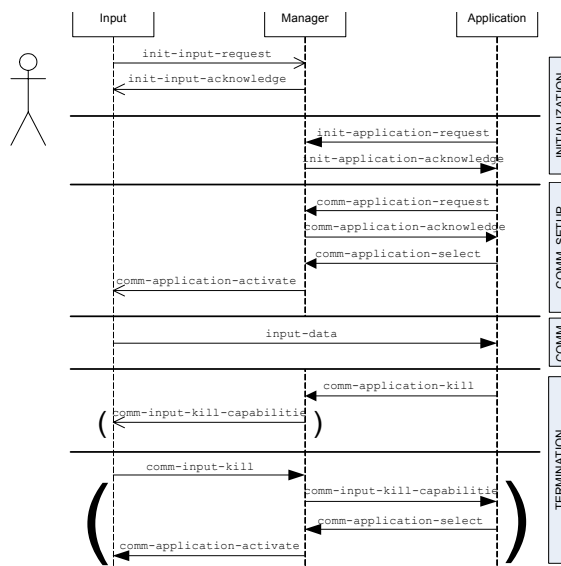


Figure 2 – Diagrama de Sequência (Protocolo)

registar no gestor novos módulos de entrada assim como suas capacidades. É igualmente possível, registar as capacidades desejadas de cada aplicação de forma a identificar os modulos de entrada que suportam tais capacidades. A terceira fase é a configuração de uma comunicação para estabelecer um canal de comunicação entre uma dada entrada e uma aplicação com o objectivo desta receber dados de entrada. Finalmente, o protocolo suporta a suspensão/re-estabelecimento de um canal de comunicação entre uma entrada e uma aplicação e sua finalização que é coordenada pelo Gestor.

Inicialização: A inicialização de um módulo de entrada é executada pela mensagem *init-input-request* usando uma canal provisório de comunicação. Como ilustrado na Figura 3, cada entrada fornece a sua identificação assim como a lista de capacidades suportadas pelo módulo de entrada. A identificação é formada pelo tipo do dispositivo (exemplo: Fala, EMG, Dispositivo de Seguimento...) e a chave que será usada pelo gestor para atribuir um identificador global (exemplo Fala:1) ao modulo de entrada. Esta informação é armazenada no gestor, e a nova identificação é utilizada pelo módulo de entrada de forma a criar um canal que estará disponível para as aplicações interessadas. As aplicações, tal como descrito na Figura 2, requerem também uma inicialização afim de serem identificadas pelo gestor. Por outro lado, a lista de capacidades requisitadas é fornecida e armazenada na matriz de comunicação do Gestor. O Gestor define um identificador global (exemplo Paint:1) e fornece uma lista dos módulos de entrada e das capacidades disponíveis para satisfazer as necessidades da aplicação. A Figura 3 apresenta um exemplo da resposta em *init-application-acknowledge*. Esta informação é usada posteriormente pela aplicação de forma a estabelecer a comunicação com os módulos de entrada necessários para suportar a interacção desejada.

Comunicação e Interação (Figura 4): As aplicações podem obter, a qualquer momento, a lista das entradas e

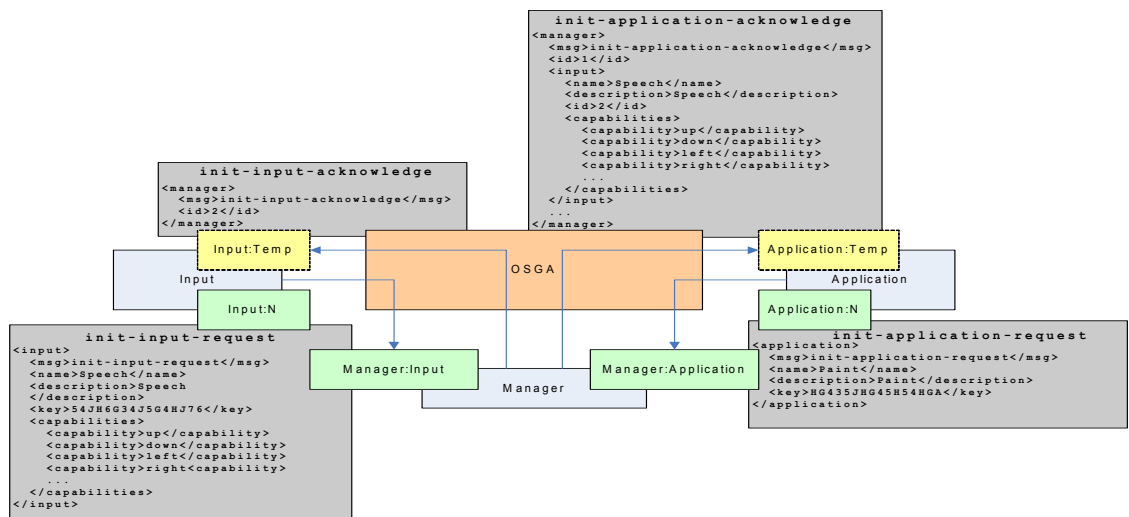


Figure 3 – Diagrama de Inicialização

capacidades (comm-application-request), e são informadas pelo gestor do estado actualizado da plataforma (comm-application-acknowledge). Com essa informação, as aplicações podem seleccionar as capacidades conforme as preferências dos utilizadores como pode ser visto no seguinte exemplo XML.

