
2006.1

Aplicações de TV Digital sensíveis a contexto para
dispositivos móveis.

TRABALHO DE GRADUAÇÃO EM
SISTEMAS DISTRIBUÍDOS

Aluno: Paulo Martinelli Hemmlepp {pmh@cin.ufpe.br}
Orientador: Carlos Ferraz {cagf@cin.ufpe.br}

Resumo

A TV digital é uma realizada em muitos países e em breve também será no Brasil. A TV digital móvel por outro lado ainda é uma inovação, muitos padrões ainda estão em desenvolvimento ou em teste. O desenvolvimento de aplicações interativas para TV digital móvel também está em fase inicial, e este é o momento certo para que se proponham uma evolução que é o desenvolvimento de aplicações cientes de contexto. Aplicações cientes de contexto têm como característica poder se adaptar baseando-se no ambiente que estão sendo executadas e nas preferências do usuário, mas para isso é necessário fornecer uma infra-estrutura para que as mesmas sejam suportadas.

Palavras chave: TV digital móvel, interatividade, ciência de contexto, aplicações ciente de contexto.

Agradecimentos

Agradeço primeiramente a minha mãe, Sueli Martinelli, que sempre me apoiou.

Agradeço ao meu orientador Carlos Ferraz pela confiança dada em mim na realização deste trabalho e pelo apoio dado.

Agradeço também Mário Fried, gerente do grupo de TV Digital do CESAR, por ser um ótimo chefe e permitir que eu tenha acesso a equipamentos de teste além de horas cedidas para me dedicar a este trabalho.

Agradeço Paulyne Jucá e Andrino Coelho por todas as dúvidas tiradas, revisões do meu trabalho e apoio dado a qualquer hora e dia da semana. Sem vocês não teria conseguido realizar esse trabalho tão bem.

Agradeço também todos que participam e participaram do grupo de TV Digital do CESAR, por serem ótimos companheiros, Diogo Pedrosa, Luís Zelaquett, Tarcísio Quirino, Tarcísio Gurgel, Walberg, Átila, Gustavo e Erick Lopes.

Agradeço todos os meus amigos da faculdade, em especial Ana Elizabeth, Carina Soares e Leonardo Lira por todos os trabalhos em grupos que realizamos. E também Geraldo Fernandes, Bengt, Arthur, Rodrigo, Bruno Pereira pelos momentos divertidos e saídas que realizamos.

Índice

1	Introdução	7
1.1	Objetivos	8
1.2	Estrutura da Monografia	8
2	Ciência de Contexto.....	10
2.1	Introdução: Computação Pervasiva	10
2.2	Definição de Contexto	11
2.3	Tipos e Categorias de Contexto.....	12
2.4	Computação Ciente de Contexto	15
2.5	Obtenção de Contexto.....	16
2.6	Arquiteturas de Sistemas Cientes de Contexto.....	17
2.7	Exemplos de Aplicações Cientes de Contexto	18
3	TV Digital	20
3.1	Padrões de TV Digital Terrestre.....	20
3.2	Padrões de TV Digital Móvel.....	23
3.3	Interatividade em TV Digital.....	27
3.4	Permitindo Interatividade em TV Digital Móvel	28
4	<i>Context Toolkit</i>	32
4.1	Dificuldades em Desenvolver Aplicações Cientes de Contexto	32
4.2	Principais Componentes.....	34
4.3	Arquitetura de um Sistema Utilizando o Context Toolkit	35
5	Estudo de Caso	37
5.1	Cenário	37
5.2	Desenvolvimento	38
5.2.1	Identificação dos Contextos Relevantes.....	38
5.2.2	Modelando a Aplicação Baseado no <i>Context ToolKit</i>	39
5.2.3	Implementação e Utilização da Aplicação	39
6	Conclusão.....	42
6.1	Contribuições.....	42
6.2	Dificuldades Encontradas.....	43
6.3	Trabalhos Futuros	43

Índice de Figuras

Figura 1 - Características do padrão ATSC	21
Figura 2 - Características do padrão DVB.....	22
Figura 3 - Características do padrão ISDB	22
Figura 4 - Time Slicing no padrão DVB-H	25
Figura 5 - Exibição de dados meteorológicos e financeiros no MediaFLO	26
Figura 6 - Exemplos de layout no HisTV.....	29
Figura 7 - Layout do HisTV sendo usado na prática.....	29
Figura 8 - Aplicação interativa desenvolvida em LAsER	30
Figura 9 - Arquitetura de um middleware utilizando JSR 272	31
Figura 10 - Arquitetura comum ao se usar o <i>Context Toolkit</i>	36
Figura 11 - Arquitetura da Camada de Obtenção de Contexto.....	40
Figura 12 - Tela inicial para usuário que gosta de teatro e museus	41
Figura 13 - Tela inicial para usuário que gosta de jogos.....	41
Figura 14 - Tela após o usuário andar por alguns quarteirões	41

Índice de Tabelas

Tabela 1 – Dimensões Semânticas de contexto	12
Tabela 2 – Categorias de Contexto	13
Tabela 3 – Informações de contexto por entidades	38
Tabela 4 – <i>Widgets</i> para obtenção de contexto para a aplicação	39

1 Introdução

A TV digital interativa já é uma realidade em inúmeros países, e em breve também será no Brasil, com suas transmissões começando possivelmente no final de 2007. Os telespectadores deixam de ser figuras passivas em frente à televisão e passam a interagir com aplicativos que são transmitidos pela emissora de TV via broadcast.

A transmissão de TV digital para dispositivos móveis também já é realidade em vários países. Algumas tecnologias são mais antigas como DMB ou *MediaFLO*, outras estão acabando de sair da fase de especificação e teste, como é o caso do DVB-H, que foi bastante testado na Copa do Mundo de Futebol de 2006.

Entretanto, o desenvolvimento de aplicações para TV digital móvel ainda está em nos seus primeiros estágios, muitas especificações para permitir aplicações em TV digital móvel ainda estão sendo escritas, como é o caso da JSR 272 que permite desenvolver aplicativos em Java para dispositivos móveis com capacidades de receber TV móvel, que começou a ser especificada no início do ano de 2006.

Como a interatividade em TV digital móvel ainda está na sua fase inicial, esse é o momento certo de que se proponha uma mudança ou uma evolução, que no nosso trabalho é a sugestão de se fazerem aplicações que sejam cientes de contexto.

Aplicações cientes de contexto têm como característica poder se adaptar baseando-se no ambiente que estão sendo executadas e nas preferências do usuário. Mas para que elas sejam possíveis de ser implementadas, é preciso ter toda uma infra-estrutura de sensores e de softwares para que se consiga obter as informações de contexto necessárias para sua execução.

1.1 Objetivos

Este trabalho tem por objetivo mostrar o que é computação ciente de contexto e os tipos de contexto existente, para que os desenvolvedores de aplicações possam tirar o máximo proveito de contexto ao fazer o planejamento de uma aplicação. Além disso, será mostrado um *toolkit* que facilita a implementação de aplicações cientes de contexto, o *Context Toolkit*.

Na área de TV digital será mostrada uma breve visão dos padrões existentes pelo mundo, tanto de TV fixa, como para a móvel, e as formas de desenvolvimento de aplicações existentes atualmente para os dispositivos com TV digital móvel.

Por fim será mostrado um cenário de uma possível aplicação para TV digital móvel com ciência de contexto, como identificar os contextos relevantes e como modelar esse cenário usando o *Context Toolkit*. Como prova de conceito será desenvolvido um protótipo usando a linguagem de programação Java.

1.2 Estrutura da Monografia

Este trabalho está organizado da seguinte forma:

No capítulo 2 será apresentada a ciência de contexto, mostrando as definições de contexto, suas categorias e o que é computação ciente de contexto. Será abordado também como o contexto é obtido e exemplo de aplicações que foram feitas por outros pesquisadores.

O capítulo 3 trata da TV digital de modo geral, mostrando os padrões terrestres e móveis, interatividade e como é feita a interatividade em TV digital móvel.

