



UNIVERSIDADE CATÓLICA DE PELOTAS
CENTRO POLITÉCNICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Sensibilidade ao Contexto na TV Digital

por

Luthiano Rodrigues Venecian

Disciplina de Tópicos Especiais
em TV Digital Interativa

Ministrado por Prof. Paulo Roberto Gomes Luzzardi

Pelotas, novembro de 2008

Sumário

1. Introdução	7
2. Computação Ubíqua	8
2.1. Computação Sensível ao Contexto	9
3. Explorando Sensibilidade ao Contexto na TV Digital	10
3.1. Modelo de Contexto para TV Digital Interativa.....	12
4. Arquiteturas de suporte de aplicações sensíveis ao contexto na TV Digital	14
4.1. ContexTV	14
4.1.1. Cliente ContexTV	15
4.1.2. Servidor <i>ContexTV</i>	18
4.2. Context Toolkit.....	19
4.2.1. Principais Componentes	20
4.2.2. Arquitetura de um Sistema Utilizando o <i>Context Toolkit</i>	21
5. Trabalhos relacionados	22
6. Considerações Finais	22
7. Futuros Trabalhos	23
Referências Bibliográficas.....	24

RESUMO

Com o advento dos ambientes de Computação Ubíqua, tornou-se crescentemente importante para as aplicações, tirar pleno proveito das informações contextuais. O artigo discute algumas aplicações da TV Digital explorando o potencial de enriquecê-las com conceitos de sensibilidade a contexto. Sensibilidade a contexto, tem sido aceito como um paradigma apropriado para enriquecer aplicações móveis e ubíquas. Desse modo, são apresentadas duas arquiteturas: *ContexTV* e *Context TollKit*. O *ContexTV* auxilia o desenvolvedor de aplicações no cenário de convergência ao prover uma infra-estrutura para notificações de eventos e processamento de contexto e ao possibilitar a utilização de uma modelagem genérica de contexto conforme a necessidade da aplicação. Enquanto que o *Context TollKit*, fornece uma estrutura básica para o desenvolvimento, além de fornecer um grande número de componentes reusáveis.

ABSTRACT

With the advent of the Ubiquitous Computing environments, it has become increasingly important for applications, take full advantage of contextual information. The article discusses some applications of Digital TV exploring the potential of improve them with concepts of sensitivity to context. Sensitivity to context, has been accepted as an appropriate paradigm to enrich mobile applications and ubiquitous. Thus, they are only two architectures: ContexTV and Context ToolKit. The ContexTV helps the developer of applications in the scenario of convergence by providing an infrastructure for event notifications and processing of context and allowing use of a generic modeling of context as the need for the application. While the Context ToolKit, provides a basic structure for development, and provides a large number of reusable components.

LISTA DE FIGURAS

Figura 1 – Representação do Contexto.....	13
Figura 2 - Cenário de convergência e ContexTV.....	15
Figura 3 - Arquitetura do cliente J2ME MIDP 2.0.....	16
Figura 4 - Arquitetura do Servidor.....	18
Figura 5 - Arquitetura comum ao se usar o <i>Context Toolkit</i>	21

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BP	<i>Baseline Profile</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
STB	<i>Set-Top Box</i>
URD	<i>Unidade de Recepção Digital</i>
XML	<i>eXtensible Markup Language</i>

1. Introdução

A princípio, quando alguém se depara com o assunto TV Digital, logo vem à mente é a alta qualidade de áudio e definição de imagem. Entretanto, a grande mudança é baseada em três conceitos: interatividade, portabilidade e conectividade. Conceitos que suportados por *software* dão “vida” à tela da TV Digital.

Sem o *software*, a TV Digital, utilizando formatos de compressão e padrões de transmissão, simplesmente converte um sinal analógico em digital. Com *software*, adiciona à imagem o poder da interatividade, transmitindo informações que dão “vida” à TV. “Vida” que ganha “voz” quando um canal de retorno, viabilizado por uma rede com ou sem fio, uma linha discada, um celular ou qualquer outro meio que permita a conexão do telespectador para o envio de uma operação por ele solicitada, estiver conectado à TV Digital.

