UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

SUPORTE PARA INTERACÇÃO MULTIMODAL BASEADA EM TELEVISÃO

José Alexandre Cardoso

MESTRADO EM INFORMÁTICA

2010

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

SUPORTE PARA INTERACÇÃO MULTIMODAL BASEADA EM TELEVISÃO

José Alexandre Cardoso

PROJECTO

Trabalho orientado pelo Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte

MESTRADO EM INFORMÁTICA

2010

Agradecimentos

O desenvolvimento deste trabalho foi levado a cabo com o apoio de várias pessoas, apoio esse evidenciado em diferentes níveis. A todos os que me apoiaram o meu sincero agradecimento.

Quero destacar em primeiro lugar a minha família pois são eles o mais importante na minha vida. Agradeço à minha esposa Ana e aos meus filhos Rodrigo e Matilde por tantas vezes me quererem tirar do escritório mas compreenderem sempre quando não era possível; enchendo-me de mimos quando a saída acabava por acontecer. Agradeço aos meus Pais e ao meu irmão não só pelo orgulho que me transmitem quando me vêem empenhado no meu percurso académico e profissional mas também pelos fins-de-semana que não podemos partilhar e que teremos que compensar no futuro.

Ao nível académico quero agradecer ao Professor Carlos Duarte pelo sim imediato quando lhe foi sugerido que fosse meu orientador de mestrado; agradecer-lhe também pela impressionante clarividência quando o assunto versava sobre interacção multimodal, área em que o meu conhecimento era muito limitado mas em que me senti sempre bem acompanhado.

Agradecer também à equipa constituída pelo David Costa, o Daniel Costa e o Pedro Feiteira, que exercitou a Framework MMX através da implementação de dois protótipos.

Ao nível profissional quero agradecer à minha empresa, Novabase Digital TV, nas pessoas do Paulo Sousa e do Pedro Afonso que sendo meus superiores sempre me deram a liberdade suficiente para poder realizar este trabalho sem que com isso me tenham desvalorizado de alguma forma (bem pelo contrário). Agradecer também aos meus colegas que colmataram as minhas ausências com especial destaque para o Vitor Fonseca, o Filipe Norte e o Rui Silva.

Resumo

A televisão é um elemento fundamental na vida de milhões de pessoas. Com os avanços tecnológicos evidenciados nos últimos anos, também a televisão evolui para patamares tecnológicos mais avançados. As utilizações dadas a tal sistema têm vindo a diversificar-se, passando da simples apresentação de áudio e vídeo para a disponibilização de aplicações mais complexas. Tanto assim é que os aparelhos utilizados habitualmente em conjunto com a televisão no tratamento de conteúdos de áudio e vídeo, como, por exemplo, os videogravadores ou os amplificadores de som, têm vindo a ser substituídos por aparelhos mais avançados, as Set Top Boxes, que vão para além do tratamento de imagem e som.

A recente evolução e massificação de diferentes formas de aceder a dispositivos computacionais como as STBs ou os telemóveis, aliadas à forma ubíqua como a televisão está presente no dia-a-dia de milhões de pessoas, evidenciam a oportunidade de utilizar este meio através de diferentes modalidades de entrada e saída. Esta utilização de diferentes modalidades de entrada e saída permite, não só, alargar o espectro de utilizadores das aplicações desenvolvidas, mas também proporcionar uma experiência mais diversificada aos actuais utilizadores.

Para além da interacção com as aplicações através de diferentes modalidades de entrada/saída, um dos grandes desafios que se colocam, centra-se em perceber como é que os programadores de aplicações conseguem, de forma eficaz, desenvolver ou adaptar aplicações que utilizem em simultâneo as várias modalidades de entrada/saída.

Neste sentido desenvolveu-se a Framework MMX, que pretende disponibilizar algumas camadas de abstracção que serão úteis no desenvolvimento de aplicações, permitindo-lhes estarem logicamente separadas das modalidades utilizadas. Para além da Framework MMX propriamente dita, desenvolveram-se protótipos para exercitarem e ajudarem ao desenvolvimento desta. Destes protótipos destacam-se o Tabuleiro de Xadrez, desenvolvido em conjunto com a Framework, o Jogo do Galo desenvolvido por uma equipa diferente para aferir da qualidade da Framework MMX e ainda a Aplicação EPG definida para testar a Framework em ambiente de Televisão e implementada pela mesma equipa que desenvolveu o Jogo do Galo. O protótipo Jogo do Galo evidenciou algumas deficiências na Framework que foram corrigidas antes de se iniciar a implementação da Aplicação EPG. A Aplicação EPG mostrou a utilidade e facilidade de uso da Framework na integração de diferentes modalidades numa aplicação.

Palavras-Chave: Multimodal; Xadrez; Televisão; Framework MMX;

Abstract

Television is a key device in the life of millions of people. With the technological achievements over the last years, television has also evolved to the highest technological standards. The usage given to such a system is becoming more diverse, moving from the classical audio and video presentation to more complex applications. The proof is that the usual devices used together with the TV set, like video recorders or sound amplifiers, have been replaced by more advanced devices, like the Set Top Boxes, which go beyond the presentation of image and sound.

The recent evolution and massification of different ways of interacting with computational devices like STBs or cell phones, together with the ubiquitous way that television is present in the day-to-day life of millions of people, present an opportunity to use this system through different input/output modalities. This usage of different input/output modalities allows, not only, to broaden the spectrum of users of the developed applications, but also to provide a more diverse experience to current users.

Beyond the interaction with the applications using different input/output modalities, one of the big challenges is to understand how can application developers develop or adapt, in an effective way, applications that use simultaneously several input/output modalities.

To answer to this challenge, the Framework MMX was developed, aiming to provide a few abstraction layers that are useful in developing applications, allowing them to be logically separated from the used modalities. In addition to the Framework MMX, a set of prototypes with different goals was also developed: Chess Board was developed together with the Framework to assist in its definition; Tic Tac Toe game was developed by a different team to evaluate the quality of the Framework MMX; and an EPG Application defined to test the Framework in a television environment. The Tic Tac Toe game showed some flaws in the Framework which were corrected before starting the EPG Application development. The EPG Application showed that the Framework is useful and easy to use when integrating different modalities in an application.

Keywords: Multimodal; Chess; Television; Framework MMX;

Conteúdo

LISTA DE FIGURAS

GLOSSÁRIO

API	Application Program Interface
DVB	Digital Video Broadcasting
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
GUIDE	Gentle User Interface for Disabled and Elderly people
IDE	Integrated Development Environment
JMS	Java Message Service
MMX	MultiModal Framework
MHP	Multimedia Home Platform
STB	Set Top Box
SVN	Subversion

Capítulo 1

Introdução

Este projecto surge enquadrado no Mestrado em Informática do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa.

A escolha do tema "Suporte para interacção multimodal baseada em televisão" surge devido à minha experiência profissional, desde sempre ligada à área da televisão digital, mas também devido à existência do projecto Europeu GUIDE [1], que versa sobre interacção multimodal baseada em Televisão, no qual o Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa terá um papel preponderante.

1.1 Motivação

A Televisão tem sido, desde que foi lançada no mercado na década de 1930, um elemento fundamental na vida de milhões de lares. O valor do acesso à informação e ao entretenimento através deste meio de comunicação é globalmente reconhecido. A televisão permite, portanto, a uma grande franja da população do mundo civilizado, o acesso a experiências que não seriam alcançáveis de outra forma.

Desde a criação da Televisão que a tecnologia tem evoluído a uma velocidade vertiginosa. Esta evolução tecnológica não diminuiu a importância da Televisão, pois esta foi sendo adaptada às novas exigências e aproveitando as novas oportunidades. Para além do avanço tecnológico na qualidade dos conteúdos, da infra-estrutura e dos dispositivos, também os modelos de negócio foram alterados. Uma das grandes inovações foi a utilização de sinal digital em substituição de sinal analógico. A transmissão de sinal digital possibilitou que conteúdos fossem vendidos através de dispositivos que interagem com a Televisão para garantir o acesso condicional a esses conteúdos. Esses dispositivos variam na forma e capacidade, sendo que o que mais se destaca são as Set Top Boxes (STBs). As STBs são dispositivos que podem ter várias funcionalidades dependendo dos requisitos. Estas funcionalidades respondem cada vez mais às necessidades e desejos dos utilizadores (telespectadores!), existindo uma aposta por parte dos fabricantes de STBs no desenvolvimento de aplicações cada vez mais

sofisticadas. Para além os avanços na tecnologia das STBs, também os avanços recentes das interfaces pessoa-máquina, como tecnologias baseadas em gestos, toque (multi-toque), voz e outras modalidades de entrada e saída, começam a possibilitar a sua utilização pela população em geral. Mas mais do que a utilização de cada uma destas modalidades per si, importa conjugar a utilização de várias modalidades diferentes para que sejam utilizadas em simultâneo conforme as necessidades específicas dos utilizadores.

Utilizadas da forma correcta, estas diferentes modalidades de interacção podem abrir novos horizontes à utilização da Televisão (recorrendo a uma STB), fornecendo novas funcionalidades, mas, principalmente, alargando o espectro de utilizadores de forma a abranger utilizadores com necessidades especiais. Desta conjunção entre modalidades de entrada/saída com STBs surge o projecto GUIDE [1], co-financiado pela União Europeia. Como objectivo principal deste projecto surge o desenvolvimento de uma Framework que permita aos programadores de aplicações para STBs (ou outros dispositivos computacionais) integrar facilmente novas modalidades de entrada/saída. Deste modo, as STBs passam a poder explorar várias modalidades de entrada/saída que se adaptem às especificidades dos utilizadores, nomeadamente dos utilizadores com necessidades especiais, destacando-se as pessoas idosas que, por norma, sofrem de alguma dificuldade motora, sensorial ou cognitiva, e frequentemente, da combinação destas.

Figura 1: Guide - Enquadramento geral

Tendo em conta o projecto GUIDE, na sua vertente de interacção multimodal para Televisão, o presente projecto pretende responder ao desafio de desenvolver uma Framework que permita aos programadores integrar facilmente várias modalidades de entrada/saída. A integração destas modalidades de entrada/saída, visa permitir ao utilizador interagir de um modo natural com diferentes aplicações desenvolvidas para correr num ambiente de Televisão suportada por uma STB.

1.2 Objectivos

O objectivo deste trabalho é, de uma foram genérica, responder ao desafio de desenvolver uma Framework que permita aos programadores integrar facilmente várias modalidades de entrada/saída, que permitam interagir com diferentes aplicações desenvolvidas para correr num sistema computacional. Futuramente, pretende-se que esta Framework possa ser utilizada em ambiente de Televisão suportada por uma STB. A concretização deste objectivo, bem como a forma de o atingir, pode ser melhor compreendida através da sua partição em objectivos mais finos:

- Desenho e desenvolvimento da Framework MMX:

  Esta Framework deverá permitir a fácil integração de diferentes modalidades de entrada/saída. Como base de comunicação entre os diferentes módulos será utilizado um sistema de mensagens (Publisher/Subscriber) onde cada um dos componentes de entrada será considerado um Publisher e cada um dos componentes de saída será considerado um Subscriber. De notar que os componentes de entrada publicam as mensagens e essas mensagens são tipicamente recebidas por um módulo (a aplicação) que não um componente de saída. Esse módulo publicará então mensagens de saída, essas sim, subscritas pelos componentes de saída. Para além dos componentes de entrada/saída multimodais, também outros módulos distintos podem fazer uso do sistema de mensagens de forma manter a arquitectura o mais modular possível. Estas variantes nos módulos que assumem papéis Publisher e/ou Subscriber estendem-se também aos componentes de entrada que podem em certas circunstâncias actuar como Subscribers e aos componentes de saída que, também em certas circunstâncias, podem actuar como Publishers.

  Um dos requisitos principais da Framework será encontrar uma forma eficiente de realizar a integração das várias modalidades para que possam ser utilizadas conjuntamente.

- Desenvolvimento de um protótipo:

  O protótipo a desenvolver será um tabuleiro de Xadrez cuja principal funcionalidade é permitir exercitar as diferentes modalidades de entrada/saída, bem como a comunicação entre os diferentes módulos. Este protótipo não pretende cobrir todos os casos de uso de um tabuleiro de xadrez, mas apenas aqueles que permitam exercitar os componentes de entrada/saída a serem utilizados, bem como os conceitos definidos na Framework.

- Validação da Framework

A Framework será validada através do desenvolvimento de um ou mais protótipos, a desenvolver por programadores que nunca estiveram envolvidos na definição da Framework. Para além deste exercício servir para validar a Framework, também fornecerá informação a ser utilizada no refinamento da mesma.

1.3 Metodologia

O desenvolvimento deste projecto assentou numa metodologia centrada nos utilizadores da Framework MMX – os programadores de aplicações multimodais.

O planeamento inicial não previa iterações no desenvolvimento, mas este foi alterado de forma a que programadores não envolvidos no desenvolvimento da Framework MMX a validassem e providenciassem informação sobre a utilização da mesma, informação essa que seria utilizada no seu refinamento. Neste sentido, o trabalho dividiu-se nas seguintes fases:

- Desenvolvimento do protótipo Tabuleiro de Xadrez e da primeira versão (v0.1) da Framework MMX.
  Esta fase decorreu durante os primeiros oito meses do projecto resultando num protótipo funcional de um tabuleiro de xadrez. Este protótipo utiliza a Framework MMX para suportar diversas modalidades de entrada e saída em simultâneo.
- Validação da versão v0.1 da Framework MMX – o protótipo Jogo do galo.
  Após a conclusão do tabuleiro de xadrez bem como da primeira versão (v0.1) da Framework MMX, esta foi entregue a uma equipa de desenvolvimento de forma a ser validada. A equipa de desenvolvimento implementou o protótipo Jogo do Galo sobre a Framework MMX, o que permitiu identificar alguns pontos a melhorar nesta. Esta primeira fase de validação demorou cerca de duas semanas.
- Desenvolvimento da versão v0.2 da Framework MMX.
  Com a informação recolhida na primeira fase de validação, procedeu-se ao refinamento da Framework MMX. As alterações incidiram principalmente no isolamento entre a Framework e as aplicações, resultando não só numa nova versão (v0.2) da Framework MMX, mas também na adaptação dos protótipos já desenvolvidos (Tabuleiro de Xadrez e Jogo do Galo) à nova versão da Framework. Estes desenvolvimentos, embora tivessem melhorado substancialmente a Framework MMX, demoraram apenas uma semana.
- Validação da versão v0.2 da Framework MMX – a Aplicação EPG.