O *Context Toolkit* que será usado no auxílio do desenvolvimento de uma aplicação ciente de contexto será explicado no capítulo 4. Expondo quais problemas ele permite solucionar, quais são seus principais componentes e como modelar a aplicação baseado na sua arquitetura.

No capítulo 5 será exposto um cenário plausível para uma aplicação ciente de contexto para dispositivo móvel, também será mostrado como identificar quais são os contextos relevantes e como modelar a aplicação usando o *Context Toolkit*.

O Capítulo 6 encerra este trabalho com as conclusões, contribuições, dificuldades encontradas e sugestões de trabalhos futuros.

2 Ciência de Contexto

Neste capítulo será apresentado a ciência de contexto, mostrando suas definições, categorias e tipos e o que é computação ciente de contexto. Será abordado também como contexto é obtido e exemplos de aplicações que foram desenvolvidas por outros pesquisadores.

2.1 Introdução: Computação Pervasiva

Computação pervasiva foi idealizada pelo trabalho de [Weiser91]. Nele, Weiser prevê um mundo repleto de computação, onde as pessoas encontram-se imersas entre diversos dispositivos computacionais, sejam estes móveis ou fixos, compartilhando-os.

Para que os usuários pudessem usufruir de uma computação móvel ininterrupta e transparente foram descritas três características que se tornaram o pilar da computação pervasiva:

Interfaces naturais: Interfaces naturais são interfaces de comunicação homem-máquina que reconhecem ou reproduzem as formas de comunicação comuns aos seres humanos. Por exemplo, um sistema que tenha reconhecimento de comandos de voz, face, emoção, ou então sistemas que reconheça a escrita de uma caneta, o que é muito mais natural do que usar mouses ou teclados.

Captura automática de experiências reais: Permitir que eventos que ocorrem em nossa volta sejam armazenados por computadores, permitindo um acesso posterior, como uma forma de memória auxiliar às pessoas.

Aplicações cientes de contexto: Por fim, para que se consiga obter uma computação pervasiva, é preciso aplicações que adaptem seu comportamento baseando-se na informação adquirida do ambiente físico e computacional.

2.2 Definição de Contexto

Todo tipo de atividade está envolto e é influenciado por um contexto. Por exemplo, uma frase dita por uma pessoa pode ter seu significado alterado baseado nos movimentos que ela realiza com sua mão, ou seja, o contexto que a comunicação está inserida modifica o seu significado.

A comunicação homem-máquina é mais limitada do que a homem-homem, pois as máquinas não são capazes de reconhecer o contexto em que as coisas ocorrem. Se fosse provido um meio de acesso ao contexto para os computadores, seria aumentada a riqueza da comunicação homem-máquina.

Entender contexto é um passo importante para que o desenvolvedor possa escolher quais contextos irão ser usados na aplicação que será desenvolvida por ele.

No dicionário *Merriam-Webster*, a palavra contexto tem como definição: “*as condições inter-relacionadas em que alguma coisa existe ou ocorre*”. Inúmeros pesquisadores escreveram trabalhos em que redefinem o significado da palavra contexto a fim de adaptá-la à realidade computacional. Dentre estas, a mais citada é a de [Dey99]:

“Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação em si.”

Essa definição é uma das mais usadas, pois ela torna mais fácil ao desenvolvedor classificar o contexto em uma aplicação, pois se um pedaço de informação pode ser usado para caracterizar uma situação do usuário com a interação com a aplicação, então essa informação é contexto. Como qualquer pedaço de informação pode ser caracterizado como uma característica de contexto é comum cada pesquisador usar sua própria definição de contexto para se adaptar as suas necessidades.

2.3 Tipos e Categorias de Contexto

Para melhor entender contexto e poder modelar aplicações que possam tirar proveito ao máximo desse contexto, é preciso antes entender quais os tipos de contextos que podem ser utilizados. Vários trabalhos tentam classificar os tipos de contexto, aqui iremos mostrar algumas classificações comuns encontradas.

Em [Dey99] é apresentada uma categorização inicial que é contexto primário e secundário. Primário será composto por localização, identidade, tempo e atividade, ou seja, situações que caracterizam uma entidade e por isso são mais importantes que os outros tipos de contexto. Informações que podem ser inferidas ou encontradas através do contexto primário são classificadas como contexto secundário, como por exemplo, o e-mail de um usuário através do nome do mesmo.

[Abo00] sugere o uso das cinco dimensões semânticas para contexto conhecidas como os 5 “W” que são:

Dimensão	Descrição
Who	Quem realiza uma determinada atividade, quem pode alterar o contexto ou que pode ser notificado caso o contexto seja alterado.
Where	Onde o contexto está. Esta é uma das dimensões mais usadas devido ao grande interesse de sistemas baseados em localização.
When	A informação temporal para determinar quanto tempo uma entidade está dentro de um contexto. Esta dimensão associada com a dimensão “Where” permite rastrear os caminhos que uma entidade tomou durante um período.
What	O que o usuário está fazendo neste momento. Geralmente necessita de sensores para determinar qual é a atividade, o que torna isso uma tarefa complexa.
Why	Determinar o porquê o usuário está realizando determinada atividade, essa é uma das tarefas mais difíceis por envolver questões de inteligência artificial.

Tabela 1 – Dimensões Semânticas de contexto

Quando comparamos as cinco dimensões semânticas com a classificação de contexto primário e secundário, podemos notar que o contexto primário engloba “who”, “where”, “when” e “what” e que o contexto secundário englobaria “why”.

No trabalho de [Chen00], são propostas quatro categorias de contexto:

Contexto	Descrição
Contexto computacional	Conectividade de rede, largura de banda, custo computacional e recursos próximos como impressoras e terminais.
Contexto do usuário	Preferências do usuário, localização, mobilidade do usuário, pessoas próximas.
Contexto físico	Temperatura, luminosidade, pressão e umidade.
Contexto de tempo	Dia, mês ou hora.

Tabela 2 – Categorias de Contexto

Outra classificação de contexto muito utilizada é:

- Persistência:
 - Permanente: contexto que não modifica com o tempo. Por exemplo, o nome ou a identidade de um usuário.
 - Temporário: informação contextual que muda como posição, temperatura, etc.

- Evolução:
 - Estático: nessa categoria ficam contextos que não se alteram com muita frequência, como o endereço do usuário.
 - Dinâmico: contexto que muda muito rápido, como a posição do usuário.

- Meio:
 - Físico: referente a contextos com informações tangíveis, como temperatura, umidade, posição ou capacidade de rede. Qualquer informação que possa ser obtida através de sensores.
 - Intangível: informações que não podem ser obtidas diretamente por uso de sensores, como o nome do usuário e suas preferências.

- Relevância para a aplicação ou serviço:
 - Necessária: conjunto obrigatório de informações contextuais a serem obtidas para que uma aplicação ou serviço possam ser executados.
 - Acessório: informações de contexto que se forem obtidas podem ser usadas para prover um melhor serviço ou aplicação.

- Situação Temporal
 - Passado: informações de contexto do passado. Tem como finalidade formar uma história de contextos de um usuário.
 - Presente: o contexto neste exato instante.
 - Futuro: informações contextuais que podem ser preditas baseadas nos contextos passados e presente. Por exemplo, onde o usuário estará amanhã baseado num histórico da rotina semanal dele. Com isto, o sistema ciente de contexto poderá se preparar baixando o mapa da área onde ele possivelmente estará.

- Interação (Entre uma aplicação ciente de contexto e a parte do sistema responsável para obter o contexto)
 - Contexto *Push*: o sistema responsável por fornecer contexto envia periodicamente o contexto para a aplicação.
 - Contexto *Pull*: a aplicação tem de explicitamente requisitar ao sistema as informações de contexto.

2.4 Computação Ciente de Contexto

Ciência de contexto, para sistemas computacionais, é uma área de estudo que engloba pesquisas desde: computação móvel, computação pervasiva, redes sensíveis a serviços e *grid computing*. Em todas essas áreas de pesquisa contexto tem sido usado para aumentar a interação entre usuário-máquina e máquina-máquina, tentando prover computação em qualquer lugar a qualquer hora.