A interatividade e a conectividade permitem à TV Digital submeter ao telespectador um conteúdo e viabilizar uma reação. Isto significa que é facultado ao telespectador a participação e integração a um determinado conteúdo transmitido. Outro aspecto é a portabilidade, que viabiliza a utilização da TV Digital em qualquer momento em qualquer lugar.

A TV Digital proporciona mudanças para o negócio dos radiodifusores, pois possibilita a evolução da produção e distribuição de conteúdo, da comercialização de anúncios, patrocínios e *merchandising* estáticos ou semiestáticos. Também passa a uma programação dinâmica pela inserção de imagens alternativas, *links* patrocinados, informações adicionais, pesquisas, vendas *on-line*, jogos, treinamento. Enfim, é a TV Digital se aproximando dos conceitos e movimentos da Internet. Toda essa mudança insere a TV na era da convergência digital.

A convergência digital é viabilizada por uma rede global de tecnologias de informação, com interação de todos os tipos de serviços eletrônicos: áudio, vídeo, texto e dados, e para onde convergem diferentes combinações e integrações a multimídia.

O benefício que a melhoria da imagem e som traz já é grande por si só. A imagem e o som recebidos passam a ser de alta qualidade. É uma evolução que pode parecer simples, mas traz consigo uma enorme complexidade na sua implantação, pois requer a substituição dos equipamentos de recepção, afetando diretamente milhões de usuários. Para que a imagem e som sejam realmente melhorados, os televisores terão que ser substituídos ou, na melhor das hipóteses, adaptados a outro equipamento que fará a recepção e decodificação do sinal digital, os chamados *set-top-boxes*.

A inteligência da TV Digital é um *software*, chamado de *middleware*, que é processado na própria TV Digital, ou de um *set-top-box* com a função de suportar as aplicações que são transmitidas junto com a imagem recebida. Desse modo, o *software* está presente no coração da TV Digital e nas diversas aplicações que a diferenciam da TV analógica.

Este artigo discute algumas aplicações da TV Digital explorando o potencial de enriquecê-las com conceitos de sensibilidade a contexto. Sensibilidade a contexto, tem sido aceito como um paradigma apropriado para enriquecer aplicações móveis e pervasivas (Weiser, 1991). Um sistema pode reduzir sua necessidade de interações explícitas do usuário, ao interpretar suas ações anteriores e sentir o ambiente que o cerca, por exemplo (Hong, 2005). O uso de contexto pode tornar a experiência do usuário mais interessante, ajudá-lo a realizar tarefas rotineiras e permitir interações sociais ao compartilhar contexto entre usuários (Tamminen, 2004).

2. Computação Ubíqua

A expressão Computação Ubíqua foi idealizada pelo pesquisador Mark Weiser (Weiser, 1991), quando vislumbrou a expansão do paradigma de interação tradicional entre usuárias e máquina – teclado, *mouse* e monitor – para uma abordagem que vai além do modelo humano de interação natural.

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

- **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 reconheçam 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 das pessoas.
- **Aplicações sensíveis de contexto:** Por fim, para que se consiga obter uma Computação Ubíqua, é preciso aplicações que adaptem seu comportamento baseando-se na informação adquirida do ambiente físico e computacional.

2.1. Computação Sensível ao Contexto

Sensibilidade ao contexto, para sistemas computacionais, é uma área de estudo que engloba pesquisas desde: Computação Móvel, Computação Ubíqua, 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 sensível ao contexto, se ele usa contexto para prover informações ou serviços ao usuário (Dey, 1999). 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.

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.

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 (Dey, 1999): “*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.

3. Explorando Sensibilidade ao Contexto na TV Digital

O principal propósito da computação sensível ao contexto é permitir adaptações dinâmicas de aplicações e serviços para diferentes propósitos, como a utilização adequada de recursos ou criação de interfaces com usuário mais ricas e eficazes (da Rocha e Endler, 2005).