Application->Manager:Application(Filter: <none>)

```
<application>
  <msg>comm-application-select</msg>
  <name>Paint</name>
  <id>1</id>
  <input>
    <name>Speech</name>
    <id>2</id>
    <comm-type>XML</comm-type>
    <capabilities>
      <capability>up</capability>
      <capability>down</capability>
    </capabilities>
  </input>
  ...
</application>
```

O gestor é responsável por comunicar aos módulos de entrada (com as capacidades requisitadas) que existem aplicações que estão interessadas em receber seus dados correspondente a uma dada capacidade. Desta forma, uma entrada só publica os dados que foram requisitados, isto é, no cenário de duas aplicações A e B em que a aplicação A quer “esquerda”, “direita” e “seleccionar” e a aplicação B quer “acima”, “abaixo” e “seleccionar”, só as entradas “esquerda”, “direita”, “acima”, “abaixo” e “seleccionar” serão publicadas dado que correspondem a união de dados que foi requisitado por cada aplicação. Cada aplicação recebe exclusivamente os dados que requisitou tirando partido da funcionalidade de filtragem oferecida pela OSGA. A comunicação é estabelecida entre os módulos de entradas e as aplicações que subscreveram os tópicos dessas entradas. Utilizando esse

mecanismo de subscrição, consegue-se uma comunicação de n:n entre módulos de entrada e aplicações. Os mecanismos de filtragens permitem uma melhor regulação do tráfego de mensagens graças ao uso de filtros e descrições ricas (mensagens trocadas) que podem ser actualizadas a qualquer momento. Os módulos de entrada publicam as mensagens identificadas pelo tópico cuja a nomenclatura é a seguinte: {tipo da entrada}:{identificação}:data.

Input -> Paint:1:Data (Filter:up,down)

```
<input>
  <msg>input-data</msg>
  <data>up</data>
</input>
```

Suspender e Continuar: A comunicação entre um módulo de entrada e uma aplicação pode ser suspensa por iniciativa da aplicação através do cancelamento da subscrição da entrada. Para restabelecer a comunicação, basta subscrever novamente o tópico desejado. Caso um módulo de entrada termine durante a suspensão de uma comunicação, o Gestor informa a aplicação do sucedido (comm-application-acknowledge). A aplicação pode opcionalmente comunicar ao gestor que suspendeu a comunicação de uma entrada. Essa informação é registada pelo Gestor.

Terminar: As aplicações podem terminar a sua sessão em qualquer altura desde que publiquem essa intenção ao Gestor (comm-application-kill). O Gestor informa os módulos de entradas caso não haja mais aplicações a requisitar um determinado dado. Dessa forma, é possível evitar tráfego de mensagens desnecessário. Por exemplo, se a entrada estiver a publicar “para cima” e “para baixo” mas a única aplicação que estava interessada em “para cima” terminou então a entrada começa publicar apenas “para baixo”. Quando uma entrada termina a comunicação (comm-input-kill), o Gestor informa as aplicações afectadas que têm de modificar a sua dependência de entradas/capacidades. A lista actual das entradas e respectivas capacidades é então enviada para as aplicações efectuarem uma nova selecção (como em comm-application-select).

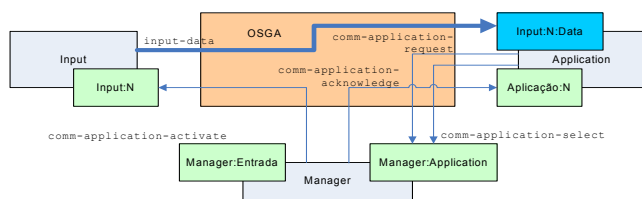


Figure 4 – Diagrama de Comunicação

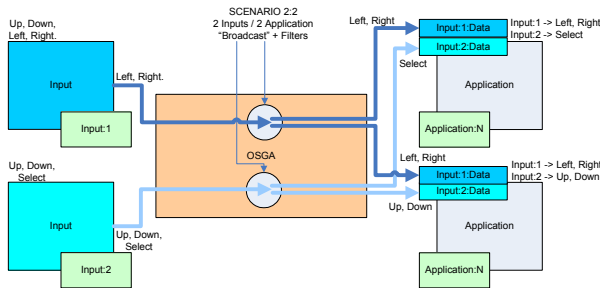


Figure 5 – Cenário colaborativo

Keep-alive: Para detectar situações de falha na comunicação dos módulos de entrada ou das aplicações com o gestor quanto a sua terminação, desenvolveu-se uma sinalização *keep-alive* entre essas entidades e o gestor. Se uma falha for detectada o protocolo Terminar será seguido dependendo da entidade onde foi detectada a falha.