Um sistema é ciente de contexto, se ele usa contexto para prover informações ou serviços ao usuário [Dey99]. Aplicações cientes de contexto são aplicações que são capazes de modificar seu comportamento baseado nas informações de contexto ou são aplicações que mostram ao usuário informações de contexto.

No trabalho de [Schilit94] é apresentada uma das primeiras classificações para os tipos de aplicações cientes de contexto:

Ciente de contexto ativo: uma aplicação adapta-se automaticamente ao contexto descoberto, mudando o comportamento da aplicação. Por exemplo, um celular que entra em modo silencioso ao notar que o seu dono está em reunião.

Ciente de contexto passivo: uma aplicação apenas apresenta informações de contexto ao usuário ou as armazena para que o usuário possa consultar posteriormente. Por exemplo, um celular que exibe na tela a localização do usuário em um mapa.

[Pascoe98] por sua vez propôs uma classificação baseada nas principais funcionalidades de aplicações cientes de contexto:

Sensoriamento do contexto (*contextual sensing*): é a habilidade de detectar informações sobre o contexto e apresentá-las ao usuário. Nesta categoria, estão as aplicações que buscam informações para o usuário sobre o contexto, por exemplo, o sistema que apresenta informações sobre a impressora disponível mais próxima;

Adaptação ao contexto (*contextual adaptation*): é a habilidade de executar ou modificar um serviço automaticamente, baseando-se no contexto atual. São baseados

numa simples regra de *if-then-else*. Um comando é executado quando existe uma determinada combinação de contexto.

Aumento do contexto (*contextual augmentation*): é a habilidade de associar informações ao contexto para recuperar posteriormente. São aplicações que permitem associar dados digitais ao contexto do usuário. Por exemplo, um usuário pode associar um comentário a uma obra de arte que ele está olhando para que outras pessoas possam ler depois.

2.5 Obtenção de Contexto

Para que uma aplicação seja ciente de contexto é necessário que alguém informe a ele essas informações. Podemos definir a obtenção de contexto em dois tipos, de baixo nível e de alto nível. Nessa seção será explicado como são obtidas algumas dessas informações de contexto.

Contexto de Baixo Nível

Esse tipo de contexto geralmente é obtido através de sensores, sendo de fácil obtenção.

Localização: A localização de um usuário pode ser obtida através de um sistema de GPS caso ele se encontre do lado de fora de um prédio ou por infravermelho dentro de residências. Também já foi usado um dispositivo que emite um ultra-som inaudível para seres humanos, mas que permite que sensores determinem a localização do emissor.

Tempo: Essa informação pode ser facilmente obtida através do relógio interno de um sistema.

Objetos próximos: Pode ser obtido pelo uso de etiquetas de RFID nos objetos ou mesmo através de um banco de dados com a posição dos objetos, desde que eles não sejam alterados de lugar.

Largura de banda: A largura de banda de uma rede pode variar muito e de forma imprevista. Uma API que faça a tarefa de analisar qual o estado da rede pode fornecer essa informação.

Orientação: Um sensor com um tubo de mercúrio dentro do dispositivo permite saber qual a inclinação de um dispositivo.

Contexto de Alto Nível

Contexto de alto nível é um grande desafio da ciência de contexto. São informações como que atividade o usuário está fazendo ou o porquê ele está fazendo. Uma abordagem utilizada é a de visão de máquina, onde uma câmera observa o usuário e tenta inferir o que ele está fazendo. Outra abordagem é o uso de um calendário onde o usuário informa tudo que ele deveria fazer durante o dia, mas isso torna o sistema intolerante a imprevistos. Uma terceira abordagem muito utilizada é o uso de inteligência artificial para reconhecer contextos complexos baseado nas informações de contexto de baixo nível obtidos pelos sensores.

2.6 Arquiteturas de Sistemas Cientes de Contexto

Aplicações cientes de contexto podem ser implementadas de várias formas, baseado nos requisitos, como os sensores são acessados (local ou remotamente), quantidade de usuários, recursos disponíveis pelo dispositivo.

Normalmente existem três formas arquitetura de aplicações ciente de contexto:

Arquitetura de Acesso Direto aos Sensores: normalmente usado quando o dispositivo possui sensores instalados nele. A aplicação acessa diretamente as informações obtidas pelo sensor, ganhando velocidade de acesso da informação. Em contra partida a aplicação fica fortemente acoplado aos *drivers* dos sensores, o que dificulta a extensão da aplicação e a capacidade de fazer sistemas distribuídos.

Arquitetura Baseada em *Middleware*: Uma camada de *middleware* é inserida entre a aplicação e a camada de sensores, permitindo esconder detalhes de baixo nível dos sensores e com isso aumentado a extensibilidade da aplicação.

Servidor de Contexto: É uma evolução da arquitetura baseado em *middleware*, permitindo que múltiplos clientes acessem concorrentemente as informações dos sensores. É similar a arquitetura cliente-servidor, onde o servidor terá ao seu acesso os sensores e as informações geradas e o cliente irá requisitar essas informações. Problemas como protocolo de comunicação, performance de rede e qualidade de serviço devem ser levados em conta ao se usar um servidor de contexto.

2.7 Exemplos de Aplicações Cientes de Contexto

Nesta sessão serão citados alguns exemplos de aplicações cientes de contexto que foram desenvolvidas ao longo do tempo.

Active Badge Location System [WHF92]

É considerada uma das primeiras aplicações ciente de contexto da história. Um conjunto de sensores infravermelho instalados no ambiente permitem determinar a posição do usuário e com isso redireciona uma chamada telefônica ao telefone mais perto do usuário.

CyberGuide [LKA96]

Os usuários equipados com PDAs são guiados através de caminhos e rotas dentro de um museu. Informações adicionais sobre a obra como história e vida do artista que o usuário está olhando são exibidas.

O usuário também pode deixar comentários sobre cada obra visitada para que outros usuários leiam. A aplicação também compila um diário para o usuário com todos os lugares que ele visitou e faz sugestões de outros lugares para visitar.

Comunicação ciente de contexto em hospitais [JCAF 05]

Cada usuário como médicos e enfermeiras são equipados com dispositivos móveis para escrever mensagens que serão enviadas quando um conjunto de circunstâncias ocorrerem. Por exemplo, uma enfermeira pode formular uma mensagem que será enviada ao primeiro médico a entrar num determinado quarto depois das oito horas da noite.

3 TV Digital

TV digital é uma evolução da TV analógica tradicional, permitindo melhor imagem, melhor som, mais canais e a possibilidade do telespectador interagir com a TV através de aplicações vindas junto com o áudio e vídeo. Seu surgimento tem como início o ano de 1987 nos Estados Unidos, quando o ACATS (*Advisory Committee on Advanced Television*) decidiu desenvolver um sistema totalmente digital para televisão. Em seguida foi criado o ATTC (*Advanced Television Test Center*) que entre período de 1990 a 1992, testou seis propostas para TV digital, mas que não atendiam totalmente os requisitos que eram alvos da instituição. Em 1993 os europeus também decidem criar um consórcio para desenvolver seu próprio padrão. Mais tarde, em 1997, foi a vez dos japoneses. O primeiro sistema a entrar em operação foi o europeu em 1998 na Inglaterra.

Neste capítulo, falaremos sobre os padrões de TV digital terrestre e móvel presentes no mundo.

3.1 Padrões de TV Digital Terrestre

ATSC [ATSC]

O comitê ATSC (*Advanced Television System Committee*) desenvolveu o padrão de TV digital americano. Atualmente além de ser adotado nos EUA, ele se encontra presente no Canadá, México, Taiwan e Coréia do Sul.

Uma das principais características deste sistema é a preocupação em transmissões em HDTV (*High-definition television*) com resoluções de até 1920 por 1080 pixels e som com 5.1 canais, ou seja, uma imagem de qualidade muito superior a imagem da TV analógica e som mais realista que o oferecido pelos 2 canais da TV analógica.

A modulação de transmissão é a 8-VSB, e devido as suas características é mais adequado com o uso de transmissão por cabo. Sua transmissão por ar apresenta pouca robustez contra interferências no sinal, o que atualmente impossibilita seu uso na transmissão para TV digital móvel.