Após a conclusão da versão v0.2 da Framework procedeu-se a nova validação. A equipa de desenvolvimento implementou uma aplicação EPG sobre a Framework MMX v0.2. Desta implementação concluiu-se que as aplicações (protótipos) estão isoladas da Framework. Esta fase demorou cerca duas semanas.

- Finalmente, o relatório foi redigido. De salientar que o relatório foi sendo redigido desde a conclusão da primeira versão (v0.1) da Framework MMX sendo constantemente actualizado durante um mês e meio.

1.4 Contribuições

O principal resultado deste projecto é a Framework MMX que permite o desenvolvimento de aplicações computacionais multimodais. Esta Framework assenta numa arquitectura Publisher/Subscriber e apresenta um conjunto de APIs bem definidos que formam uma camada de abstracção adequada à integração de diversas modalidades de entrada/saída numa mesma aplicação.

Para além da Framework MMX, também resulta deste projecto um conjunto de três protótipos multimodais que serviram para definir e validar a Framework MMX. Estes protótipos são um Tabuleiro de Xadrez, um Jogo do Galo e uma Aplicação EPG multimodais. O conjunto de protótipos é distribuído junto com a Framework de forma a servirem de exemplo aquando da construção de novas aplicações que façam uso da Framework MMX.

As contribuições acima mencionadas estão disponibilizadas no sítio da internet dedicado à Framework MMX [2].

1.5 Estrutura do documento

Este documento divide-se em seis capítulos distintos. O primeiro capítulo faz a introdução do trabalho a realizar descrevendo sucintamente a motivação e os objectivos. O segundo capítulo apresenta um enquadramento do trabalho relativamente aos assuntos que mais o influenciam, como o projecto GUIDE, os sistemas embebidos e direccionados para televisão digital, as modalidades de entrada/saída e também o Xadrez. No capítulo terceiro é apresentado o desenho da Framework MMX na sua versão v0.2 e do protótipo Tabuleiro de Xadrez. No capítulo quarto é explicada a implementação da versão v0.2 da Framework MMX e do Tabuleiro de Xadrez, sendo ainda introduzida a implementação externa de outros protótipos. O capítulo quinto apresenta os resultados obtidos no decurso deste trabalho relativamente à Framework MMX e aos protótipos que a exercitaram. Finalmente são apresentadas as conclusões no sexto capítulo seguidas de algumas sugestões para trabalho futuro.

Capítulo 2

Enquadramento

Este projecto aborda vários temas que se inter-relacionaram durante a concepção e desenvolvimento do mesmo. Neste capítulo é explicado o enquadramento actual de cada um dos temas chave que basearam o trabalho desenvolvido, desde a definição dos objectivos até à análise dos resultados.

2.1 Projecto GUIDE

O projecto GUIDE serve de ponto de partida para este projecto no que diz respeito à definição de objectivos.

Um enquadramento geral da implementação do sistema GUIDE está explícito na Figura 2, onde se tornam visíveis as várias partes do sistema bem como a estrutura do Projecto. A Framework GUIDE (WP5) prevê incorporar um conjunto de mecanismos que possibilitem a utilização de multimodalidades para interacção com aplicações disponibilizadas, principalmente, a partir de uma plataforma TV. No seu núcleo, encontra-se um módulo (WP4) que deverá ser capaz de adaptar a interacção às diferentes características dos seus utilizadores. As diferentes modalidades de entrada e saída (WP3) comunicarão com este núcleo através de uma API. Outra API definirá a forma de comunicar com as aplicações (WP6) que serão construídas com base nesta Framework. O principal resultado do projecto GUIDE, para quem desenvolve aplicações, será a arquitectura de execução de aplicações multimodais adaptativas e as recomendações de desenvolvimento, que incluem preocupações ao nível da diversidade de utilizadores, e das suas características, nomeadamente, no que diz respeito ao nível da acessibilidade. Assim, para programadores de aplicações, o projecto GUIDE tenta atingir diversos objectivos com a definição desta arquitectura, dos quais se destaca a necessidade dos programadores de uma camada de abstracção que lhes permita facilmente integrar novas modalidades de entrada/saída. A Framework MMX a desenvolver no decurso do presente projecto tem em conta esta necessidade do projecto GUIDE e pretende responder aos requisitos identificados de forma a nele ser reutilizada.

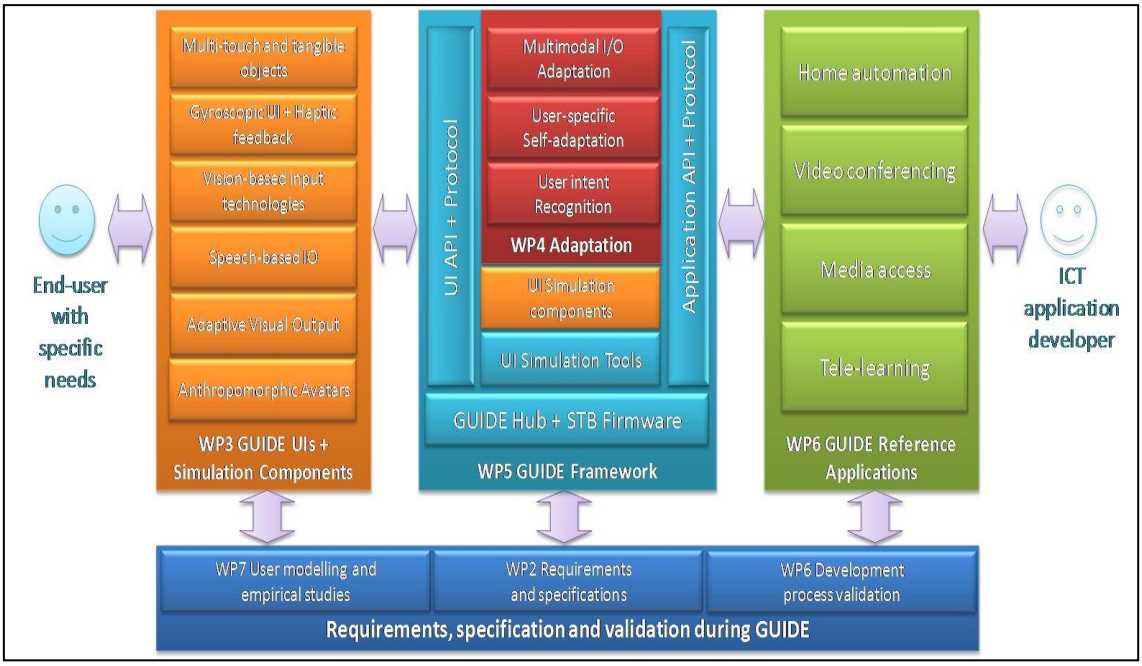

Figura 2: Sistema GUIDE – Implementação prevista

Uma vez que, no contexto do presente projecto, apenas uma aplicação será desenvolvida – Protótipo de Tabuleiro de Xadrez – pretende-se que programadores ligados ao projecto GUIDE possam exercitar a Framework na implementação de uma segunda aplicação com o objectivo de perceber a utilidade e usabilidade da mesma e se representa uma mais-valia para o projecto GUIDE. Os resultados obtidos deste exercício serão tidos em conta na concretização da Framework e apresentados na secção correspondente deste documento.

2.2 Set Top Boxes e Sistemas embebidos

As Set Top Boxes são consideradas sistemas embebidos[3] devido à configuração especializada que existe em termos da sua concepção – são concebidas para responder a uma necessidade específica relacionada com a visualização de conteúdos de Televisão (Áudio, Vídeo, Dados, etc.).

Apesar de serem consideradas sistemas embebidos, a variedade de Set Top Boxes e das suas aplicações são enormes e, portanto, a capacidade de processamento disponível em cada modelo de STB varia muito.

Enquanto sistema embebido, a resposta do mercado é ainda limitada comparada com outros mercados como, por exemplo, o dos dispositivos de comunicação móvel (telemóveis). No caso específico dos telemóveis, o mercado e os utilizadores investiram de tal forma que as técnicas de miniaturização e a capacidade de processamento destes dispositivos são avançadíssimas. Em contraste, no mercado das STBs, o investimento é menor, sendo mais direccionado para os conteúdos do que para as funcionalidades, pelo

que a capacidade de processamento destes dispositivos tende a ser mais limitada que nos telemóveis. Apesar de tudo, existem no mercado alguns modelos de STBs que têm já uma considerável capacidade de processamento e são estes modelos que devem ser tidos em conta quando falamos de funcionalidades avançadas, como a interacção multimodal e o suporte para diversas aplicações.

O facto de existirem modelos de STBs com considerável capacidade de processamento, não invalida a necessidade de ter em conta que é um sistema embebido e que questões como limitação da capacidade de processamento, limitação da capacidade de armazenamento e requisitos de dimensões padrão destes dispositivos devem ser tidos em conta quando se desenvolvem aplicações para os mesmos

Para desenvolver aplicações para sistemas embebidos utiliza-se, por norma, uma máquina de desenvolvimento diferente da máquina alvo onde é suposto executar a aplicação. A aplicação é posteriormente validada num simulador ou, preferencialmente, na máquina alvo.

No contexto do presente projecto, não existia nenhum protótipo de STB que pudesse ser utilizado, razão pela qual se optou por desenvolver e testar a Framework e a aplicação num PC.

2.3 Televisão Digital

Devido ao crescimento do mercado de televisão digital proporcionado, principalmente, pelos fornecedores de conteúdos que optam por fornecer soluções de acesso condicional onde os utilizadores têm de pagar conforme os conteúdos que pretendem visualizar, mas também, mais recentemente, pelo decretar do fim do sinal analógico de televisão [4], os protocolos utilizados para a transmissão de conteúdos para televisão digital vêm desde de há muito a serem padronizados. Uma das normas actualmente mais bem implantadas e aceites é a norma DVB [5]. Esta norma começou por definir o modo como o sinal de televisão digital deveria ser transmitido em termos físicos e em termos de sinalização dos serviços transmitidos.

A aceitação e potencialidade da televisão digital aliada à norma DVB foi percebida pelo mercado, razão que levou o organismo responsável por esta norma – ETSI – e os seus parceiros, a investir na normalização de outros casos de uso no contexto da televisão digital, sendo uma das normas criada a Multimedia Home Platform (MHP) [6].

A MHP é uma norma aberta que visa definir uma camada de abstracção que permite que aplicações desenhadas para televisão sejam transmitidas e executadas numa STB. Esta norma baseia-se em JAVA [7], exigindo que as STBs corram uma máquina

virtual de JAVA e que exista a camada que exporta uma API bem definida pela norma MHP.

Tendo em conta a existência de normas específicas para Televisão Digital, o presente projecto não quebra qualquer possibilidade de que estas normas sejam respeitadas e persegue ainda o objectivo de facilitar a implementação dessas normas utilizando, por exemplo, a linguagem JAVA como linguagem de programação da Framework e do protótipo.

2.4 Modalidades de Entrada/Saída

No contexto actual de aplicações para sistemas computacionais, surgem cada vez mais diferentes modalidades de entrada e saída que permitem interagir com essas aplicações. Muitas dessas modalidades têm-se implantado e generalizado, como é o caso do reconhecimento de gestos, de toque (multi-toque) e voz que, por exemplo, são comummente utilizados para interagir com telemóveis. Outras modalidades também surgiram com sucesso em aplicações menos generalistas, como é o caso do WiiMote da consola Wii da Nintendo, que permite reconhecimento de gestos, funciona como dispositivo apontador, produz vibração e sons. Também o projecto Natal da Microsoft [8] promete revolucionar a interacção com dispositivos computacionais, sendo capaz de reconhecer os movimentos corporais e faciais, bem como voz, incluindo o seu tom.

2.5 Tabuleiro de Xadrez

O Xadrez é um jogo sobejamente conhecido e popular por todo o mundo. É um jogo de tabuleiro em que cada jogador realiza a sua jogada alternando com o outro jogador. Este tipo de interacção simples e bem controlada foi decisiva para a escolha do Xadrez como tema para o protótipo.

O facto de ser um tabuleiro também é importante pois facilita a definição do espaço de acção sobre o qual as várias modalidades irão actuar.

O Xadrez tem também uma série de regras que têm de ser cumpridas em cada jogada. A validação das regras de Xadrez está fora do âmbito do presente projecto. No entanto é importante salientar o papel da validação de regras para exercitar o retorno que é dado ao utilizador.

Também fora do contexto do presente projecto está a implementação de qualquer motor com inteligência artificial que consiga jogar Xadrez.

Apesar de o Xadrez não ser, em si, um objectivo do projecto, é importante ter em conta as especificidades técnicas deste jogo, por exemplo, em termos de componentes

(casas do tabuleiro, peças, …) e a forma como estas especificidades permitem exercitar a Framework a desenvolver.