A seguir será apresentado alguns cenários do uso da TV Digital, enriquecidos com funcionalidades sensíveis a contexto:

- **Datacasting:** Serviços de *datacasting* cuja apresentação de informações leva em conta a situação do usuário. Um exemplo desta aplicação pode ser vista em (Matsumura, 2003), onde um mapa e avisos sobre uma situação de emergência

são apresentados de formas diferentes para usuários em contextos distintos. Guias e serviços de turismo são outros tipos de aplicações estudados na literatura de sensibilidade a contexto (Cheverst et Al, 2000) e poderiam ser disponibilizados por *datacasting*.

- **Notificações:** O contexto do mundo que cerca o usuário é notificado na tela quando pertinente e enriquecedor para sua experiência. Por exemplo, exibição de notificações sobre programas relacionados ao seu perfil, ou de amigos assistindo o mesmo evento de TV, algo que poderia gerar interessantes interações sociais.
- **Gravação de conteúdo:** São previstos serviços de gravação de conteúdo para consumo posterior, inclusive feita sem que o usuário tenha conhecimento prévio (JSR272, 2005). A ativação deste serviço poderia ser sensível ao contexto, beneficiada por perfil de usuário ou por inferência de atividades que o usuário costuma fazer (Madhavapeddy, 2005), ou de produtos que o usuário costuma observar e comprar, como propõe o projeto AURA da Microsoft (Smith et AL, 2004).
- **Propaganda:** Funcionalidades de gravação de conteúdo (PVR) é um fator que dificulta aceitação de propaganda por usuários. Entretanto, modelo de propaganda sensíveis a contexto é uma ferramenta poderosa para entregar anúncios personalizados baseados no contexto do usuário.

Viabilizar as funcionalidades sensíveis a contexto das aplicações acima exige implementação em vários níveis. São necessárias a aquisição de contexto por sensores ou técnicas de inteligência artificial para processar dados de contexto e transformá-los em informação de mais alto nível. Além disso, por se tratarem de aplicações distribuídas, a disseminação desta informação pelos vários elementos distribuídos é essencial para que haja, por fim, a efetiva utilização de contexto inferido.

O escopo deste trabalho não é propor ou implementar tecnologias para aquisição e interpretação de contexto. O *ContextTV* e *Context Toolkit* apresentado a seguir consiste em uma arquitetura de *middleware* extensível para lidar com as aspectos de distribuição e armazenamento de contexto de forma transparente.

3.1. Modelo de Contexto para TV Digital Interativa

Contexto pode ser descrito como informações que podem ser utilizadas para caracterizar a estrutura de uma entidade em um caso específico. Uma entidade que pode ser uma pessoa, um lugar ou um objeto relevante para qualquer tipo de interação entre usuários e aplicações, incluindo o usuário e a própria aplicação. Parâmetros, tais como:

- a) onde está o usuário;
- b) quem é o usuário;
- c) a forma como o usuário trabalha;
- d) quando a atividade está sendo feito.

A Figura 1, descreve um contexto geral que inclui informações de localização, identidade, atividade e tempo. A seguir, são descritas informações contextuais, utilizadas especificamente em aplicações na TV Digital Interativa:

- **Localização:** são informações de localização do usuário, por exemplo, dentro de sua casa. Como "usuário está na sala" ou "usuário está em dormitório", podem ser usadas para determinar a localização contextual do usuário.
- **Identidade:** no ambiente de contexto podem ser caracterizados como a identidade do dispositivo, conteúdo e identidade do usuário. Identidade do usuário é especificado pelo usuário como perfil descrito e armazenados no Set-top box (STB) em casa. A identidade do usuário tipicamente inclui informações como nome, idade, profissão, sexo, estado civil, canais favoritos, cinema, etc. A identidade do dispositivo, especifica atributos como a resolução, as características, conectividade, etc. A identidade do conteúdo é armazenado no STB como Guias de Programas Eletrônicos, após ter sido obtidas a partir da emissão televisiva. Essas Guias, também consiste de outras informações relacionadas, como categoria, tipo de canal, gênero, orientação parental, sumário, etc.
- **Atividade:** Informações sobre a atividade de contexto pode ser caracterizado pela atividade do usuário, dispositivo e do histórico do usuário / dispositivo.