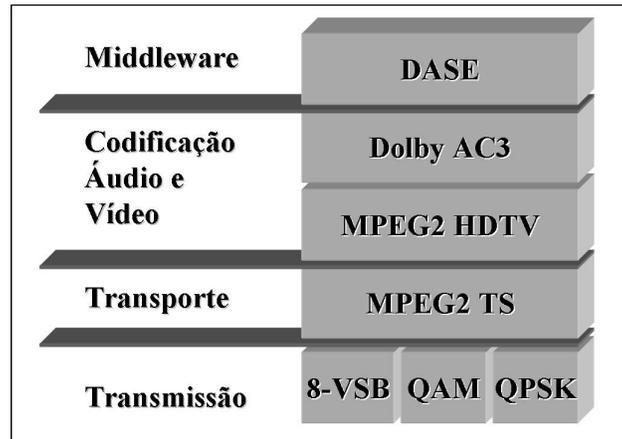


Figura 1 - Características do padrão ATSC

DVB [DVB]

O padrão DVB (*Digital Video Broadcasting*) é um conjunto de especificações desenvolvidas pelo *DVB Project*, um consórcio composto por mais de trezentos membros, como fabricantes de equipamentos, operadores de rede e órgãos de regulamentação. Por ser um padrão aberto e pelo esforço que *DVB Project* em sempre melhorar as especificações, tornaram esse padrão o mais usado pelo mundo, sendo adotado nos países da União Européia e em outros países como Austrália, Nova Zelândia, Malásia, Hong Kong, Singapura, Índia e África do Sul.

O padrão DVB é formado por um conjunto de documentos que definem os diversos padrões adotados, incluindo aqueles relacionados à transmissão, transporte, codificação e *middleware*. O padrão DVB permite diversas configurações para a transmissão, e os principais padrões de transmissão adotados pelo DVB são: DVB-T: (transmissão terrestre por radiodifusão); DVB-C (transmissão via cabo); DVB-S (transmissão via satélite).

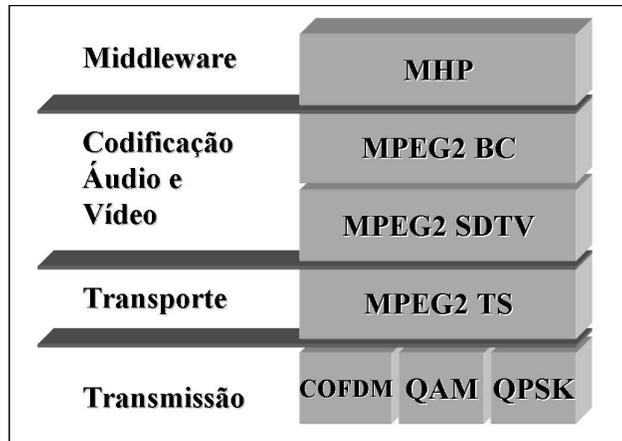


Figura 2 - Características do padrão DVB

ISDB [ISDB]

O *Integrated Services Digital Broadcasting* (ISDB) é o padrão desenvolvido pela organização japonesa DiBEG (*Digital Broadcasting Experts Group*).

Por ter sido um dos últimos padrões a ser concebido, ele pôde incorporar inúmeras melhorias no sistema de modulação. Com isso, o sistema adotado é mais robusto se comparado com a modulação de outros padrões.

Uma característica da modulação do ISDB é que ele permite transmitir um programa em HDTV e um programa para dispositivo móvel ao mesmo tempo ou então três programas em SDTV (*Standard-definition television*) e um programa para dispositivo móvel ao mesmo tempo.

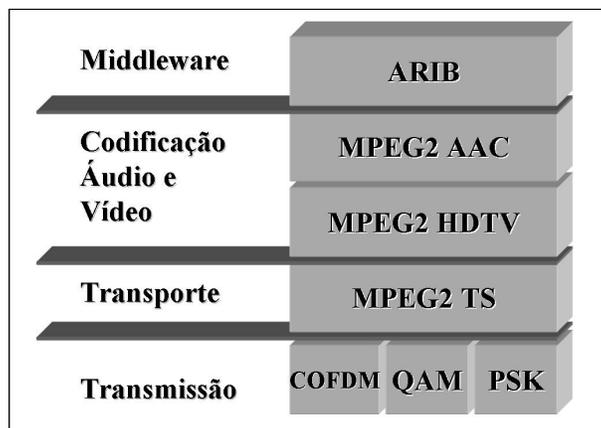


Figura 3 - Características do padrão ISDB

Sistema Brasileiro de TV Digital

O Sistema Brasileiro de TV digital foi definido pelo decreto 5820/06 assinada pelo presidente Luís Inácio Lula da Silva em 2006. Como base ele usará o sistema de modulação do ISDB, devido suas características de robustez contra interferência.

Além disso, algumas modificações¹ desenvolvidas por pesquisadores brasileiros serão acrescentadas para melhorar ainda mais a robustez do sinal e se espera a adoção do padrão de compressão de áudio e vídeo H.264.

3.2 Padrões de TV Digital Móvel

A possibilidade de assistir notícias, esportes ou qualquer outro programa de TV em dispositivos móveis como celulares ou PDA já é uma realidade. A transmissão para dispositivos móveis é feita em *broadcast* da mesma forma que para dispositivos fixos.

De acordo com pesquisas com consumidores Europeus realizada pela Nokia, mais de 70% dos consumidores acham uma boa ou excelente idéia poder assistir TV no celular [NOKIA].

A transmissão de TV digital para dispositivos móveis apresenta características diferentes da tradicional transmissão para dispositivos fixos, uma vez que celulares e PDAs possuem telas de tamanho reduzido, uma antena pequena e geralmente embutida dentro do aparelho e com uma bateria de tempo de vida reduzida, um dos maiores limitadores da TV digital móvel. Os receptores fixos por sua vez possuem antenas grandes que ficam em telhados, telas grandes e fonte constante de energia.

¹ Ainda não foi bem definido quais modificações irão entrar no padrão Brasileiro, mas certamente o *middleware* adotado será um diferente de todos os outros padrões.

ISDB

O sistema de TV digital ISDB tem como característica dividir cada canal de transmissão em 13 segmentos. Usando 12 segmentos uma transmissora pode transmitir um programa em HDTV ou pode usar 4 segmentos para um programa SDTV, o que permite transmissão de 3 programas SDTV usando 12 segmentos. Em todos os casos sobra 1 segmento que é reservado para transmissão para dispositivos móveis.

Um serviço comercial de TV digital Móvel disponível no Japão se chama 1seg [SEG], justamente por essa característica de usar 1 segmento do canal. No caso do serviço do 1seg o vídeo é codificado em H.264 e o áudio em AAC. A resolução máxima do vídeo é de 320 por 240 pixels.

DVB-H [DVBH]

O *Digital Video Broadcast Handheld* (DVB-H) é um padrão aberto e foi criado especificamente para transmissão de conteúdo de TV digital para dispositivos móveis. DVB-H é baseado no DVB-T, o qual é utilizado na Europa para transmissão terrestre. Dentre as principais diferenças com os outros padrões, podemos citar:

IP Datacasting: datagramas² IP são transmitidos encapsulados dentro da *transport stream*³, ou seja, na transmissão normal de TV digital, pacotes IP são enviados.

Multiprotocol Encapsulation: Com o *IP Datacasting* é possível adicionar cabeçalhos nos datagramas IP, o que aumenta em 25% seu tamanho, entretanto com esses cabeçalhos extras é possível acrescentar informações de FEC (*forward error correction*), que melhora a robustez e performance da recepção do dispositivo.

Time Slicing: a transmissão dos dados é feita em forma rajadas, utilizando uma taxa de transferência instantânea bastante alta se comparada ao modo convencional, e

² Pacote da camada de rede com informações completas de endereços, permitindo que seja roteada até seu destino final.