Por outro lado, o mundo do Xadrez é vasto e merece uma análise mais profunda. Uma vez feito o protótipo, não é de descartar a possibilidade de estendê-lo para uma aplicação mais abrangente que inclua as regras, inteligência artificial, etc., sempre conjugado com a Framework MMX e a interacção multimodal.

Sabendo que o Xadrez é também um jogo muito explorado em termos de computação a vários níveis, como jogos online, super computadores capazes de derrotar o mais galardoado dos humanos [9], dispositivos portáteis de entretenimento, é importante perceber como são tratadas as especificidades do jogo nessas diversas implementações e como essas implementações podem ser adaptadas e/ou aproveitadas num contexto de interacção multimodal de uma STB. Destas especificidades, destacam-se a representação das jogadas e do tabuleiro[10], as jogadas especiais e as regras a cumprir [11], a possibilidade de ser distribuído estando os dois jogadores em localizações distintas, a necessidade de correr em sistemas embebidos como telemóveis e consolas de jogos, etc.

Capítulo 3

Desenho da solução

Este projecto apresenta duas vertentes principais: a Framework MMX e o protótipo do tabuleiro de Xadrez. Estes dois módulos têm objectivos distintos.

A Framework MMX visa fornecer um mecanismo para que várias modalidades de entrada/saída possam ser facilmente integradas no desenvolvimento de aplicações, enquanto o protótipo do tabuleiro de Xadrez visa exercitar a Framework MMX.

3.1 Requisitos da Framework MMX

A Framework MMX pretende ser uma Framework que permita a integração fácil de novas modalidades de entrada/saída. Para atingir este objectivo definiram-se os seguintes requisitos:

- Linguagem de Programação
  A Linguagem de Programação escolhida para um projecto deste tipo é importante, embora não seja decisiva. É importante que a Linguagem de Programação escolhida responda às várias necessidades identificadas, nomeadamente:
  - Reutilização da Framework MMX no projecto GUIDE
    Embora a Linguagem de Programação não esteja ainda oficialmente definida para o projecto GUIDE (ou para cada um dos seus módulos) crê-se que a Linguagem de Programação a utilizar seja JAVA.
  - As aplicações devem correr num sistema embebido
    A utilização da Linguagem JAVA requer recursos significativos por parte do dispositivo. No entanto, existem STBs que já suportam soluções com Java e empresas especializadas no tema: Alticast [12], Myriad[13], etc.
  - As aplicações devem respeitar e potenciar as normas definidas para Televisão Digital

Tendo em conta a norma MHP, JAVA é a linguagem que faz mais sentido para aplicações em STBs.

- Publisher Subscriber (Ver ponto abaixo).
  O servidor utilizado JSM é implementado em Java e compatível com clientes Java entre outros.

Para além das razões acima apresentadas existem outras razões que beneficiam a escolha de Java como Linguagem de Programação:

- GoogleTV
  O colosso mundial Google está neste momento a investir num projecto chamado GoogleTV [14]. Este projecto tenta aproveitar o sistema operativo Android [15] e a máquina virtual Java que a Google já utiliza em telemóveis.
- Linguagem de Programação orientada a objectos
  Linguagens de Programação orientadas a objectos no contexto de aplicações que se querem de implementação rápida e fácil fazem todo o sentido. Sendo o Java uma Linguagem de Programação orientada a objectos faz sentido ser utilizada no contexto do presente projecto visto que se pretende disponibilizar uma Framework modular.

Para além de JAVA, outras linguagens de programação foram consideradas com especial destaque para C/C++. Estas linguagens ainda dominam em termos de linguagem usada para sistemas embebidos, e mais especificamente STBs, principalmente devido à optimização de performance que é necessária.

De salientar ainda que, visto estarmos numa arquitectura Publisher/Subscriber, é perfeitamente possível que os diferentes módulos que compõem o sistema possam ser implementados com recurso a Linguagens de Programação diferentes. Isto porque a comunicação entre os módulos está abstraída pela utilização das mensagens do sistema de Publisher/Subscriber.

- Arquitectura Publisher/Subscriber
  A Framework deve ser flexível de forma a permitir que diferentes tipos de componentes possam ser tratados o mais independentemente possível uns dos outros. Um exemplo desta necessidade é pretender-se correr a aplicação numa máquina diferente da máquina onde são corridos os controladores dos dispositivos de entrada/saída. Na Figura 3 está exemplificado o funcionamento típico de um sistema Publisher/Subscriber onde existe um servidor (Topic server) que aceita que os diversos Publishers publiquem mensagens em determinados tópicos (T1, T2, …). Estas mensagens são entregues aos Subscribers que se registaram no servidor para receberem mensagens publicadas em determinados tópicos. O servidor entrega as mensagens a todos

os Subscribers que subscreveram os tópicos onde as mensagens foram publicadas.

Uma arquitectura Publisher/Subscriber tem várias implicações no desenvolvimento de uma Framework ou aplicação:

- Exige que a Framework esteja bem estruturada pois os módulos têm de obedecer à arquitectura já definida (Publisher/Subscriber).
- Exige que os módulos sejam desenvolvidos de acordo com o modo de uso do protocolo do servidor (Publisher/Subscriber).
- Exige um componente adicional que é o próprio servidor Publisher/Subscriber que tem de ser escolhido de acordo com necessidades como o protocolo que utiliza e que clientes suporta.

O servidor Publisher/Subscriber a utilizar deveria ser o mais independente possível de forma a poder ser facilmente substituído. A necessidade de substituição do servidor pode prender-se com diversas razões, como a optimização de recursos ou a utilização de um servidor que já está implantado no ambiente onde se pretende utilizar a Framework.

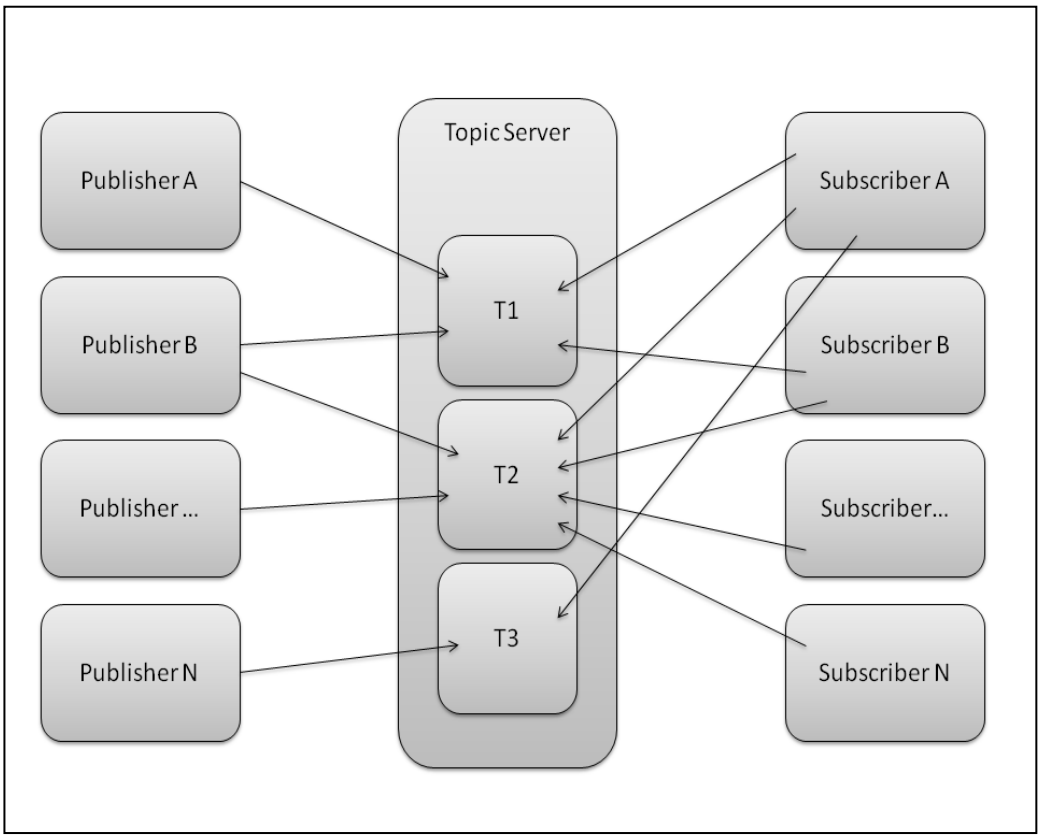

Figura 3: Arquitectura Publisher/Subscriber

Considerando os pontos mencionados, o sistema Publisher/Subscriber escolhido foi o Java Message Service (JMS) [16].

- API bem definida para troca de mensagens

  Sendo o servidor Publisher/Subscriber o módulo central para a interacção entre os vários módulos do sistema, a API que permite a comunicação com este servidor tem de ser bem definida de forma a responder às necessidades de comunicação entre os módulos. Estas necessidades centram-se basicamente na troca (envio e recepção) de mensagens.

  Uma das características que deve ser contemplada é a necessidade de sincronização entre as mensagens, onde é importante existir informação que permita perceber a ordem cronológica das mensagens trocadas. Este requisito será garantido pelo servidor Publisher/Subscriber e não pela API de troca de mensagens, embora ambas as aproximações fossem viáveis. A opção de ser o servidor Publisher/Subscriber responsável pela sincronização das mensagens, é justificada pela existência da funcionalidade no servidor escolhido (JMS), o que permite manter a Framework isolada dessa funcionalidade. Caso se opte pela utilização de um outro servidor Publisher/Subscriber, deve-se ter em conta que, caso esta funcionalidade não exista, tem de ser contemplada na implementação da abstracção para troca de mensagens.
- Categorização das diferentes modalidades

  O foco principal do presente projecto é a utilização simultânea de diferentes modalidades de entrada/saída. As modalidades devem ser distinguidas entre modalidades de entrada e modalidades de saída, e dentro de cada uma destas categorias em outras que façam sentido de acordo com as especificidades de cada modalidade.
- Categorização das acções que podem ser realizadas

  O objectivo último de qualquer evento de entrada (seja ele despoletado por um componente de entrada ou por outra entidade com um papel diferente como por exemplo um temporizador) é fazer com que algo aconteça – uma acção. As acções que podem acontecer dentro do contexto de uma aplicação são limitadas e devem ser bem definidas. Categorizar essas acções permite que um conjunto de acções possa ser partilhado, em termos conceptuais, por diversas aplicações.
- Definição das mensagens consoante a categorização das modalidades e as acções pretendidas

  A comunicação entre os módulos só é possível se estiver bem definida, de forma a que um módulo consiga interpretar correctamente aquilo que outro módulo pretende transmitir-lhe. Esta comunicação é feita através de mensagens trocadas com recurso ao servidor Publisher/Subscriber. São estas mensagens que têm de estar bem definidas e de acordo com a informação que se quer

trocar. A categorização das modalidades e das acções são o primeiro passo para que a definição das mensagens seja correcta.

3.2 Arquitectura da Framework MMX

A Framework MMX está desenhada para que a sua utilização seja fácil e corresponda aos requisitos acima definidos. A arquitectura definida para a versão v0.2 da Framework MMX é apresentada de seguida.

3.2.1 Arquitectura Funcional

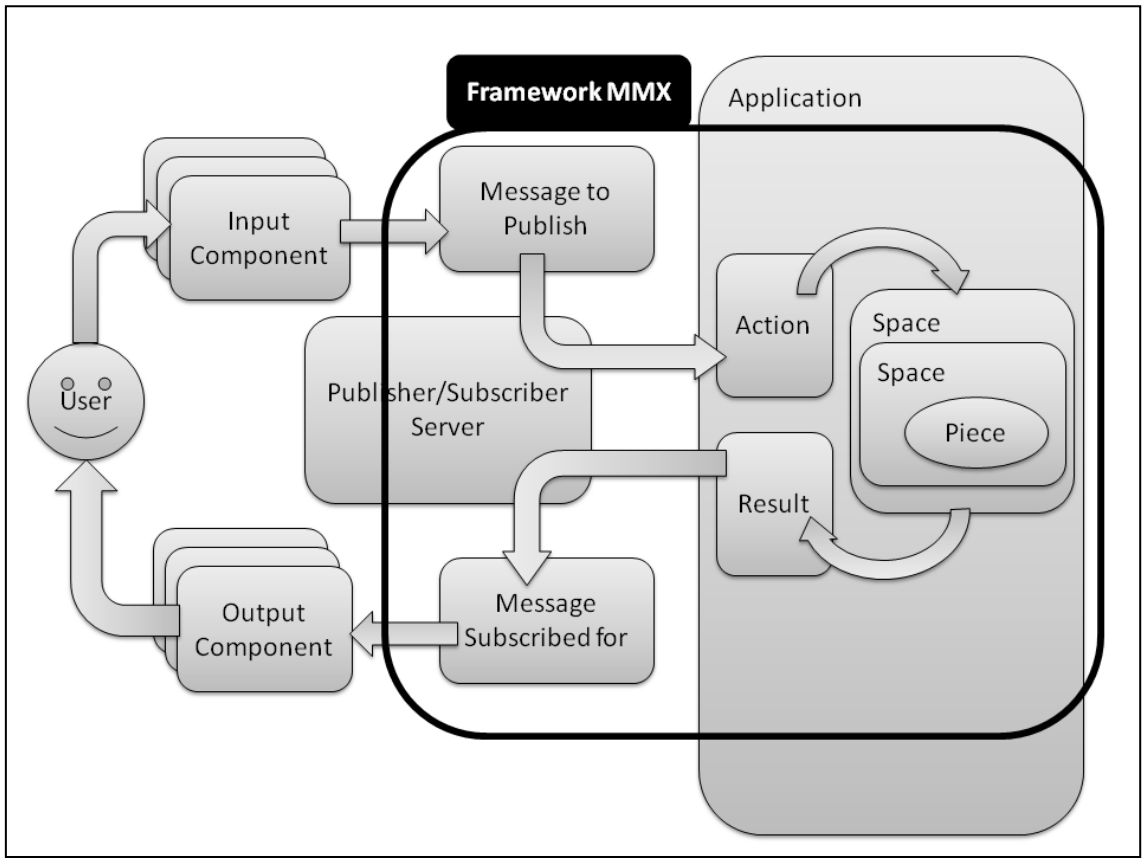

Figura 4: Arquitectura Funcional da Framework MMX

O primeiro passo na definição da arquitectura é obviamente a análise dos requisitos definidos para a Framework. Da análise dos requisitos descritos vários componentes chave são identificados conforme esquematizado na Figura 4.