Atividade do usuário, tais como "usuário assistindo televisão com a família", "usuário vendo TV ao mesmo tempo que trabalha em casa", "usuário vendo TV ao mesmo tempo fazendo serviços domésticos", etc; podem ser utilizados para elaborar as informações do contexto da atividade da entidade. Atividade do dispositivo, tipicamente inclui informações do dispositivo de informação do uso do usuário. Dispositivo do usuário e informações sobre seu histórico, normalmente inclui interação com os usuários e anúncios de programas realizados em outras datas, suas compras efetuadas, websites e páginas da web visitados, como também de canais e programas visualizados anteriormente.

- **Tempo:** São informações atuais do contexto do usuário, que possibilitam saber o comportamento do usuário ao assistir TV, a forma e opções de mudança da programação em diferentes tempo do dia, assim como de diferentes dias da semana.

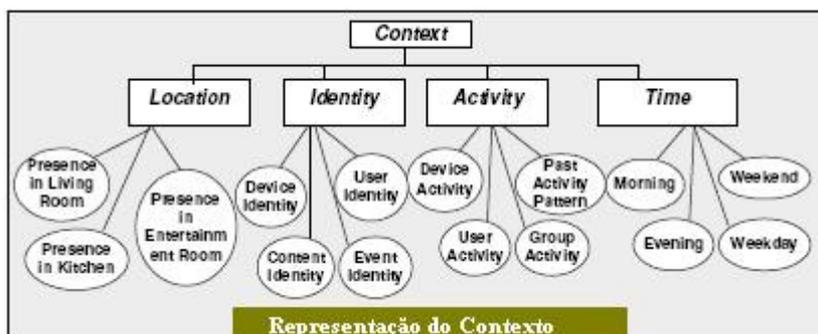


Figura 2 – Representação do Contexto

4. Arquiteturas de suporte de aplicações sensíveis ao contexto na TV Digital

Nesta seção são apresentadas duas arquiteturas para aplicações sensíveis ao contexto na TV Digital: *ContexTV* e *Context TollKit*.

4.1. *ContexTV*

ContexTV (da C.A. Neto, 2006) é uma arquitetura de *middleware* extensível para lidar com os aspectos de distribuição e armazenamento de contexto de forma transparente. Sua principal atuação é na TV Digital Móvel.

Para obter o contexto de entidades, mecanismos são implementados, geralmente referidos como “sensores”. O termo “sensor” normalmente é associado a um *hardware* especializado que obtenha, por exemplo, a localização de um dispositivo ou a presença de pessoas em um ambiente. Entretanto, “sensor”, neste artigo é tratado como uma entidade mais genérica, responsável por fazer a aquisição e representação inicial de uma situação ou característica de uma entidade, podendo inclusive ser puramente implementado por *software*. Pode-se referir a “sensores” simplesmente como “fontes de contexto” no restante do texto.

Classifica-se os diferentes sub-sistemas envolvidos em um sistema sensível ao contexto como provedores ou consumidores. Exemplificando no cenário de TV Digital convergente, a aplicação rodando no dispositivo pode ser consumidora de contexto, utilizando a localização deste e as condições climáticas no local para se adaptar, ou como provedora de contexto, enviando para outros dispositivos o *status* de seu usuário.

Do ponto de vista de um consumidor, pode-se considerar fontes de contexto de duas formas: **fontes de contexto locais** e **fontes de contexto externas**.

- **Fontes de contextos locais:** São aquelas inferidas sem a necessidade de outros componentes distribuídos. Citando alguns exemplos, pode ser um componente

de *software* capaz de obter dados diretamente de sensores no dispositivo através de APIs específicas ou um componente de GUI que permita o usuário informar explicitamente seu *status*.

- **As fontes de contexto externas:** Representam contexto que não pode ser adquirido localmente no consumidor. Podem vir de instâncias distribuídas da mesma aplicação, ou podem ser providos por outros sistemas (como as condições climáticas da cidade).