3. CONTEXTO DE UTILIZAÇÃO

A plataforma apresentada neste artigo é a base para todos os protótipos de interação e aplicação futuros no nosso ambiente imersivo, LEMe Wall [1]. Assim, para além de possibilitar uma estrutura colaborativa, possibilitamos também a reutilização de todos os protótipos de interação criados noutros contextos. Além do ecrã de larga escala existente, o ambiente é complementado por um conjunto de sensores e actuadores que permitem aumentar a imersão. Através de vários dispositivos de interação não convencionais (Sensores de pressão, sensores de posicionamento, câmaras para reconhecimento de gestos e microfones para reconhecimento de voz) podemos ter diversos utilizadores a interagir simultaneamente com o ambiente no mesmo ou em diferentes contextos/aplicações. A utilização da nossa plataforma neste ambiente permite que cada módulo de Entrada criada seja reutilizado por qualquer investigador que desenvolva uma aplicação e torna fácil adicionar novas Entradas e Aplicações ao leque de componentes existente.

3.1 Cenários de Aplicação

A Figura 5 apresenta um cenário com duas aplicações e duas Entradas. Com a descrição deste cenário, pretendemos demonstrar as vantagens da estrutura apresentada, contemplando duas entradas diferentes que controlam uma aplicação, comandos diferentes de uma entrada usados por duas aplicações simultâneas e também comandos de uma entrada que afectam ambas as aplicações. Consideremos um cenário colaborativo onde dois utilizadores estejam a desenhar um modelo na LEMe Wall. As Entradas seleccionadas são o TabletPC e os gestos, operados pelos utilizadores. A Aplicação usada é o Gides++[3], que foi desenvolvida no nosso grupo de investigação. Com a nossa abordagem, os utilizadores podem operar dentro da mesma aplicação simultaneamente, o que aumenta a colaboração e o desempenho do processo de desenho (Figura 6). Assim, os utilizadores podem interagir sobre a mesma aplicação mas focando em aspectos ou tarefas diferentes. Vamos



Figure 6 –Desenho Colaborativo na LEMe Wall

agora considerar um cenário com duas aplicações, um visualizador de informação geográfica e um visualizador de imagens satélite de alta-resolução. Cada utilizador controla uma das aplicações através de uma determinada Entrada. Considerando que os utilizadores operam sobre a mesma área e escala dos respectivos mapas, é possível ter comandos de uma Entrada utilizados por ambos e que afecte ambas as aplicações. Assim, neste cenário, considerando as acções de Zoom ou Pan, conseguimos manter o sincronismo entre as duas aplicações. Este exemplo contempla comandos enviados para uma aplicação e para duas aplicações independentes. Os cenários apresentados podem ser realizados simultaneamente criando redes mais complexas de comunicação.

4. CONCLUSÕES

Apresentámos uma estrutura para gerir aplicações multimodais em ambientes distribuídos focando na extensibilidade e reutilização das modalidades de entrada. Atingimos este objectivo separando entradas e aplicações e controlando a sua comunicação através de um protocolo baseado em mensagens. A contribuição principal deste trabalho é a possibilidade para adaptar em tempo real as entradas e aplicações conforme as capacidades disponíveis e a vontade do utilizador num determinado momento. São apresentados cenários colaborativos motivadores que demonstram as possibilidades da plataforma.

5. REFERENCES

- [1] Araújo, B., et al. Leme Wall: Desenvolvendo um sistema de Multi-projecção. Proceeding of 13^o EPCG, Vila Real, Portugal, Out 2005.
- [2] Flippo, F., et al. A framework for rapid development of multimodal interfaces. In *Proc. of the 5th ICMI 03*. ACM Press, New York, NY, 109-116
- [3] Jorge, J.,Silva, N., Cardoso, T. Gides++, Proceedings of 12^o EPCG, Porto, Portugal, Out 2003
- [4] Krahnstoeber, N., et al. *A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays*. in 4th IEEE ICMI 2002.
- [5] <http://www.osga.net>, Junho 2006
- [6] <http://www.xmlblaster.org>, Junho 2006