A Figura 4 representa não só os componentes chave da Framework MMX mas também as ligações que existem entre eles. Segue então uma explicação desses componentes e ligações:

- Utilizador (User): O utilizador do sistema é o actor que interage com a solução desenvolvida. Este utilizador tem objectivos próprios na utilização da solução que expressa através da criação de eventos despoletados com recurso aos

componentes de entrada. Também é ele quem consome as saídas apresentadas pelos componentes de saída.

- Componentes de entrada (Input Components): Os componentes de entrada são o ponto de entrada na interacção do utilizador com a solução. Os componentes de entrada são vários e permitem ao utilizador despoletar eventos (Message to Publish) que serão entregues à Aplicação através de servidor Publisher/Subscriber.
- Aplicação (Application): A aplicação recebe os eventos despoletados pelo utilizador (ou por outra entidade, como por exemplo um temporizador), interpreta-os tendo em conta que estes foram formatados numa Mensagem com um formato bem conhecido pela aplicação e pelos componentes de input, e realiza a Acção que o utilizador pretendia caso esta Acção seja válida; caso a Acção não seja válida é gerado um relatório de erro que será tratado com um resultado (Result). Em seguida, a aplicação despoleta os eventos necessários, caso existam, para dar a conhecer ao utilizador o resultado da acção que foi instruída. Estes eventos são enviados para os componentes de saída através do servidor Publisher/Subscriber.
- Espaço e Peça (Space e Piece): A Framework está definida para trabalhar com aplicações que tenham na sua base conceitos como Espaço e Peça. O conceito de Espaço determina um espaço delimitado por pontos cartesianos que contém um ou mais subespaços não sobrepostos. Cada Espaço pode conter zero ou um elemento (Peça). Estes conceitos são específicos de determinadas aplicações e podem não se adaptar a outras aplicações. No entanto é possível substituir estes conceitos por outros mais adequados a aplicações específicas sem a necessidade de alterar os outros componentes da arquitectura funcional da Framework.
- Componentes de saída (Output Components): Os componentes de saída disponibilizam informação sobre a aplicação ao utilizador. É através destes componentes que o utilizador recebe os eventos despoletados pela aplicação de forma a perceber qual o estado actual da Aplicação.
- Servidor Publisher/Subscriber (Publisher/Subscriber Server): O servidor Publisher/Subscriber garante o correcto envio e entrega entre os componentes que publicam mensagens e os componentes que as subscrevem, recendo-as posteriormente. De notar que na Figura 4 apenas aparecem salientadas as mensagens produzidas pelos componentes de entrada e as consumidas pelos componentes de saída pois pretende-se destacar o papel destes componentes numa Framework que pretende integrar várias modalidades de entrada/saída diferentes.

3.2.2 Arquitectura estrutural

Os componentes acima descritos que serão modelados para serem alvo de implementação são representados no seguinte diagrama de classes (Figura 5):

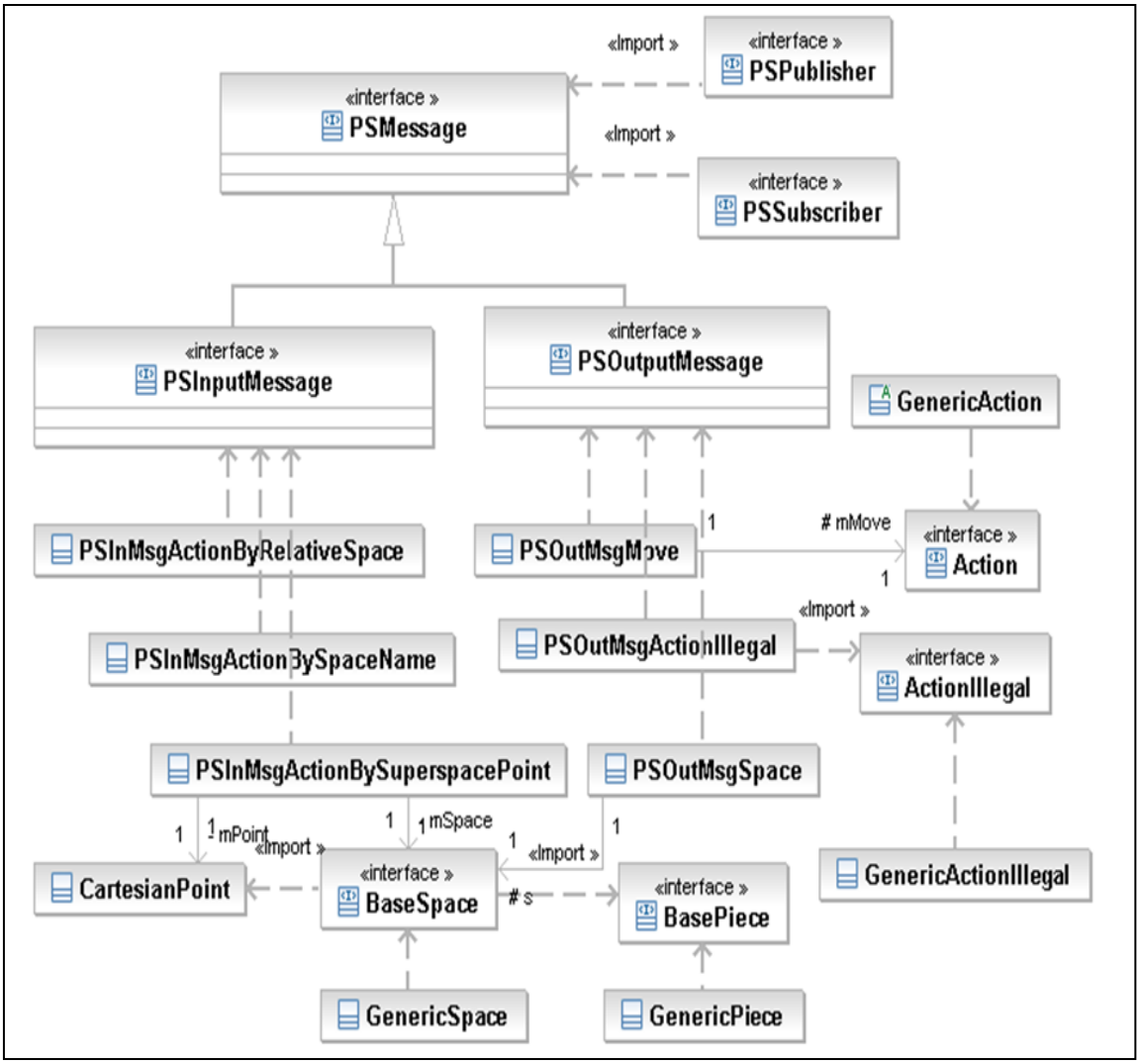

Figura 5: Diagrama de classes da Framework MMX

As Interfaces apresentadas visam criar camadas de abstracção suficientemente claras mas rígidas para que a arquitectura funcional descrita seja respeitada. Para cada uma das Interfaces definidas é apresentada de seguida uma explicação:

- Package pubSub

  Este package contém as Interfaces que permitem abstrair o servidor Publisher/Subscriber que é utilizado. De salientar que o servidor Publisher/Subscriber a utilizar é completamente secundário e que este deve poder ser facilmente substituído conforme as necessidades externas à Framework que o sistema apresente. Apenas duas Interfaces são definidas neste package:

  - PSPublisher

Esta Interface fornece uma abstracção que permite publicar mensagens que serão recepcionadas através da abstracção fornecida pela Interface PSSubscriber.

Esta Interface apenas define um método:

- sendMessage

  Este método permite publicar uma mensagem definida pela Interface PSMessage.

- PSSubscriber

  Esta Interface fornece uma abstracção que permite receber mensagens que foram enviadas através da abstracção fornecida pela Interface PSPublisher.

  Esta Interface apenas define um método:

  - onMessageArrival

    Este método é invocado quando uma nova mensagem PSMessage é publicada.

O facto destas duas Interfaces serem definidas em separado não significa que possa existir um servidor Publisher e um servidor Subscriber visto que isso contraria a definição de servidor Publisher/Subscriber. É assumido que as mensagens Publicadas através da Interface PSPublisher serão entregues a quem as Subscreveu através da Interface PSSubscriber e vice-versa. Portanto, para uma classe que implemente a Interface PSPublisher deve haver uma classe correspondente que implemente a Interface PSSubscriber, onde o conjunto destas duas classes garanta o correcto envio e recepção das mensagens publicadas.

- Package space

  Este package engloba definições relativas a um espaço. Tanto este package como o package piece foram definidos para fazerem face a um conjunto de aplicações que podem fazer uso destes conceitos. No entanto, caso surjam aplicações que pretendam usar a Framework MMX e para as quais outros conceitos façam mais sentido, a Framework é flexível o suficiente para que estes packages sejam substituídos mantendo todos os outros packages em termos de Interfaces.

  Este package apenas define uma Interface:

  - BaseSpace

    Esta Interface define um espaço Cartesiano definido por um ponto de origem e um ponto de término. Visto este Espaço apenas ser definido por uma origem e um término, os Espaços possíveis de definir estão limitados a espaços paralelepipédicos. A opção de

limitar a definição a este tipo de espaços foi apenas por uma questão de simplificação uma vez que, em ambiente de Televisão, as aplicações, por norma, são apresentadas em espaços rectangulares (bidimensionais), sendo que esta Interface permite a utilização de espaços tridimensionais mas que continuam a obedecer à forma rectangular/paralelepipédica.

Para além das fronteiras do Espaço, o Espaço definido por esta Interface obedece a outras propriedades com explicado pelos seguintes métodos exportados por esta Interface:

- addSubSpace; getSubSpace; getSubSpaces
  O Espaço pode ter Zero ou mais SubEspaços, ou seja, espaços que se encontrem dentro da sua fronteira.
- getSuperSpace; setSuperSpace
  O Espaço pode estar contido num espaço maior.
- getElement; setElement
  O Espaço pode conter um elemento (Piece). Neste momento esta propriedade restringe o número de elementos contidos no espaço a Zero ou Um. Esta restrição existe apenas por uma questão de simplicidade e pode facilmente ser alterada caso se identifique essa necessidade.
- select; unselect; isSelected
  O Espaço tem um estado booleano que indica se está seleccionado ou não.
- setSpaceName; getSpaceName
  O Espaço pode ser identificado através de um nome.

- Package piece

  Este package engloba definições relativas a um elemento/objecto/peça. Esta Peça é um objecto que se pode posicionar num Espaço e sobre a qual se podem realizar Acções.

  Tal como o package space, também este package foi definido para fazer face a um conjunto de aplicações que podem fazer uso deste conceito. No entanto, caso surjam aplicações que pretendam usar a Framework MMX e para as quais outros conceitos façam mais sentido, a Framework é flexível o suficiente para que este package seja substituído mantendo todos os outros packages em termos de Interfaces.

  Este package apenas define uma Interface:

  - BasePiece

Esta Interface define uma Peça. Uma peça é basicamente um qualquer objecto que está contido num espaço. Este objecto disponibiliza os seguintes métodos:

- setSpace; getSpace
  Permite associar a Peça a um Espaço e obter o Espaço a que a Peça está associada.
- select; unselect; isSelected
  A Peça tem um estado booleano que indica se está seleccionada ou não.
- getName
  A Peça pode ser identificada através de um nome.

- Package message

  Este package engloba as interfaces que definem as mensagens trocadas através do servidor Publisher/Subscriber. Em termos de definição de métodos e membros, as Interfaces definidas estão vazias uma vez que nenhuma propriedade específica é obrigatória. Isto dá flexibilidade total em termos do conteúdo das mensagens transmitidas. No entanto três Interfaces estão definidas neste package:

  - PSMessage
    Definição de uma mensagem no contexto da arquitectura Publisher/Subscriber.
  - PSOutputMessage
    Definição de uma mensagem de saída no contexto da arquitectura Publisher/Subscriber.
  - PSInputMessage
    Definição de uma mensagem de entrada no contexto da arquitectura Publisher/Subscriber.

  Estas três Interfaces estão definidas apenas para abstrair o que é uma mensagem, sendo que as Interfaces PSOutputMessage e PSInputMessage descendem da Interface PSMessage.

  A existência destas três Interfaces permite também uma fácil adaptação futura caso venha a ser necessário especializar de alguma forma as mensagens.

  Para além das Interfaces foram definidas algumas classes de uso genérico que implementam as Interfaces definidas:

  - PSInMsgActionByRelativeSpace

Mensagem enviada quando se pretende que seja tomada uma acção ao indicar um espaço através da sua localização relativa a outro espaço.

- PSInMsgActionBySpaceName
  Mensagem enviada quando se pretende que seja tomada uma acção ao indicar um espaço identificado pelo nome.
- PSInMsgActionBySuperspacePoint
  Mensagem enviada quando se pretende que seja tomada uma acção ao indicar um ponto cartesiano pertencente a um SuperEspaço.
- PSOutMsgSpace
  Mensagem enviada dando conta da existência de um espaço.
- PSOutMsgActionIllegal
  Mensagem enviada dando conta que a Acção que se tentou realizar foi considerada ilegal.