Para lidar com a distribuição e heterogeneidade de provedores e consumidores, o *ContextTV* implementa uma camada de *middleware* que lida com os detalhes de distribuição, armazenamento e acesso de contexto para a aplicação no cenário de convergência (figura 2).

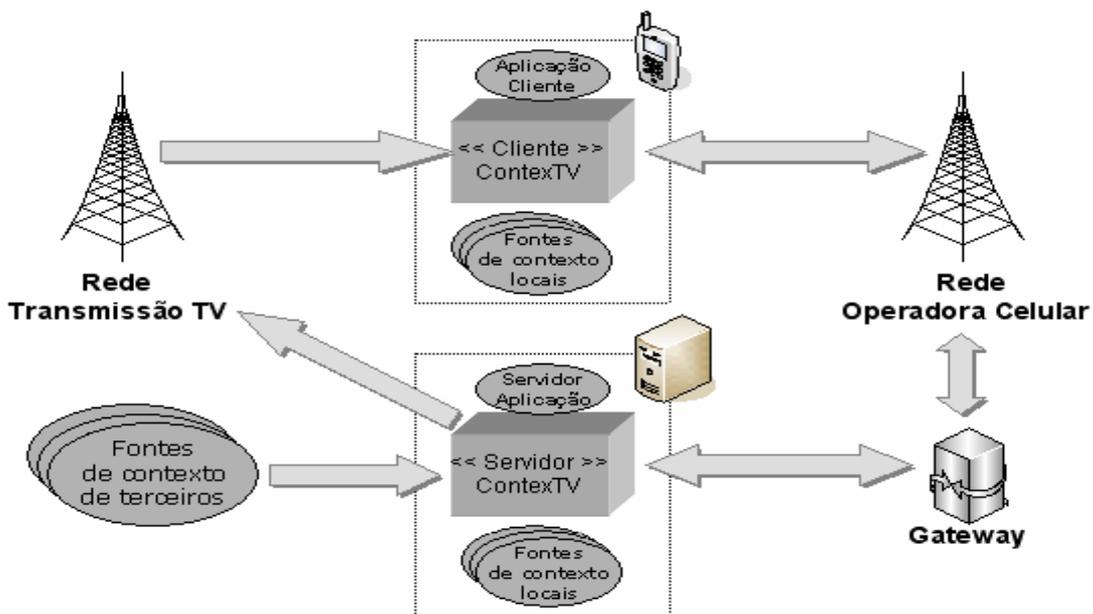


Figura 2 - Cenário de convergência e ContextTV

A seguir são detalhadas as arquiteturas do ContextTV no cliente e no servidor.

4.1.1. Cliente ContextTV

A figura 3, detalha a arquitetura do *ContextTV* no cliente. A implementação atual do *ContextTV* é feita sobre o ambiente J2ME MIDP 2.0, com as devidas extensões especificadas pela JSR 272.