³ Fluxo de pacotes MPEG-2 usados na transmissão de TV Digital.

aguardando um tempo t para o envio da próxima rajada de dados. O *time slicing* é utilizado com o propósito de reduzir o consumo de energia das baterias dos dispositivos móveis. Essa redução se deve ao fato do receptor do aparelho ficar ligado somente no tempo que a rajada de dados ocorre. No intervalo o receptor fica desligado economizando energia.

Na figura abaixo podemos ver que somente durante o tempo das barras verticais o dispositivo gasta energia com recepção.

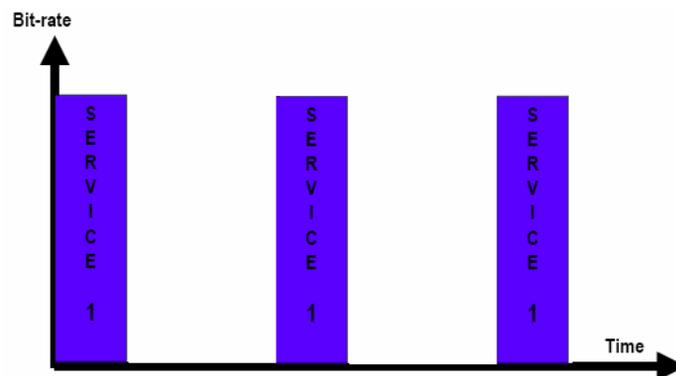


Figura 4 - Time Slicing no padrão DVB-H

MediaFLO [QUALCOMM]

MediaFlo - o FLO é de *Forward Link Only* - é uma tecnologia da Qualcomm para *broadcast* de áudio, vídeo e dados. É usado um espectro de baixa frequência na faixa de 700 MHz, permitindo o uso de menos torres de transmissão que uma rede celular normal. De acordo com informações comerciais da Qualcomm, o número de torres será de 30 a 50 vezes menor que a telefonia celular. Um exemplo dado é a cidade de São Francisco nos EUA, que precisa de centenas de torres para celular e de apenas quatro para o serviço de TV *MediaFLO*.

A especificação do *MediaFLO* promete que o serviço oferecido seja capaz de fornecer de 50 a 100 canais. O formato de vídeo fornecido é o QVGA (*Quarter Video Graphics Array*) que tem como característica ter a resolução de 320×240 pixels. A Qualcomm também afirma que a tecnologia foi desenvolvida para minimizar o

consumo de energia, o tamanho dos dispositivos móveis e que pode ser integrado nos chips dos celulares.

Além de transmitir áudio e vídeo, a tecnologia também permita a transmissão de dados. Na figura abaixo podemos ver uma aplicação que exibe os dados que estão sendo recebidos pelo dispositivo como temperatura e preço de ações da bolsa de valores.



Figura 5 - Exibição de dados meteorológicos e financeiros no MediaFLO

DMB[DMB]

O *Digital Multimedia Broadcasting* é uma extensão do padrão Eureka 147 de broadcast digital de áudio. Esse padrão recebeu grandes investimentos do governo da Coreia do Sul e permite tanto transmissões por satélite S-DMB ou por torres fixas T-DMB. Na Coreia do Sul o T-DMB pode ser visto de graça pela população, mas a tecnologia apresenta problemas de cobertura e recepção dentro de prédios. O S-DMB é oferecido como serviço pago e oferece 11 canais de vídeo. O consumo de bateria é um problema crítico do DMB.

3.3 Interatividade em TV Digital

A TV digital tem como principal diferença da TV analógica permitir que aplicações similares a que são executadas em computadores pessoais sejam transmitidas em conjunto com o áudio e vídeo. As emissoras e agências de publicidade podem fazer programas que sejam mais atraentes ao telespectador, permitindo que ele possa interagir com aplicações associadas a um determinado comercial ou programa de TV, fazendo com que o usuário de TV digital não seja apenas um elemento passivo, mas sim um usuário ativo com a programação, interagindo através da TV na hora que lhe convier. Existem várias classificações para o tipo de interação que o telespectador pode ter com as aplicações de TV digital, umas das mais usadas [MORRIS] se baseia na presença e utilização de um canal de retorno no dispositivo de recepção.

Canal de retorno se refere a um meio que o dispositivo possa se comunicar com um servidor, podendo ser um modem ou uma placa de rede *ethernet* ou sem fio. Os níveis de interatividade suportados pela TV digital são:

- **Broadcast-only ou interatividade local:** nessa situação o dispositivo não apresenta ou não está conectado a um canal de retorno. Com isso a aplicação só é executada localmente não havendo uso do canal de retorno. Mesmo assim é interessante, pois jogos ou propagandas interativas podem ser executadas sem a necessidade do canal de retorno.
- **Two-way Interactivity:** o dispositivo de recepção possui um canal de retorno e com isso ele pode enviar informações para um servidor e vice-versa. Aplicações como *home banking*, *e-mail* e navegação na *internet* se tornam possíveis.
- **Internet TV:** nesse caso o dispositivo possui um canal de retorno que está sempre ativo e com largura de banda considerável. Aplicações que podem acessar a *Internet* de forma livre estão presentes nesse nível.

3.4 Permitindo Interatividade em TV Digital Móvel

A fim de permitir interatividade do usuário com seu dispositivo móvel como por exemplo, um celular, é necessário que o dispositivo tenha recursos de software para a execução de aplicações. Existem três abordagens de software para celular que permitem interatividade: *browser* declarativo, transição multimídia, e *middleware* baseado em Java.

Browser Declarativo

Um *browser*, similar aos que se usa para navegar na internet em computadores pessoais, tem de estar presente no terminal móvel. A aplicação é desenvolvida de forma declarativa, ou seja, usando “tags” similar a de html e xml. Um conjunto de elementos padronizados como campos de texto, botões e campos para figuras podem ser usados, customizados e organizados de forma simples em *layouts* pré-definidos. É possível ainda sincronizar o áudio e vídeo com eventos que ocorrem na aplicação.

Essa abordagem é mais fácil e de menor custo, pois se faz necessário apenas um *browser*. O numero limitado de elementos suportados torna as aplicações mais simples. Essa característica também pode significar aplicações menos ricas e interessantes.

Um exemplo de *browser* declarativo comercial é o HisTV [HISTV]. Ele permite dividir a tela em áreas retangulares, denominadas de *ifields*, que são divididos baseados em *layouts* predefinidos. Cada *ifield* pode conter figuras ou textos e pode ser sincronizado com o vídeo, tornando possível construção de aplicações como quiser, onde uma pergunta aparece na tela ao mesmo tempo em que alguém pergunta no vídeo.

<p>ID 1 : vsc 240x180+0+0</p>	ID 13 : stb 20x20+240+	ID 21 : stb 20x20+260+	ID 21 : stb 20x20+280+	ID 22 : stb 20x20+300+
	<p>ID 3 80x40+240+20</p>			
	<p>ID 4 80x40+240+60</p>			
	<p>ID 5 80x40+240+100</p>			
	<p>ID 6 80x40+240+140</p>			
<p>ID 2 240x40+0+180</p>				
ID 7 80x40+240+180				
ID 8 60x20+0+220	ID 9 60x20+60+220	ID 10 60x20+120+220	ID 11 60x20+180+220	ID 12 80x20+240+220

Figura 6 - Exemplos de layout no HisTV



Figura 7 - Layout do HisTV sendo usado na prática

Transmissão multimídia

Ainda em desenvolvimento, a idéia consiste em usar certas características do fluxo de dados existente no padrão MPEG-4 para desenvolver aplicações interativas [LASER].

Uma tecnologia que está sendo adotada é a LAsER (*Lightweight Application Scene Representation*). Sua especificação é a ISO/IEC 14496-20, e faz parte do conjunto de especificações de MPEG-4. Ela é uma especificação de serviços *Rich Media* voltada para dispositivos móveis, sistemas embarcados e dispositivos domésticos. Um serviço de *Rich Media* é uma coleção interativa e dinâmica de dados multimídia, como áudio, vídeo, gráficos e texto. Pode ser, por exemplo, um vídeo enriquecido com sobreposição de gráficos vetoriais. Essa tecnologia foi inspirada na tecnologia proprietária da *Macromedia Flash*, para computadores pessoais.