De notar que as mensagens de entrada aqui definidas (PSInMsgActionByRelativeSpace; PSInMsgActionBySpaceName; PSInMsgActionBySuperspacePoint) não pressupõem a realização de uma acção específica mas sim a realização de uma acção decidida pelo receptor da mensagem conforme o seu estado e os parâmetros recebidos na mensagem.

- Package action
  Este package engloba as Interfaces que definem as acções efectuadas na aplicação (ou aplicações) em consequência da recepção de uma mensagem de entrada. De salientar que a acção real pode ser efectuada em consequência de qualquer outro acontecimento, como, por exemplo, um evento disparado por um temporizador. Este conceito de acção pode ser visto como a intenção do utilizador (ou outro agente) quando despoletou o evento de entrada.
  Este package apenas define duas Interfaces:
  - Action
    Definição de acção conforme explicado acima.
    Esta Interface define cinco métodos:
    - isLegal
      Este método permite saber se a acção em causa é válida.
    - getActionIllegal
      No caso de acção não ser legal este método permitir obter a Acção Inválida.
    - perform

Este método ordena que a Acção seja realizada.

- unperform

  Este método permite desfazer a acção, ou seja, voltar ao estado anterior.

- isPerformed

  Este método permite saber se a Acção já foi executada.

- ActionIllegal

  Definição de uma acção considerada ilegal/inválida.

  Esta Interface disponibiliza dois métodos:

  - getAction

    Este método permite obter a Acção que considerada inválida.

  - getDescription

    Este método permite obter uma descrição da ilegalidade da acção.

Os packages definidos pretendem ser o mais independentes possível, de forma a permitir que a Framework seja estendida e adaptada a outras realidades. Por exemplo, caso se considere que o conceito de Espaço (Space) não se adequa à realidade de uma aplicação, a MMX permite que a Interface Space seja substituída por outra Interface que satisfaça as necessidades; obrigando apenas que outras Classes sejam implementadas (inclusive referentes a outras Packages/Interfaces como é o caso das Mensagens) conforme o grau de interacção/ligação que têm com o package Space.

3.3 Requisitos do protótipo Tabuleiro de Xadrez

O protótipo do tabuleiro de xadrez pretende exercitar a Framework MMX de forma a servir de prova de conceito.

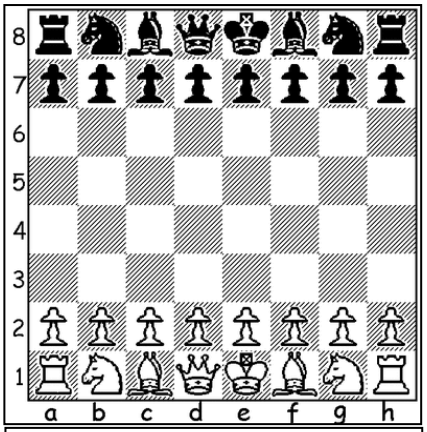

Figura 6: Tabuleiro de Xadrez - disposição inicial

Um tabuleiro de Xadrez obedece a vários requisitos como especificado pelas regras do Jogo de Xadrez. No entanto, o objectivo deste protótipo não é implementar um tabuleiro de xadrez completamente funcional, mas sim implementar um tabuleiro de xadrez que permita exercitar os conceitos definidos pela Framework MMX. Especificidades do jogo de xadrez, como a completa validação das regras, dos casos especiais ou qualquer tipo de inteligência para que o protótipo possa funcionar também como jogador, estão completamente fora do âmbito deste projecto. Nesse sentido o protótipo do tabuleiro de xadrez apresenta os seguintes requisitos:

- Estrutura real de um Tabuleiro de Xadrez
  Apesar do objectivo não ser a completa implementação de um Jogo de Xadrez o protótipo deve apresentar as características básicas de um tabuleiro de xadrez:
  - O tabuleiro divide-se em sessenta e quatro células dispostas numa matriz de oito colunas (de A a H) e oito linhas (de 1 a 8).
  - As células do tabuleiro estão identificadas através de um nome composto pela coluna e a linha correspondentes (de A1 a H8).
  - No tabuleiro estão inicialmente dispostas trinta e duas peças sendo dezasseis da equipa BRANCA e dezasseis da equipa PRETA. Estas peças correspondem às verdadeiras peças de xadrez estando dispostas segundo as regras do xadrez como apresentado na Figura 6.
  - O primeiro jogador a jogar é o jogador da equipa BRANCA.
  - Os Jogadores jogam alternadamente.
  - Uma jogada consiste em mover uma peça.
  - Se um jogador move a sua peça para uma célula ocupada por uma peça da equipa adversária essa peça da equipa adversária é capturada.
  - Jogadas consideradas ilegais:
    - Mover uma peça para uma célula ocupada por uma peça da mesma equipa.
    - A mesma equipa jogar duas vezes seguidas.

  Algumas das características reais de um jogo de xadrez não foram tidas em conta. Assim:
  - As peças podem ser movimentadas sem qualquer restrição relativamente à trajectória ou à distância.
  - Não existe a noção de:
    - Vitória
    - Derrota

        - Empate
        - Xeque
        - Controlo de tempo
    - Jogadas especiais não são contempladas:
        - Roque
        - En passant
        - Promoção do peão

  As características contempladas permitem exercitar vários conceitos definidos pela Framework MMX como por exemplo:

    - Action

      A movimentação de uma peça de xadrez pode ser considerada uma acção. Mais simples ainda: o seleccionar de uma peça para ser movida pode ser considerada uma acção.
    - ActionIllegal

      A quebra de qualquer uma das regras contempladas pode ser considerada uma acção ilegal/inválida. Por exemplo, seleccionar uma peça BRANCA quando era a equipa PRETA a jogar; ou mover uma peça BRANCA para uma célula onde já reside outra peça BRANCA.
    - Space

      O tabuleiro de xadrez pode ser considerado um espaço que contém sessenta e quatro subespaços que representam as células.
    - Piece

      Cada peça de xadrez pode ser considerada uma Peça que reside num espaço (célula). Esta peça pode ser seleccionada para ser movida.

  Para além de permitir exercitar os conceitos definidos pela Framework MMX o facto de o protótipo obedecer em grande parte às características reais de um jogo de xadrez torna a experiência do utilizador mais fácil e intuitiva.

- Compatível com arquitectura da Framework MMX

  Sendo que o principal objectivo do protótipo é exercitar a Framework MMX, este tem de respeitar a arquitectura desta última tendo em conta os vários componentes do sistema:

    - Os componentes de entrada e de saída devem ser disjuntos da aplicação.
    - A comunicação entre os componentes de entrada/saída e a aplicação deve ser assegurada por mensagens trocadas através de um servidor Publisher/Subscriber.

- Exercitar vários componentes de entrada/saída
  Visto ser um dos objectivos principais do projecto a possibilidade de várias modalidades de entrada/saída serem utilizadas em simultâneo o protótipo pretende exercitar o uso das seguintes modalidades:
  - Entrada: Reconhecimento de voz, dispositivos apontadores (rato, seguimento de olhar), Teclado, Texto (linha de comando)
  - Saída: Texto, Imagem

3.4 Arquitectura do protótipo Tabuleiro de Xadrez

O protótipo tabuleiro de xadrez foi desenhado tendo em conta os objectivos e os requisitos definidos para o mesmo.

3.4.1 Arquitectura Funcional

Respeitando a arquitectura da Framework MMX (ver Figura 4) o protótipo tabuleiro de xadrez apresenta a seguinte arquitectura funcional (ver Figura 7):

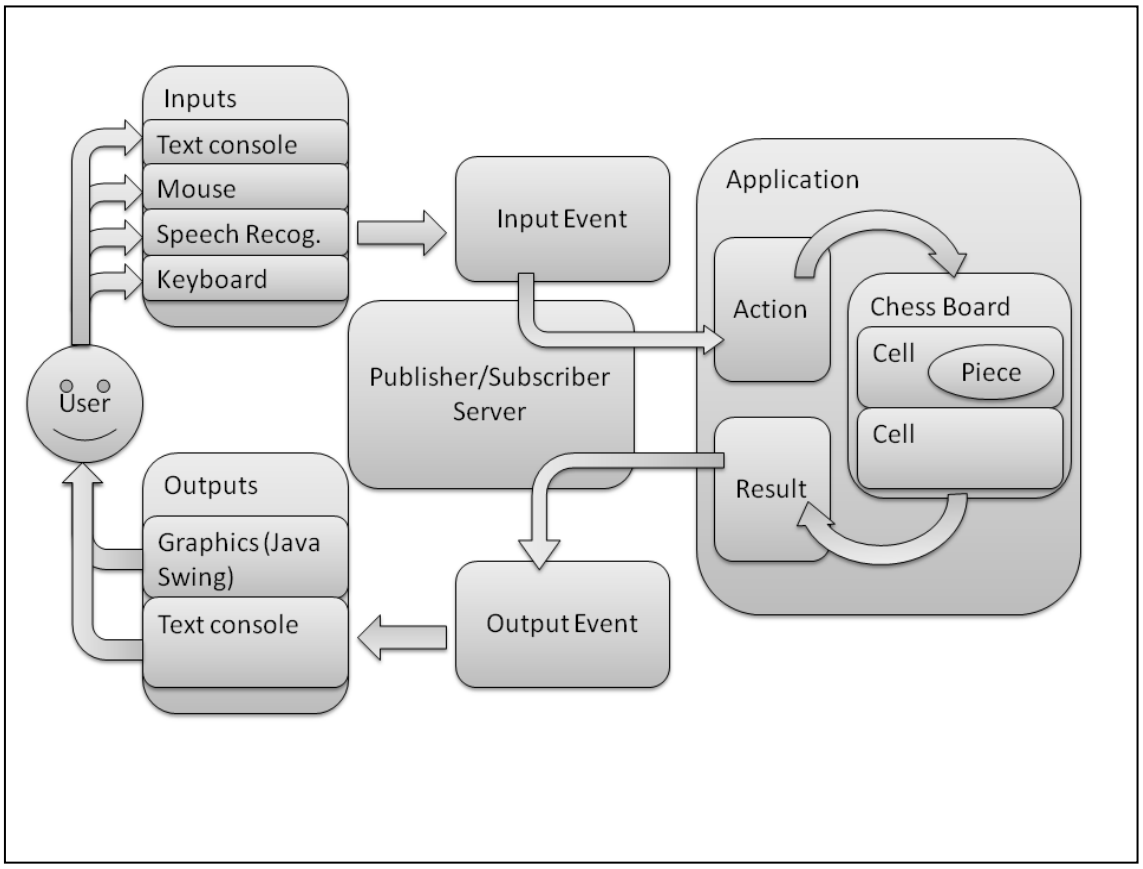

Figura 7: Arquitectura funcional do protótipo tabuleiro de xadrez.

Cruzando a arquitectura da Framework MMX representada na Figura 4 com a arquitectura do protótipo tabuleiro de xadrez representada na Figura 7 consegue perceber-se facilmente as similaridades entre ambas, ou seja, é notória a preocupação

que o protótipo não só respeite a Framework mas, principalmente, que a utilize na integração de várias modalidades de entrada/saída.

A arquitectura do protótipo contempla as modalidades identificadas como requisitos, fazendo uso das Interfaces da Framework conforme explicadas em 3.2.2 acima.

3.4.2 Arquitectura estrutural

Em termos de arquitectura estrutural, o protótipo também respeita e faz uso da arquitectura da Framework, tendo especializado algumas Interfaces para irem de encontro às suas necessidades específicas. As classes definidas especificamente para o protótipo tabuleiro de xadrez são:

- Package mmx.space.chess
  Este package contém a classe correspondente ao espaço tabuleiro de xadrez. Considerou-se também ter uma classe específica para as células do tabuleiro de xadrez mas a Interface BaseSpace definida na Framework era suficiente.
  - ChessBoardSpace
    Esta classe implementa a Interface BaseSpace sendo estendida em alguns métodos necessários às características específicas de um tabuleiro de xadrez como:
    - Noção da última equipa que jogou.
    - Permitir que apenas uma célula (subespaço) esteja seleccionada.
    - O espaço é dividido numa matriz de oito colunas por oito linhas dando origem a sessenta e quatro células.
- Package mmx.piece.chess
  Este package contém as classes que definem as diferentes peças de um jogo de xadrez:
  - ChessPiece
    Esta classe representa uma peça de xadrez, implementando a Interface BasePiece da Framework.
  - BlackBishopPiece até WhiteRookPiece
    Estas classes representam cada uma das Peças do jogo de xadrez, estendendo a implementação da classe ChessPiece apenas ao nível da instância sem acrescentar qualquer novo método.
- Package mmx.action.chess
  Este package contém as classes que correspondem a Acções que podem ser realizadas no tabuleiro de xadrez.

- ChessActionCellSelect; ChessActionCellUnselect
  Estas Acções permitem seleccionar ou desseleccionar uma célula do tabuleiro de xadrez.
- ChessActionPieceSelect; ChessActionPieceUnselect
  Estas Acções permitem seleccionar ou desseleccionar uma Peça do tabuleiro de xadrez.
- ChessActionPieceMove
  Esta Acção permite mover uma Peça no tabuleiro de xadrez.