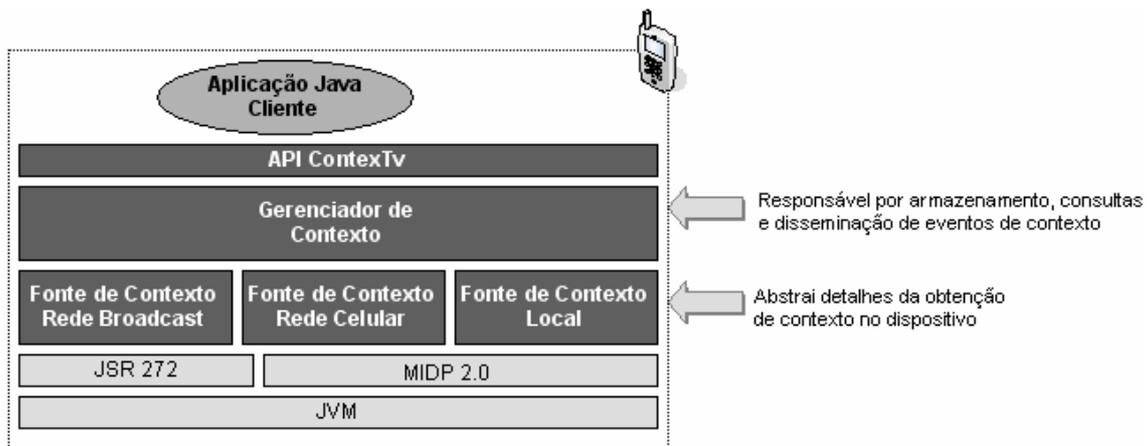


Figura 3 - Arquitetura do cliente J2ME MIDP 2.0

No dispositivo, Fontes de Contexto Locais são representadas por componentes de *software* responsáveis pela aquisição local de contexto. Seja, por exemplo, a partir de arquivos de configuração, interfaces gráficas, diretamente de sensores ou por interpretação feita localmente sobre outros contextos. No caso de fontes de contexto externas, o *ContextTV* provê dois componentes responsáveis por encapsular os detalhes de comunicação entre cliente e servidor. Considerando o cenário de convergência, os dois componentes correspondem à comunicação feita via canal de *broadcast* e feita via canal de interatividade da rede celular.

A Fonte de Contexto de Rede *Broadcast* utiliza a API provida pela JSR 272 para receber notificações de contexto. Estas notificações são recebidas em formato de arquivos XML, cujo conteúdo corresponde a mensagens no Protocolo de Intercâmbio de Contexto proposto pelo MUPE (*Multi-User Publishing Environment*) em (Suomela et al, 2004). A implementação da JSR 272 é responsável por receber os arquivos enviados por *data broadcasting*. Por sua vez a Fonte de Contexto de Rede *Broadcasting* monitora estes arquivos e é responsável pelo *parsing* das mensagens em uma representação orientada a objetos.

A Fonte de Contexto de Rede Celular implementa a comunicação entre cliente e servidor por este canal utilizando a API do *Generic Connection Framework* (GCF) da MIDP 2.0 (Ortiz, 2003). Assim como pelo canal de *broadcast*, a comunicação também é feita em mensagens do Protocolo de Intercâmbio de Contexto. Esta comunicação se dá pelas seguintes formas:

- a) Uma consulta HTTP do cliente para o servidor, sobre o estado de determinados contextos de interesse da aplicação;
- b) A recepção de eventos enviados diretamente pelo servidor. Podem ser enviados pelo uso de *Inbound Connections* (*sockets* abertos no cliente que recebem mensagens do servidor) ou por SMS enviados via rede celular.

Gerenciar fontes de contexto heterogêneas de uma forma transparente é a tarefa do Gerenciador de Contexto, cujos serviços são expostos para aplicações através da ContextTV API. As principais funcionalidades utilizadas pela aplicação são:

- a) Fazer consultas pelo estado de determinado contexto (ciência de contexto síncrona);
- b) Subscrever-se e ser notificada de eventos de mudança de estado de contexto (ciência de contexto assíncrona);
- c) Para que sejam monitoradas pelo Gerenciador de Contexto, a aplicação registra fontes de contexto locais, de rede celular ou de rede *broadcast*. O Gerenciador de Contexto guarda histórico e implementa mecanismos de *cache* para aumentar a performance e evitar ao máximo conexões ao servidor;
- d) Registrar uma fonte de contexto local como “disseminadora”. Dessa forma, o Gerenciador de Contexto no cliente utiliza o (GCF) para conectar-se ao servidor e notificá-lo de mudanças no estado do contexto adquirido por esta fonte. Essa disseminação para o servidor pode ser configurada dada que uma política de qualidade de contexto seja informada (atualizações mais frequentes em detrimento de maior uso de conexão).