Figura 8 - Aplicação interativa desenvolvida em LAsER

Middleware Baseado em Java

O uso de um *middleware* baseado em Java permite desenvolver aplicações com lógica complexa, com controle total sobre a interação. Este tipo de aplicação pode ser transmitida junto com o fluxo de áudio e vídeo ou baixada pelo canal de retorno. Da mesma forma que os padrões de TV digital terrestre possuem *middlewares* Java como por exemplo, o *MHP*, está sendo especificado o JSR 272 [JSR272], que permite que o *middleware* MIDP acesse informações do fluxo de dados da transmissão de TV digital móvel e gerencie as aplicações vindas no mesmo.

JSR 272

JSR 272, denominada *Mobile Broadcast Service API for Handheld Terminals*, busca definir uma API básica para interação com serviços de *broadcast* para dispositivos móveis, sendo uma extensão do ambiente MIDP, bastante usado atualmente para desenvolver aplicações e jogos para celulares. Essa especificação associada com a atual especificação do ambiente MIDP será a base para o *middleware* Java de TV digital móvel.

Sua especificação define:

1. Busca e execução de serviços.
2. Acesso e execução de serviços.
3. Assinatura e compra de serviços.
4. Recepção e gerenciamento de aplicações transmitidas por broadcast.
5. Parametrização de aplicações.

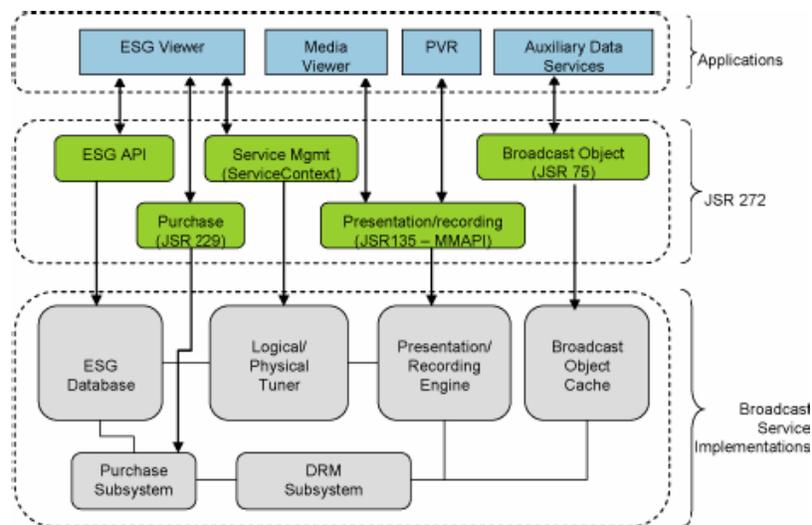


Figura 9 - Arquitetura de um middleware utilizando JSR 272

4 Context Toolkit

O *Context Toolkit* [CTK01] é um software, desenvolvido em Java, que ajuda no desenvolvimento de aplicações cientes de contexto, fornecendo uma estrutura básica para o desenvolvimento, além de fornecer um grande número de componentes reusáveis.

4.1 Dificuldades em Desenvolver Aplicações Cientes de Contexto

O desenvolvimento de aplicações cientes de contexto esbarra numa série de dificuldades e problemas técnicos. O *Context Toolkit* busca solucionar os seguintes problemas:

Separação de Responsabilidades

Um dos motivos que explicam porque contexto não é mais usado em aplicações é o fato de não haver um jeito comum de obter e manipular contexto. Geralmente nos trabalhos científicos o contexto é manipulado de forma improvisada.

Os desenvolvedores escolhem técnicas que são mais fáceis de implementar, o que impede o reuso. Uma forma comum é a aplicação ter acesso direto aos sensores, ou então usando servidores para esconder detalhes sobre os sensores. Quando a aplicação tem acesso direto aos sensores, é preciso escrever código que trate de detalhes dos sensores. Isso torna a aplicação difícil de ser reusada. Um *toolkit* permite uma abstração que permite a separação da semântica da aplicação com o tratamento de baixo nível de recebimento de contexto. Se todos os objetos que tratam a obtenção de contexto tiverem uma interface em comum, todas as aplicações tratarão esses objetos de forma similar.

Interpretação de Contexto

Para adicionar mais abstração para a aplicação, os dados obtidos pelos sensores devem passar por uma interpretação, a fim de converter dados de baixo nível em de alto nível ou então combinar várias informações de sensores em uma única informação de alto nível. Esse processo de interpretação deve ser de fácil reuso por múltiplas aplicações, por isso ela devendo ser implementada pela arquitetura.

Comunicação Distribuída e Transparente

As aplicações que usam contexto geralmente não estão conectadas diretamente aos sensores, além disso, os sensores são distribuídos e múltiplas aplicações precisam acessar seus dados. É preciso haver mecanismos, para fornecer comunicação transparente entre os sensores e a aplicação, sem que seja necessário se preocupar com protocolos de comunicação e codificação de dados.

Disponibilidade o Tempo Todo de Aquisição de Contexto

Os componentes de aquisição de contexto têm de executar de forma independente das aplicações que usam contexto. Como não se sabe em que momento a aplicação solicitará a informação, esses componentes tem de estar executando o tempo todo.

Armazenamento e História de Contextos

Além da necessidade da disponibilidade de aquisição de contexto o tempo todo, é desejável o armazenamento de informações de história de contexto. Essa história pode ser usada para determinar padrões e prever comportamentos. Por isso deve haver mecanismos para capturar e armazenar o contexto mesmo quando nenhuma aplicação está usando esta informação.

Descoberta de Recursos

Par que uma aplicação possa se comunicar com um sensor é preciso antes saber que tipo de informações ele provê, onde está localizado e como se comunicar com ele. Para esconder esses detalhes da aplicação a arquitetura tem de fornecer métodos de descoberta de recursos. Com o mecanismo de descoberta de recursos, quando uma aplicação é iniciada, ela informa quais contextos ela necessita e o mecanismo automaticamente acha quais são os meios de obter esse contexto.

4.2 Principais Componentes

Dentre os inúmeros componentes fornecidos pelo *Context Toolkit*, os mais importantes são os *widgets*, *aggregators*, *interpreters* e *discoverers*. Suas funcionalidades são:

Widgets

São os objetos responsáveis por estar mais próximo aos sensores que irão obter as informações de contexto. Eles oferecem as seguintes funcionalidades:

1. Permitem esconder a complexidade do uso de sensores para a aplicação.
2. Eles fornecem abstração suficiente para esconder detalhes desnecessários à aplicação.
3. Eles são componentes reusáveis e customizáveis de obtenção de contexto. Um *widget* de obtenção de localização pode ser usada por várias aplicações diferentes.
4. Fornecem funções de *callbacks* que notificam a aplicação quando uma mudança significativa de contexto ocorre.

Interpreters

Interpreters são responsáveis por aumentar o nível da informação obtida pelos sensores. Por exemplo, um sensor pode fornecer informações de baixo nível como

coordenadas geográficas e o *interpreters* fornece o nome de uma rua baseado nessa posição.

Aggregators

Os *Aggregators* servem para agrupar as informações vindas de vários *widgets* ou *interpreters*. Como os sensores estão geralmente espalhados, em vez de a aplicação ter de se comunicar com esses sensores um a um, ela pode só se comunicar com um *aggregator*. Isso é muito útil para fornecer de uma vez o contexto de uma entidade, como uma pessoa ou lugar.

Discoverers

Esse componente é responsável pelo registro dos recursos disponíveis, como *widgets*, *interpreters* e *aggregators*. Quando um dos outros três componentes são criados, eles informam sua presença e capacidades ao *Discoverer* que os registram para facilitar sua futura localização e utilização.