Para além das classes definidas exclusivamente especializando Interfaces da Framework MMX, também os componentes de entrada/saída seleccionados foram modelados de forma a interagirem com a aplicação como definido pela Framework MMX. Os componentes foram agrupados num package 'chess' onde residem os componentes que funcionaram como aplicações que correm de forma independente. Este pacakage é composto pelos seguintes sub-packages:

- chess.worker
  Este package contém a aplicação Jogo de Xadrez propriamente dita, sendo responsável por receber as mensagens dos componentes de entrada, tratar essas mensagens realizando as acções apropriadas e, caso se justifique, enviar as mensagens resultantes para os componentes de saída. É neste package que se encontra toda a lógica subjacente a um Jogo de Xadrez.
- chess.inputSpeech
  Este package contém a aplicação correspondente ao componente de entrada capaz de reconhecer comandos de voz. Como qualquer componente de entrada esta aplicação é responsável por enviar as mensagens correspondentes aos eventos despoletados pelo utilizador aquando da utilização deste módulo.
- chess.inputConsole
  Este package contém a aplicação correspondente ao componente de entrada capaz de reconhecer comandos de texto escritos numa linha de comando. Como qualquer componente de entrada esta aplicação é responsável por enviar as mensagens correspondentes aos eventos despoletados pelo utilizador aquando da utilização deste módulo.
- chess.inputOutputSwing
  Este package contém a aplicação que utiliza a tecnologia Java.Swing e funciona como componente de entrada – através da utilização do rato e do teclado – e como componente de saída – através da representação gráfica do tabuleiro de xadrez e de mensagens na linha de comando.

Enquanto componente de entrada esta aplicação é responsável por enviar as mensagens correspondentes aos eventos despoletados pelo utilizador aquando da utilização deste módulo.
Enquanto componente de saída esta aplicação deve receber as mensagens enviadas pela aplicação definida no package chess.worker e tratá-las de forma a que a representação gráfica do tabuleiro seja actualizada de acordo, e/ou que na linha de comando seja escrito o texto apropriado, conforme o caso.

Capítulo 4

Implementação e Desenvolvimento

A implementação do protótipo vai de encontro ao definido na arquitectura de forma a atingir os objectivos propostos para a Framework e para o Protótipo Tabuleiro de Xadrez.

A solução foi implementada com recurso à Linguagem de programação JAVA. Esta Linguagem de Programação tem algumas regras implícitas [17] que foram tidas em conta na implementação tanto da Framework como do protótipo Tabuleiro de Xadrez.

O ambiente de desenvolvimento utilizado foi o Eclipse [18] pois possui ferramentas que facilitam a adopção de regras de codificação para Java de forma automática. Para além de utilizar o Eclipse como IDE (Integrated Development Environment), a solução também foi compilada e exercitada com recurso à linha de comando do sistema operativo Microsoft Windows XP[19].

As instruções necessárias para compilar e exercitar a solução podem ser encontradas no sítio internet da Framework MMX [2].

4.1 A Framework MMX

A implementação da Framework teve em conta não só a integração de novas modalidades de entrada/saída mas também toda a arquitectura definida. O principal objectivo da Framework é fornecer aos programadores uma forma facilitada de incorporarem novas modalidades de entrada/saída para a interacção do utilizador com novas aplicações.

Os principal esforço na implementação da Framework prendeu-se com:

- A implementação em Java das interfaces previamente apresentadas na arquitectura.
  Este passo foi basicamente uma transcrição do previamente descrito no diagrama de classes.

- A criação de classes genéricas que implementam algumas das Interfaces definidas. Estas classes foram implementadas pois foram identificados métodos nas Interfaces que teriam um comportamento comum independentemente da utilização futura da Framework.
    - Package mmx.action
        - GenericAction
        - GenericAtionIllegal
    - Package mmx.piece
        - GenericPiece
    - Package mmx.space
        - GenericSpace
    - Pacakge mmx.message
    - Package mmx.pubSub.jms

      Este package implementa duas classes correspondentes às interfaces definidas no package mmx.pubSub, sendo que esta implementação faz uso do servidor Java Message System:
        - JMSPublisher
        - JMSSubscriber

      De notar que a implementação destas classes recorre à utilização de Topics e não de Queues. Esta opção foi tomada para permitir que todos os módulos subscritores recebam todas as mensagens que subscreveram.

Especial cuidado foi necessário na documentação da Framework pois, sendo uma Framework, tem como objectivo ser clara a sua forma de uso. Para atingir este fim as regras para gerar documentação definidas pelo JavaDoc foram seguidas.

4.2 O Protótipo Tabuleiro de Xadrez

A Framework MMX foi desenvolvida em conjunto com o protótipo tabuleiro de xadrez. O objectivo deste protótipo é, não apenas, exercitar a Framework mas, principalmente, ajudar na definição da mesma, pois só com uma aplicação experimental que faça uso da Framework se consegue efectivamente levar a Framework de encontro às necessidades das aplicações reais.

O Protótipo acabou por se dividir em várias aplicações de forma a claramente se exercitar o conceito de sistema distribuído utilizando o servidor Publisher/Subscriber bem como os módulos da Framework que têm subjacente este conceito.

Uma das necessidades inerentes ao protótipo, para além da exercitação da Framework, era que este fosse de fácil utilização por parte do utilizador, para que a interacção fosse intuitiva e permitisse uma fácil percepção dos conceitos definidos. Para atingir este objectivo, optou-se por construir um componente que tire partido das características do Java Swing, tradicionalmente utilizadas para interfaces gráficas com o utilizador, incluindo assim, no mesmo componente, funcionalidades quer de entrada (interacção por rato ou teclado), quer de saída (representação gráfica). Esta opção foi concretizada através da implementação do componente chess.inputOutputSwing. Este componente recorre ao Package java.awt que permite criar interfaces gráficas de utilizador e que se conjuga com dispositivos de entrada comuns em aplicações computacionais como o rato e o teclado. Em suma, este componente chess.inputOutputSwing faz uso das modalidades de entrada/saída de maior utilização em aplicações computacionais: interface gráfica; rato e teclado; permitindo uma utilização intuitiva do protótipo em causa.

Em conjunto com o componente de entrada e saída chess.inputOutputSwing, desenvolveu-se a aplicação que funciona como núcleo do protótipo em termos de funcionalidades relacionadas com Xadrez: chess.worker. Esta aplicação tem a capacidade de validar as acções e de as efectuar de forma a actualizar o estado do tabuleiro de Xadrez.

A aplicação chess.worker faz uso das mensagens definidas pela Framework, não estendendo estas mensagens para mensagens específicas desta aplicação. Embora a Framework permita que as mensagens sejam estendidas conforme as necessidades das aplicações, a opção de não estender as mensagens permite utilizar as mensagens já definidas de forma a aferir se estas são de factos úteis. Visto as mensagens utilizadas serem muito simples, alguma inteligência tem de ser implementada do lado a aplicação. No caso do tabuleiro de Xadrez esta inteligência passa por interpretar a mensagem recebida e definir qual a acção a efectuar, não só com base na mensagem, mas também tendo em conta o estado do tabuleiro. Por exemplo, se o tabuleiro tiver uma peça seleccionada e uma célula de destino também, quando é recebida uma mensagem mmx.message.PSInMsgActionBySuperspacePoint relativa à célula de destino, a aplicação concluí que deve mover a peça da célula actual para a célula de destino após a validação da legalidade dessa acção. Este cenário é mostrado na Figura 8.

Uma vez assegurada tanto a interacção básica com o protótipo através do componente chess.inputOutputSwing, como a funcionalidade típica do tabuleiro de Xadrez, avançou-se para outro objectivo que consistiu em adicionar mais componentes de entrada/saída ao protótipo com o objectivo de exercitar a harmoniosa co-existência destes. Foram então criados os componentes chess.inputConsole e chess.inputSpeech.

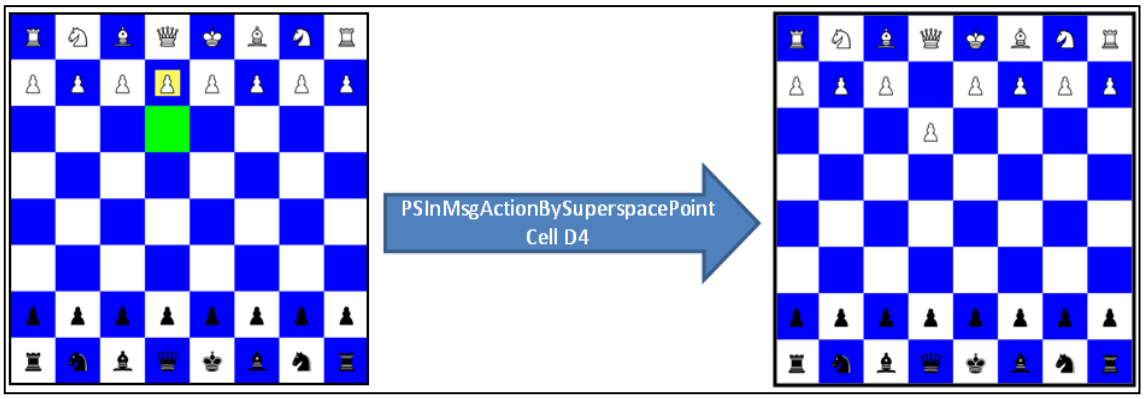

Figura 8: Tabuleiro de Xadrez - Acção baseada no estado do tabuleiro.

Estes componentes de entrada utilizam respectivamente a escrita e a fala. Para a interpretação da escrita e/ou da fala é necessário definir uma gramática para ser utilizada para reconhecer os comandos entrados pelo utilizador. A gramática definida para ambos os componentes incluía comandos simples, relativos às células, de forma a utilizar as mensagens previamente definidas. Esta gramática incluía o nome das células (de a1 a h8) e selecção relativa das células (cima, baixo, direita, esquerda, …). Optou-se por não se implementar uma gramática mais avançada que incluísse ordens exclusivas do Xadrez como "Mover a1 para a2". Esta opção foi tomada para manter toda a funcionalidade já existente, utilizando simultaneamente diferentes modalidades de entrada mas sem alterar a definição das mensagens anteriormente definidas. Caso se optasse por se implementar uma gramática mais avançada seria necessário definir novas mensagens que conteriam a informação necessária para que a aplicação pudesse entender a acção a realizar.

O componente chess.inputConsole foi implementado com recurso a java.util.Scanner enquanto o componente chess.inputSpeech foi implementado com recurso ao reconhecedor de voz Sphinx [20].

4.3 Protótipos de Validação

De forma a validar a Framework MMX, uma equipa externa ao desenvolvimento da Framework desenvolveu dois protótipos: O Jogo do Galo e a Aplicação EPG.

4.3.1 O protótipo Jogo do Galo

Implementada a Framework MMX v0.1 e o protótipo tabuleiro de Xadrez houve a preocupação de garantir que a Framework e o protótipo tabuleiro de Xadrez estavam isolados entre si, tanto em termos de definição como de implementação. Para testar e assegurar a independência entre Framework e protótipo, optou-se pela implementação de um outro protótipo, Jogo do Galo, que deveria ser implementado fazendo uso da

Framework mas completamente independente do protótipo Tabuleiro de Xadrez, socorrendo-se deste apenas a título exemplificativo.

A implementação do protótipo Jogo do Galo foi levada a cabo por uma equipa que não tinha tido contacto prévio com a Framework MMX, com vários objectivos:

- Ajudar a isolar a Framework MMX do protótipo Tabuleiro de Xadrez e de outros protótipos/soluções subsequentes.
- Analisar as limitações/dificuldades e as mais-valias apresentadas pela Framework a vários níveis:
    - Adicionar diferentes componentes de entrada
    - Adicionar diferentes mensagens
    - Adicionar diferentes acções

As conclusões apresentadas pela equipa que desenvolveu o protótipo Jogo do Galo, apresentadas em mais detalhe no capítulo seguinte, foram tidas em conta na disponibilização de uma nova versão da Framework (v0.2).

Esta nova versão apresenta uma clara distinção em termos de definição e implementação entre a Framework e os protótipos que a utilizam – implementados em packages Java diferentes. Foi adicionado à Framework MMX o protótipo Jogo do Galo a título de exemplo, da mesma forma que o protótipo tabuleiro de Xadrez também era disponibilizado conjuntamente com a Framework. Exercitaram-se com sucesso a adição de componentes de saída para sons (tictactoe.outputSound) e para síntese de voz (tictactoe.outputVoice). Foi também definida uma nova mensagem de saída (mmx.message.PSOutMsgInfo) que permite enviar para os componentes de saída uma mensagem com informação genérica em forma de texto. Esta nova mensagem foi necessária para enviar para os componentes de saída a informação de qual o vencedor do jogo, sendo que a única alteração à Framework foi a implementação da classe relativa à mensagem.

4.3.2 Protótipo EPG

Com o objectivo de assegurar que a Framework MMX v0.2 está desenvolvida de forma a ser utilizada em aplicações bastante distintas de Jogos de Tabuleiro, a equipa que desenvolveu o protótipo Jogo do Galo desenvolveu uma aplicação que, utilizando a Framework, pretende desenvolver uma solução que apresente a informação relativa aos Programas que serão apresentados num certo número de canais durante um certo período de tempo. Este tipo de informação é conhecida com Guia TV ou EPG – Electronic Program Guide. As conclusões derivadas da implementação serão apresentadas no capítulo de resultados.

4.4 Modalidades de Entrada/Saída Utilizadas

O âmbito do projecto não abrangia o desenvolvimento de qualquer modalidade de entrada/saída mas sim a utilização conjunta de várias modalidades de entrada/saída já existentes. Parte do trabalho deste projecto passou por analisar várias modalidades existentes, tendo sido escolhidas aquelas que são facilmente integráveis em aplicações JAVA e que ao mesmo tempo estejam disponíveis sob licenças que permitam o seu uso sem qualquer custo.

Para além das especificidades técnico-legais já referidas, é importante que as modalidades escolhidas sejam abrangentes e distintas para que um considerável número de diferentes formas de interacção seja exercitado.