4.3 Arquitetura de um Sistema Utilizando o Context Toolkit

Para utilizar o *Context Toolkit*, o desenvolvedor deve modelar sua aplicação baseada nos componentes principais fornecidos pela ferramenta. Para esse processo deve-se usar uma série de passos simples e bem definidos:

1. Identificar quais são as entidades relevantes para a aplicação. Entidades são os elementos no qual se deseja obter informações de contexto, como pessoas, lugares ou ambientes.
2. Identificar em cada entidade quais informações de contexto serão necessárias.
3. Para cada informação de contexto de cada entidade, deve-se criar um *widget* para aquisição da informação.
4. Para cada entidade, devesse criar um *aggregator* para agrupar as informações coletadas pelos *widgets*.

5. A aplicação deve obter as informações de contexto se comunicando com os *aggregators* de todas as entidades envolvidas.
6. Se for preciso realizar convenções de informação de baixo nível em alto nível ou inferência, deve-se criar *interpreters* para cada operação dessas.

Seguindo os passos apresentados, a arquitetura de sistemas utilizando o *Context Toolkit* segue o padrão onde de um conjunto de *widget* captura as informações, um *aggregator* agrupa esse conjunto de *widgets* e a utilização de *interpreter* gera informações de alto nível para a aplicação.

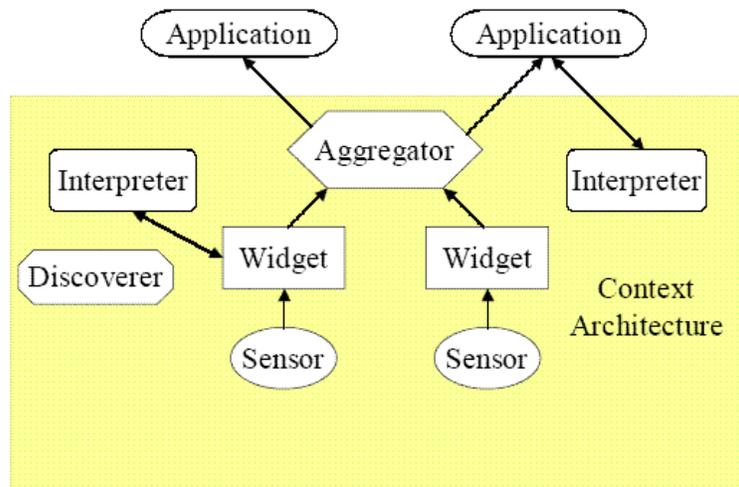


Figura 10 - Arquitetura comum ao se usar o *Context Toolkit*

Quando uma aplicação é iniciada, ela se comunica com um *discoverer* para localizar os componentes que serão necessários para seu funcionamento. Com isso a aplicação recebe informações de como se comunicar com os *widgets* e *aggregator*. A aplicação então se inscreve nos *aggregators* para o recebimento de mensagens. Nesse processo a aplicação diz quais dados deseja receber e em que condições. Os *aggregator* são responsáveis por se inscrever no recebimento de mensagens dos *widgets* que fornecem informações relevantes a entidade que o *aggregator* representa.

5 Estudo de Caso

Como prova de conceito, neste capítulo iremos propor um cenário de uma possível aplicação ciente de contexto para um dispositivo móvel com capacidade de receber TV digital. Será mostrado como distinguir os contextos relevantes para a aplicação, como modelá-la utilizando o *Context Toolkit* e a utilização dessa aplicação.

5.1 Cenário

Um turista, tendo a sua disposição um novo modelo de celular, que lhe permite assistir programas de TV digital em qualquer lugar que ele deseje e executar aplicações interativas, está visitando pela primeira vez a ilha do Porto Digital em Recife Pernambuco.

Ao chegar à Praça do Marco Zero, um ponto turístico da ilha, ele resolve usar seu celular para assistir TV, pois ele descobre que existe uma transmissão de uma rede de TV comunitária da região que tem uma programação voltada justamente para os turistas, mostrando pontos históricos abertos a visitação, como também empresas de tecnologia instaladas na região.

Após assistir por algum tempo a programação da transmissora local ele é notificado que existe uma aplicação associada à transmissão. O turista aceita que essa aplicação seja iniciada. A aplicação é carregada mostrando na tela do seu celular um mapa da área do Porto Digital.

Por essa ser uma aplicação ciente de contexto, o mapa já abre mostrando o ponto onde o usuário se encontra, uma vez que ela obteve o contexto de localização do usuário. Também são exibidos lugares de interesse do usuário baseado em suas preferências. Por exemplo, se o turista for religioso será destacado a primeira sinagoga judaica do Brasil, caso prefira jogos de computadores será destacada empresas de tecnologia especialistas em desenvolvimento de jogos, como Meantime e Jynx.

5.2 Desenvolvimento

5.2.1 Identificação dos Contextos Relevantes

Baseado no cenário apresentado anteriormente e nas classificações de contexto citadas no capítulo 2 deste trabalho, iremos determinar quais contexto serão utilizados para desenvolver a aplicação.

Primeiro, para facilitar a identificação dos contextos, vamos determinar quais entidades são destacadas no cenário. Entidades são os elementos que foram citados no cenário, como pessoas, lugares e objetos. Depois de definidas as entidades iremos definir quais contextos são relevantes para cada entidade.

Imediatamente podemos ver que o usuário é uma entidade, pois ele tem informações de contexto que são importantes. Podemos também destacar como entidade os pontos turísticos as empresas instaladas no Porto Digital.

Agora iremos determinar quais informações são relevantes de se conhecer de cada entidade e com isso iremos depois determinar os contextos.

Entidades	Informações Relevantes
Usuário	Localização Preferências
Ponto Turístico	Localização Tipo de ponto turístico
Empresas do Porto Digital	Localização História Ramo do Mercado

Tabela 3 – Informações de contexto por entidades

5.2.2 Modelando a Aplicação Baseado no *Context Toolkit*

Baseado nas regras de identificação de classes no *Context Toolkit* apresentados na seção 4.3, podemos definir que teremos um *aggregator* pra entidade: Usuário, Ponto Turístico e Empresas do Porto Digital.

Um conjunto de *widgets* foi criado para cada entidade, representando cada informação de contexto que poderá ser obtida. Na tabela abaixo estão listados os *widgets* criados:

<i>Widget</i>
Localização (Usuário)
Preferências (Usuário)
Localização (Ponto Turístico)
Tipo de ponto turístico (Ponto Turístico)
Localização (Ponto Turístico)
História (Empresas do Porto Digital)
Ramo do Mercado (Empresas do Porto Digital)

Tabela 4 – *Widgets* para obtenção de contexto para a aplicação

Para transformar informações de baixo nível em alto nível, foi criado um *interpreter* que converte uma posição geográfica fornecida pelos *widgets* de localização.

5.2.3 Implementação e Utilização da Aplicação

Baseado nas classes que foram levantadas na seção anterior, a arquitetura do sistema que irá fornecer as informações de contexto para a aplicação será:

Como estudo de caso, foi desenvolvida uma aplicação em Java, que obtém a posição do usuário na tela e mostra informações de interesse baseado em preferências pessoais armazenadas na aplicação.

A parte mais complexa do sistema é a camada de obtenção de contexto, que irá prover para a aplicação todas as informações relevantes sobre o contexto das entidades envolvidas assim como as mudanças que ocorrem.

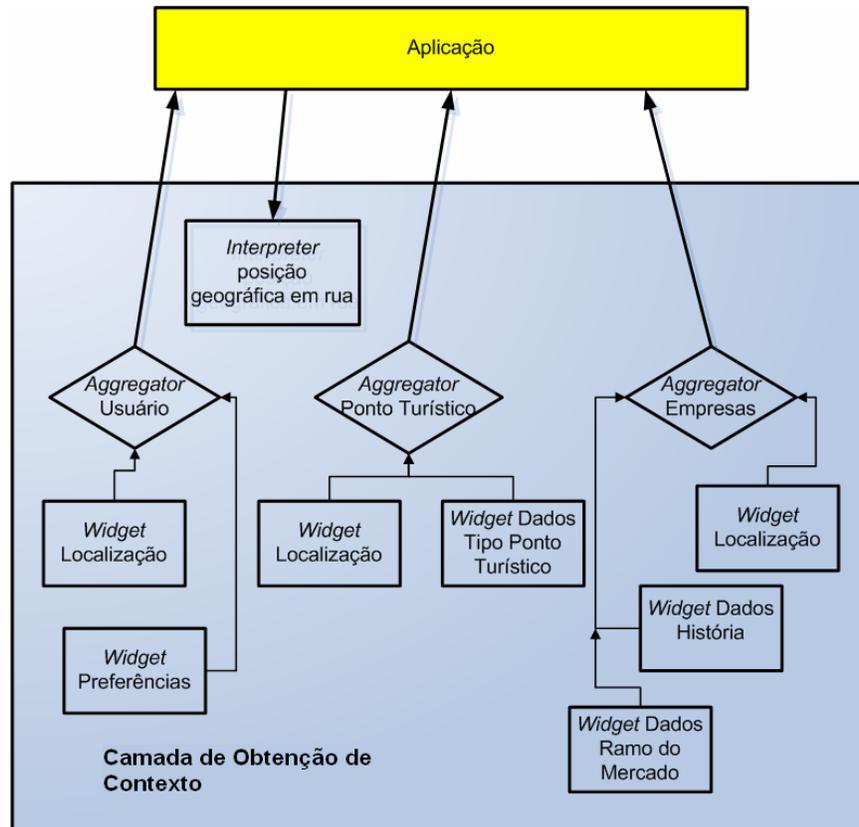


Figura 11 - Arquitetura da Camada de Obtenção de Contexto

A aplicação para o dispositivo móvel, ao ser iniciada, se comunica com um *discoverer* para saber como se comunicar com os *aggregators* e o *interpreter*. Feito isso, a aplicação obtém a posição atual do usuário e uma lista de preferências do mesmo.

Se a lista de preferências do usuário indicar que ele gosta de arte e cultura, será exibida na tela automaticamente, pontos turísticos do seu gosto.



Figura 12 - Tela inicial para usuário que gosta de teatro e museus

Por outro lado se a preferência do usuário for por jogos ou tecnologia, será exibida como tela inicial a posição dele e de empresas de desenvolvimento de jogos, caso ele queira fazer uma visita.



Figura 13 - Tela inicial para usuário que gosta de jogos

Quando o usuário se movimenta uma determinada distância, o *aggregators* responsável pela entidade usuário notifica a aplicação da mudança de localização do usuário. Fazendo a posição mostrada na tela ser atualizada.



Figura 14 - Tela após o usuário andar por alguns quarteirões

6 Conclusão

6.1 Contribuições

Como contribuições deste trabalho, podemos citar que inicialmente mostramos conceitos de computação pervasiva, depois entramos em detalhes de contexto, como tipos e classificações. Foi mostrado o que é computação ciente de contexto, como essas informações relevantes de contexto são obtidas e quais arquiteturas são usadas normalmente para prover as mesmas. Também foi mostrada exemplos de aplicações reais.

Na área de TV digital apresentamos uma rápida descrição dos padrões encontrados pelo mundo de TV digital terrestre e móvel. Foi mostrado também o que é interatividade e como ela é suportada atualmente na TV digital móvel.

Foi descrito também um *toolkit* bastante encontrado na literatura sobre ciência de contexto, o *Context Toolkit*, mostramos os problemas que ele tenta solucionar, seus componentes e como utilizá-lo.

Para finalizar foi idealizado um cenário de uma aplicação ciente de contexto para dispositivo móvel, mostramos como elicitar quais contextos são relevantes, baseado nos tipos e classificações apresentados anteriormente. Um protótipo da aplicação foi modelado baseado no *Context Toolkit* e nos contextos elicitados.

Esperamos que este trabalho tenha servido para mostrar que ciência de contexto pode ser um grande diferencial para as aplicações do futuro, tornando-as mais atrativas e intuitivas aos usuários.

Também esperamos que este trabalho tenha mostrado que aproveitando que a interatividade para TV móvel ainda está no seu começo, que sejam pensados meios de

desde o início fornecer a possibilidade de obter informações de contexto nas aplicações de TV digital móvel.

6.2 Dificuldades Encontradas

Dentre uma das principais dificuldades encontradas, podemos citar que as tecnologias para interatividade para TV digital móvel ainda estão ou acabaram de sair da fase de especificação. Por isso existe uma escassez de trabalhos científicos relevantes sobre esse assunto.

Um outro ponto foi a escolha de que *framework* ou ferramenta usar para desenvolver o protótipo da aplicação realizada neste trabalho. Existe um grande número de trabalhos descrevendo *frameworks* que possibilitam desenvolvimento de aplicações cientes de contexto, mas muitos não possuem código fonte aberto, além de serem fortemente ligados ao hardware que foi usado para fazer suas provas de conceito.

6.3 Trabalhos Futuros

Este trabalho foi apenas uma breve introdução sobre o assunto abordado. Como principal sugestão de um trabalho futuro, sugiro a implementação de um *framework* próprio voltado para o desenvolvimento de aplicações cientes de contexto para dispositivos móveis, baseado na JSR 272.

Um trabalho mais detalhando sobre o desenvolvimento de aplicações para TV digital móvel também podem ser levado em conta.

Referências

[Abo00] Abowd, G.D.; Mynatt, E.D. *Charting Past, Present and Future Research in Ubiquitous Computing*. ACM Transactions on Computer-Human Interaction, Vol 7, No. 1, Março de 2000.

[ATSC] *Advanced Television Systems Committee*. Disponível em: <http://www.atsc.org>

[Chen00] Chen, G.; Kotz, D. *A Survey of Context-Aware Mobile Computing*. Dartmouth Computer Science Technical Report TR2000-381, 2000.

[CTK01] Anind K. Dey, Daniel Salber and Gregory D. Abowd. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*. Special issue on context-aware computing in the Human-Computer Interaction (HCI) Journal, Volume 16 (2-4), 2001.

[Dey99] Dey, Anind K.; Abowd, Gregory D.. *Towards a Better Understanding of Context and Context-Awareness*. 1st International Symposium on Handheld and Ubiquitous Computing, 1999.

[DMB] *Digital Multimedia Broadcasting*. Disponível em: http://en.wikipedia.org/wiki/Digital_Multimedia_Broadcasting

[DVB] *Digital Video Broadcasting Project*. Disponível em: <http://www.dvb.org>

[DVBH] *DVB-H: Global Mobile TV*. Disponível em: <http://www.dvb-h.org/>

[HISTV] *HisTV*. Disponível em: <http://www.histv.org/>

[ISDB] *Integrated Systems Digital Broadcasting*. Disponível em: <http://www.dibeg.org/>

[JCAF 05] Jakob E. Bardram. *The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications*. Proceedings of the 3rd International Conference on Pervasive Computing, 2005.

[JSR272] *The Java Community Process number272*. Disponível em: <http://jcp.org/en/jsr/detail?id=272>

[LASER] *LASeR*. Disponível em: <http://www.mpeg-laser.org/>

[LKA96] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. *Rapid prototyping of mobile context-aware applications: the Cyberguide case study*. Proceedings of the Second Annual International Conference on Mobile Computing and Networking, 1996.

[MORRIS] Morris, Steven, Smith-Chaigneau, Anthony. *Interactive TV Standards: A Guide to MHP, OCAP and JavaTV*. Elsevier, 2005

[NOKIA] *Nokia Mobile TV Forum*. Disponível em: <http://www.mobiletv.nokia.com/>

[Pascoe98] Pascoe, J. *Adding Generic Contextual Capabilities to Wearable Computers*. 2nd International Symposium on Wearable Computers, 1998.

[QUALCOMM] *Qualcomm – MediaFLO*. Disponível em: <http://www.qualcomm.com/mediaflo/index.shtml>

[Schilit 94] Schilit, B.; Adams, N.; Want, R. *Context-Aware Computing Applications*. 1st International Workshop on Mobile Computing Systems and Applications, 1994.

[SEG] *1 seg*. Disponível em: <http://en.wikipedia.org/wiki/1seg>

[Weiser91] Weiser, Mark. *The Computer for the Twenty-First Century*. Scientific American, 1991.

[WHF92] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. *The Active Badge location system*. ACM Transactions on Information Systems, 1992.