As modalidades escolhidas:

- Entrada
    - Rato
      Exercitado através da utilização de java.awt.event.MouseListener
    - Teclado
      Exercitado através java.awt.event.KeyListener
    - Reconhecimento de Voz
      Exercitado com recurso ao reconhecedor de voz CMU SPHINX [20].
    - Interpretação de Escrita
      Exercitado com recurso a java.util.Scanner
    - Seguimento de olhar
      Exercitado com recurso a LEA - Lightweight Eyetracking Algorithm.
      Este módulo foi abandonado pois os testes com ele realizados não produziram os resultados mínimos necessários para uma integração bem sucedida.
- Saída
    - Visualização Gráfica
      Exercitado através de javax.swing
    - Síntese de Voz
      Exercitado com recurso a FreeTTS [21]
    - Sons
      Exercitado com recurso a javax.sound

4.5 Disponibilização da Framework

A Framework MMX pretende ajudar a integração de várias modalidades de entrada/saída para a interacção com aplicações que corram em sistemas computacionais, como dispositivos de Televisão Digital.

Uma vez que existe uma multiplicidade de modalidades de entrada/saída e uma multiplicidade de aplicações que podem fazer uso delas, os casos de uso da Framework são inúmeros, o que se pode concluir também pelo aumento dos protótipos/aplicações desenvolvidas bem como das várias modalidades exercitadas no decorrer deste projecto.

Para facilitar o controlo do que está a ser desenvolvido em torno da Framework MMX, bem como o próprio desenvolvimento desta, foi criado um sítio no Sourceforge.net [2]. O sítio pretende disponibilizar algumas ferramentas importantes no trabalho levado a cabo por diversas pessoas no desenvolvimento de um mesmo projecto de Software. Destas ferramentas destacam-se o SVN, o sistema de controlo de erros (bug tracking system) e o fórum/wiki.

Capítulo 5

Resultados

O resultado principal deste trabalho foi a Framework MMX. Esta Framework disponibiliza um conjunto de interfaces e classes que visam permitir a fácil integração de diversas modalidades de entrada/saída no desenvolvimento de aplicações multimodais para sistemas computacionais. Para a utilização e consulta da Framework MMX existe um sítio internet [2] que disponibiliza o código fonte, a documentação e os manuais necessários. Este sítio disponibiliza não só a Framework MMX, mas também elementos relativos aos protótipos desenvolvidos para exercitarem a Framework: Tabuleiro de Xadrez, Jogo do Galo e Aplicação EPG.

5.1 Framework MMX

A Framework MMX apresenta uma série de interfaces que foram exercitadas, analisadas e melhoradas com recurso aos três protótipos. A Framework teve até ao momento duas versões: 0.1 e 0.2.

A versão 0.1 da Framework resultou do desenvolvimento conjunto do protótipo Tabuleiro de Xadrez e da própria Framework MMX. Através da análise dos requisitos do Tabuleiro de Xadrez em conjunto com os requisitos da Framework MMX surgiu esta primeira versão 0.1 que foi disponibilizada a uma outra equipa de desenvolvimento para ser utilizada no desenvolvimento do segundo protótipo, o Jogo do Galo. Esta versão 0.1 apresentava já bastantes conceitos que iam de encontro aos objectivos propostos, como a abstracção para o sistema Publisher/Subscriber, as mensagens básicas definidas (package mmx.message) bem como as noções de espaço e peça. Para além destas noções gerais, também as noções específicas relativas ao Tabuleiro de Xadrez estavam definidas.

O passo seguinte passou por permitir que a Framework fosse utilizada por uma equipa de desenvolvimento diferente que não tivesse os vícios adquiridos pelo desenvolvimento da própria Framework e pudesse fazer uma avaliação mais independente sobre a usabilidade e adequação da Framework para os fins previstos.

Nesse sentido, a equipa de desenvolvimento seleccionada para o efeito procedeu ao desenvolvimento do protótipo Jogo do Galo.

O desenvolvimento do protótipo Jogo do Galo teve como objectivo principal avaliar a usabilidade da Framework MMX. A equipa de desenvolvimento do Jogo do Galo seguiu as instruções disponibilizadas em termos de instalação das ferramentas e aplicações necessárias e baseou-se no Tabuleiro de Xadrez para o desenvolvimento das classes que dariam origem ao Jogo do Galo. Como resultado deste desenvolvimento, surgiram algumas ideias importantes para melhoria da Framework MMX. Estas ideias deram origem à versão 0.2 e prendiam-se com:

- A Framework não estava bem documentada. Na verdade não existia um desenho da Framework como o apresentado no Capítulo 3 deste relatório. Também a documentação do código era minimalista e não obedecia a qualquer regra. Tendo em conta esta lacuna procedeu-se à definição de regras de documentação do código e ao início da sua utilização.
- Não existia a noção de Acção, ou seja, o que era feito aquando da recepção de uma mensagem de entrada pelas aplicações nucleares (Tabuleiro de Xadrez, Jogo do Galo, …) não estava devidamente definido, sendo que o que existia estava muito directamente ligado ao Tabuleiro de Xadrez, representando acções típicas de um jogo de Xadrez, mas que mesmo assim não estavam bem isoladas e definidas. Para fazer face a esta situação foi criada uma nova camada de abstracção que isola as acções (package Action).
- A Framework não estava completamente isolada do protótipo Tabuleiro de Xadrez. Existiam classes e definições que se confundiam entre Framework e protótipo. A solução passou por definir mais isoladamente as classes e as interfaces, colocando-as em pacotes diferentes (pacakge mmx e package chess). Desta forma qualquer protótipo pode ser criado dentro do seu pacote, utilizando a Framework sem o risco de se confundir o que é Framework e o que é protótipo.

Com as alterações, resultantes da exercitação da Framework MMX através do desenvolvimento do protótipo Jogo do Galo, reflectidas na versão 0.2 da Framework MMX, procedeu-se à definição de um novo protótipo que pudesse exercitar a Framework MMX num ambiente mais ligado à Televisão e com requisitos diferentes dos jogos de tabuleiro até aqui utilizados. Este novo protótipo seria então uma aplicação EPG que utilizaria dados com um formato normalizado [22]. Para além da definição básica da aplicação EPG, foi também definida a forma como a Framework poderia ser utilizada para atingir o fim pretendido: uma aplicação EPG que utiliza dados

normalizados e que faça uso da Framework MMX de forma a permitir a interacção multimodal com o utilizador. Da definição da aplicação EPG e da forma como deveria fazer uso da Framework MMX, concluiu-se que a Framework MMX estaria à altura do desafio num plano teórico, mas apenas a implementação da aplicação EPG poderia dissipar as dúvidas quanto à qualidade da Framework. A versão 0.2 da Framework MMX foi então disponibilizada juntamente com a descrição da aplicação EPG de forma a que a equipa que desenvolveu o protótipo Jogo do Galo pudesse proceder à implementação da Aplicação EPG. A implementação da Aplicação EPG decorreu de acordo com o planeado atingindo os objectivos pretendidos.

5.2 O protótipo Tabuleiro de Xadrez

O protótipo Tabuleiro de Xadrez foi desenvolvido em conjunto com a Framework MMX. Durante largo tempo, o desenvolvimento de ambos confundiu-se tendo muitas vezes sido pouco claro quais as classes pertencentes ao protótipo e quais pertencentes à Framework. Com o decorrer do desenvolvimento, esta distinção foi-se consumando atingindo-se um ponto em que as classes e interfaces de cada um teriam o seu próprio pacote, ficando clara a fronteira entre protótipos/aplicações e Framework.

Fazendo uso da Framework MMX, o protótipo tabuleiro de Xadrez funciona de forma a corresponder aos objectivos traçados. Não contempla uma série de características de um jogo real de xadrez, tais como as jogadas especiais (roque, en passant, promoção, …), nem obedece à maioria das regras de movimentação no tabuleiro, mas permite deslocar as peças e validar algumas regras de forma a exercitar todos os módulos da Framework MMX.

Em termos de implementação, o protótipo divide-se em três partes principais:

- A aplicação nuclear (chess.worker)
- A especialização das interfaces/classes definidas na Framework (chess.mmx.*)
- Os módulos correspondentes às modalidades de entrada/saída (chess.input*; chess.output*; chess.inputOutput*)

O protótipo apresenta a mesma funcionalidade a nível da aplicação nuclear, independentemente das modalidades utilizadas ou da quantidade de modalidades utilizadas ao mesmo tempo. As modalidades para que foram implementados módulos de forma a serem suportadas pelo protótipo foram: Consola de texto (chess.inputConsole); Reconhecedor de voz (chess.inputSpeech); e apresentação gráfica (chess.inputOutputSwing).

Uma imagem da representação gráfica da implementação do Tabuleiro de Xadrez é apresentada na Figura 10.

Figura 9: Representação Gráfica do Tabuleiro de Xadrez

5.3 O protótipo Jogo do Galo

O protótipo Jogo do Galo foi desenvolvido por uma equipa externa ao projecto com o objectivo de perceber a qualidade da Framework MMX em termos de implementação de novas aplicações. Este protótipo ajudou à melhor definição da Framework MMX. Após a alteração da Framework MMX para colmatar as lacunas identificadas aquando do desenvolvimento do protótipo Jogo do Galo, procedeu-se à adaptação do protótipo Jogo do Galo à nova versão 0.2 da Framework MMX.

Em termos de implementação também este protótipo se divide nas mesmas três partes principais que o protótipo Tabuleiro de Xadrez, tendo sido implementados módulos para suportar as seguintes modalidades: Apresentação Gráfica (tictactoe.inputOutputSwing), Sons (tictactoe.outputSound), Síntese de voz (tictactoe.outputVoice).

O protótipo Jogo do Galo foi implementado no mesmo ambiente de desenvolvimento que a Framework MMX, tendo sido testado num computador convencional e não numa STB.

A implementação do Jogo do Galo reaproveitou grande parte do desenvolvimento feito para o Tabuleiro de Xadrez com as seguintes conclusões:

- Seguindo o que tinha sido implementado no Tabuleiro de Xadrez, procedeu-se à implementação de uma classe que representa o tabuleiro de Jogo. Esta classe (tictactoe.mmx.space.TictactoeBoardSpace) estende a classe mmx.space.GenericSpace e implementa a interfacemmx.space.BaseSpace.
- Como modalidade de entrada e de saída, implementou-se a classe tictactoe.inputOutputSwing, que foi copiada da classe chess.inputOutputSwing e adaptada de acordo com pequenas modificações.
- O motor do Jogo do Galo foi implementado isolado no package tictactoe.worker.
- As acções não estavam isoladas pelo que as acções definidas para o Tabuleiro de Xadrez foram copiadas para o Jogo do Galo com as devidas adaptações. A versão 0.2 da Framework MMX alterou esta situação através da definição de Acções genéricas (mmx.action).
- As peças necessárias para o Jogo do Galo foram criadas partindo da classe mmx.piece.BasePiece e da interface mmx.piece.GenericPiece.
- Foram criados dois novos módulos de saída para reproduzir sons (tictactoe.outputSound) e sintetizar voz (tictactoe.ouptutVoice).
- A Framework MMX não estava documentada o que causou dificuldades na sua utilização.

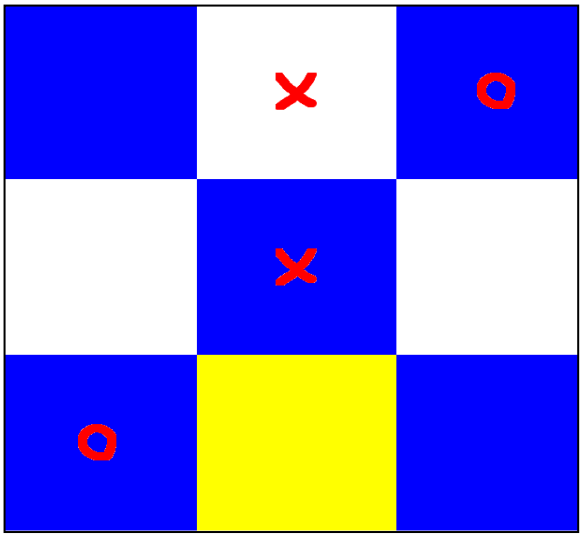

Figura 10: Representação gráfica do Jogo do Galo.

Fazendo uso da Framework MMX o protótipo Jogo do Galo está bastante completo obedecendo às regras de um verdadeiro jogo do galo, validando inclusive quando existe

um vencedor. Uma imagem da representação gráfica da implementação do Jogo do Galo é apresentada na Figura 11.

5.4 A aplicação EPG

A aplicação EPG permitiu exercitar a Framework na implementação de uma aplicação direccionada para Televisão. A implementação fez uso da versão 0.2 da Framework MMX seguindo a mesma aproximação que foi utilizada para o protótipo Jogo do Galo, permitindo concluir que, nesta versão, efectivamente existe uma clara separação entre a Framework MMX e as aplicações que a utilizam.

A implementação da aplicação EPG manteve a estrutura dos protótipos anteriormente descritos, dividindo-se em três partes:

- A aplicação nuclear onde está a lógica aplicacional relacionada com a navegação e a disponibilização de informação relativa aos canais e aos programas.
- A especialização das classes e interfaces da Framework MMX. Desta especialização salientam-se:
    - A definição de uma nova mensagem que permite à aplicação enviar a informação relativa a um programa (xmltvepg.mmx.action. XmltvepgActionGetInfo)
    - A definição de duas novas mensagens que permitem aos componentes de entrada pedir informação sobre um determinado programa (PSInMsgActionByCalendar e PSInMsgActionByHour)
    - A definição de duas Peças: Programa e Canal.
- Os componentes de entrada e saída. A aplicação foi exercitada com recurso aos seguintes componentes:
    - InputConsole que permite, através da linha de comandos escritos, navegar na grelha de programação e pedir informação sobre um programa específico.
    - InputOutputSwing que disponibiliza a representação gráfica da grelha de programação e permite utilizar o rato como dispositivo de entrada.

Visto a Aplicação EPG ser substancialmente diferente dos outros dois protótipos anteriormente implementados (Tabuleiro de Xadrez e Jogo do Galo), pois estes eram ambos jogos de tabuleiro, enquanto a Aplicação EPG não é, sobressaíram alguns detalhes importantes na forma como a Framework MMX está definida. O mais importante prende-se com a definição de Espaço, que devido à forma como é apresentada, denota uma forte ligação a espaço físico. Esta ligação é mais evidente

devido à utilização de pontos cartesianos na definição dos Espaços. De salientar que o Espaço funciona principalmente como um agregador que permite ser subdividido em sub-espaços e permite conter um elemento (Peça). O facto de os espaços serem definidos por pontos cartesianos deve-se à Framework MMX ter sido inicialmente idealizada para aplicações baseadas em televisão onde a base de apresentação da aplicação é principalmente gráfica.

Esta forte ligação entre a representação gráfica e a definição do Espaço como entidade básica para o desenvolvimento das aplicações que façam uso da Framework MMX traz uma limitação a aplicações que pretendam utilizar modalidades de saída que não contemplem a representação gráfica. Seria interessante isolar a representação gráfica dos Espaços enquanto entidades básicas e especializá-las apenas quando se justificasse. Por exemplo, numa aplicação que pretenda apenas utilizar modalidades de saída não gráficas, como sintetizar em voz na leitura de um livro, era interessante não usar Espaços cartesianos mas sim Espaços genéricos que funcionariam como agregador de sub-espaços ou peças (como capítulos ou parágrafos, por exemplo).

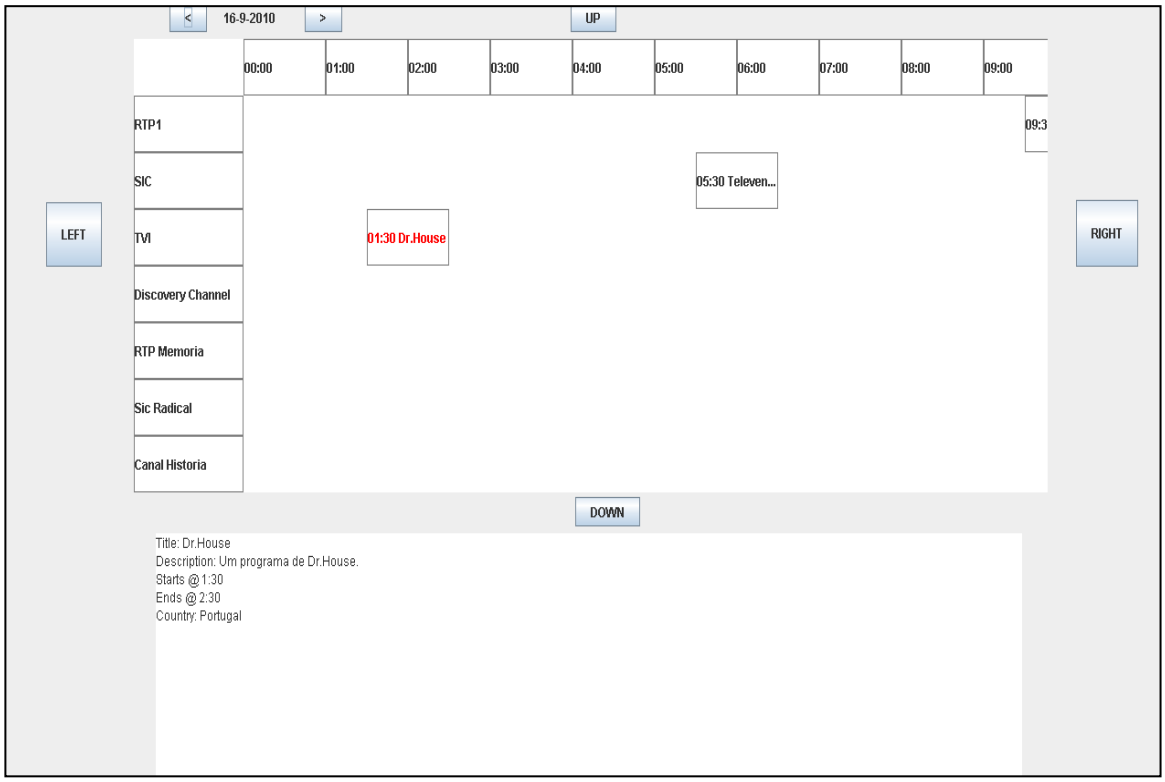

Figura 11: Representação Gráfica da implementação da Aplicação EPG

Na Figura 12 é apresentada a representação gráfica da Aplicação EPG donde se salienta os botões de navegação e as áreas de apresentação de informação desenhadas de forma a adaptar-se a um ecrã de Televisão. Esta apresentação gráfica foi alterada durante o desenvolvimento da Aplicação EPG – inicialmente a representação gráfica do EPG apresentava barras de navegação e tinha uma dimensão reduzida sendo, por isso,

não adequada para um ambiente de televisão. De salientar que a alteração do grafismo do EPG em nada alterou a utilização da Framework MMX pois os aspectos gráficos estão isolados no respectivo componente de saída.

Capítulo 6

Conclusões

Este trabalho teve como objectivo o desenvolvimento da Framework MMX, que serve de base para o desenvolvimento de aplicações multimodais. A Framework MMX assenta num arquitectura Publisher/Subscriber, permitindo que várias modalidades de entrada/saída sejam utilizadas em simultâneo para interacção com aplicações computacionais. A Framework MMX é composta por diversas abstracções que permitem abstrair qual o servidor Publisher/Subscriber em uso, bem como encapsular os eventos de entrada e saída em mensagens pré-definidas.

A Framework foi desenvolvida em conjunto com um protótipo de um Tabuleiro de Xadrez que ajudou à definição da Framework. O protótipo Tabuleiro de Xadrez permite a interacção em simultâneo de várias modalidades de entrada/saída numa arquitectura Publisher/Subscriber com recurso à Plataforma MMX. Do desenvolvimento do protótipo de Xadrez resultou a versão v0.1 da Framework MMX.

Durante o desenvolvimento da Framework MMX foi ainda desenvolvido um segundo protótipo – Jogo do Galo – que deveria fazer uso da Framework MMX. Este protótipo foi desenvolvido por uma equipa que não esteve envolvida no desenvolvimento da Framework MMX e teve com objectivo avaliar e posteriormente melhorar a Framework. Da avaliação conseguida através do desenvolvimento do protótipo Jogo do Galo resultou a versão v0.2 da Framework MMX. Versão esta que apresentava a integração de mais modalidades e uma muito mais clara distinção entre Framework MMX e protótipos/aplicações.

Com o intuito de exercitar a Framework numa aplicação que diferisse dos protótipos já desenvolvidos – Tabuleiro de Xadrez e Jogo do Galo – avançou-se para a definição e implementação de uma Aplicação EPG. Este terceiro protótipo foi desenvolvido pela equipa que desenvolveu o protótipo Jogo do Galo e permitiu concluir que a Framework MMX é adequada para a implementação de aplicações multimodais diversas como Tabuleiro de Xadrez, Jogo do Galo ou Aplicação EPG.

De salientar que a utilização da Framework MMX para integrar simultaneamente as diversas modalidades na Aplicação EPG mostrou-se bastante fácil e bem estruturada sendo possível incorporar facilmente modalidades como a síntese de voz, sendo para isso necessário seguir o exemplo de síntese de voz utilizado no protótipo Jogo do Galo.

6.1 Trabalho futuro

Este projecto teve como objectivo principal o desenvolvimento da Framework MMX. Esse objectivo foi atingido permitindo o desenvolvimento do protótipo Tabuleiro de Xadrez, o desenvolvimento do protótipo Jogo do Galo bem como o desenvolvimento de uma aplicação EPG multimodal.

Os resultados do trabalho desenvolvido até aqui abrem portas para não só para o desenvolvimento de novas aplicações multimodais mas também para o aperfeiçoamento dos componentes já desenvolvidos. Apresentam-se de seguida algumas ideias que poderão fazer sentido serem exploradas no seguimento deste projecto.

Em primeiro lugar era interessante corrigir uma das principais lacunas da Framework MMX: a falta de documentação. Para atingir este fim já foi definida a metodologia a seguir para gerar a documentação recorrendo à ferramenta javadoc.

Para além da documentação era importante exercitar a Framework MMX num ambiente distribuído do ponto de vista físico, ou seja, em que uma máquina funcionava como servidor Publisher/Subscriber e outra máquina diferente corria as aplicações desenvolvidas na Framework MMX. Neste sistema Publisher/Subscriber era também interessante implementar classes no pacote mmx.pubsub que utilizassem um sistema diferente do JMS e que utilizassem uma forma não automática de serialização das mensagens que são trocadas. Este cenário é tão mais importante quando se pretende que os diversos módulos que interagem com a Framework MMX estejam codificados em linguagens de programação diferentes. Por exemplo, se o módulo que publica uma mensagem e o módulo que a subscreve estiverem implementados em linguagens de programação diferentes, a abstracção mmx.pubSub deve ser implementada de forma a que o conteúdo da mensagem seja percebido tanto por quem publica, como por quem recebe. A forma de normalizar o conteúdo das mensagens trocadas pode ser através da utilização de uma estrutura bem definida usando XML. Note-se que, caso as mensagens definidas sejam as suficientes, esta utilização de XML pode ser encapsulada completamente no package mmx.pubsub, reutilizando completamente as aplicações já desenvolvidas sobre a Framework MMX.

Existem outros cenários onde seria importante testar a Framework MMX e, eventualmente, adaptá-la a esses cenários. Esses cenários passam por ambientes multi-tarefa e por sistemas embebidos que ficaram de fora dos cenários de testes efectuados,

sendo admissível que a implementação actual da Framework não corresponda às necessidades.

A Framework MMX apresenta ainda um baixo número de mensagens definidas. Era importante definir novas mensagens que permitissem ordens mais elaboradas como, por exemplo, ser possível utilizar uma modalidade de entrada baseada em texto ou voz para transmitir uma mensagem que instrua a aplicação a mover uma peça de um espaço para outro: “Move from a1 to b4”. As mensagens tipificadas poderiam também ser agrupadas, por exemplo em mensagens que estão relacionadas com espaços (mmx.space).

Para além da tipificação das mensagens era importante tipificar as modalidades de entrada/saída para além da divisão óbvia entre entrada e saída. Esta tipificação passaria por, por exemplo, agrupar todos as modalidades de entrada que funcionam como dispositivos apontadores (rato, wiimote) e/ou todas as modalidades de entrada que utilizam uma gramática (escrita, voz) de forma a que a implementação do suporte destas modalidades fosse reaproveitada. Esta tipificação das modalidades permitiria ainda a criação de tópicos diferentes no servidor Publisher/Subscriber de forma quase imediata, visto que neste momento apenas existem dois tópicos em uso: um para as modalidades de entrada e outra para as modalidades de saída; pois foi esta a única tipificação feita a nível de modalidades.

Uma outra extensão à Framework MMX passaria pela definição de um validador de acções genérico, que serviria de base ao desenvolvimento de validadores de acções específicos. Neste momento a validação de cada acção identificada é feita no contexto da própria acção.

Capítulo 7 Bibliografia

1. **GUIDE Consortium.** GUIDE. *Gentle User Interfaces for Disabled and Elderly Citizens.* [Online] http://www.guide-project.eu/.

2. **Cardoso, Alexandre.** FrameworkMMX. *SourceForge.* [Online] http://frameworkmmx.sourceforge.net/.

3. **Wikipedia.** Embedded_system. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Embedded_system.

4. —. Digital_television_transition. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Digital_television_transition.

5. **DVB.** DVB. *DVB.* [Online] http://www.dvb.org/.

6. **MHP.** *MHP.* [Online] http://www.mhp.org/.

7. **Sun.** JAVA. *JAVA.* [Online] http://www.java.com/.

8. **MicroSoft.** NATAL. [Online] http://www.microsoft.com/uk/wave/hardware-projectnatal.aspx.

9. **Wikipedia.** Deep_Blue. [Online] http://en.wikipedia.org/wiki/Deep_Blue_%28chess_computer%29.

10. —. Chess_notation. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Chess_notation.

11. —. Chess. *Wikipedia.* [Online] http://en.wikipedia.org/wiki/Chess.

12. **Alticast.** Alticast. *Alticast.* [Online] http://www.alticast.com.

13. **Group, Myriad.** myriadgroup. *myriadgroup.* [Online] http://www.myriadgroup.com/.

14. **Google.** TV. *Google.* [Online] http://www.google.com/tv/.

15. —. Android. *Android.* [Online] http://www.android.com/.

16. **Sun.** JMS. *JMS.* [Online] http://www.oracle.com/technetwork/java/index-jsp-142945.html.

17. **Oracle.** Code Convention. *Oracle.* [Online] http://www.oracle.com/technetwork/java/codeconvtoc-136057.html.

18. **Eclipse.** Eclipse. *Eclipse.* [Online] http://eclipse.org/.

19. **Microsoft.** Windows XP. *Microsoft.* [Online] http://www.microsoft.com/windows/windows-xp/default.aspx.

20. [Online] CMU. http://cmusphinx.sourceforge.net/.

21. **CMU.** FreeTTS. [Online] http://freetts.sourceforge.net/.

22. *XMLTV.* [Online] http://www.xmltv.org/.