4.1.2. Servidor *ContexTV*

A figura 4 detalha a arquitetura do *ContexTV* no servidor:

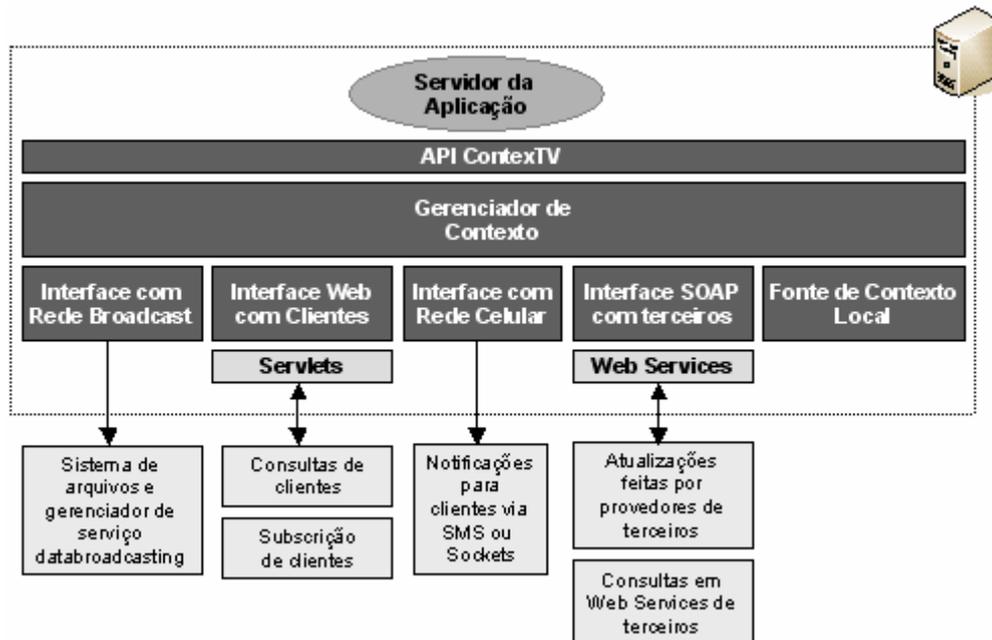


Figura 4 - Arquitetura do Servidor

Assim como no cliente, no servidor existe o Gerenciador de Contexto, cujos serviços são disponibilizados pela *ContexTV* API. A *ContexTV* API é única em clientes e servidores, e os serviços expostos para aplicação são similares em ambas as instâncias da plataforma. O tratamento dado pelo Gerenciador de Contexto a fontes de contexto locais também é semelhante a como ocorre no cliente. Dependendo de sua implementação em um ambiente real, o servidor pode utilizar como fontes de contexto, locais, serviços e funcionalidades das redes de transmissão de TV ou serviços de operadoras de rede celular, como serviços de localização via células GSM.

Considera-se na arquitetura a possibilidade entre uma integração entre gerenciador de contexto e a gerência de serviço de *datacasting* da transmissão de TV. A Interface com rede *broadcast* tem como papel gerar os arquivos XML com mensagens do protocolo de intercâmbio de contexto para enviar massivamente as notificações de mudança de contexto.

Além disso, o gerenciador de contexto no servidor conta com uma interface HTTP para receber as requisições de clientes. Considera-se também a possibilidade de uma integração a serviços das operadoras de rede celular, como o acesso a *gateway SMS* ou a permissão para abrir conexões diretas a dispositivos. Dessa forma é possível enviar notificações *unicast* assíncronas para clientes registrados para receber eventos de contexto.

Na implementação atual, fica a cargo do desenvolvedor da aplicação configurar a escolha por notificar eventos de determinados tipos de contexto via canal de *broadcast* ou por serviços de operadores de rede celular. A utilização desta funcionalidade depende da modelagem de contexto usado pela aplicação e das capacidades no ambiente real considerado para a aplicação.

Por fim, o gerenciador de contexto conta com uma interface exposta por um web service para que provedores externos de contexto possam alimentar o *ContextTV* com contexto. Além disso, a Interface (SOAP) pode conter componentes que realizem consultas em web services de terceiros (como Google, Yahoo e serviços de previsão de tempo por exemplo) e inserir esses dados de contexto na plataforma.

Por se tratar de uma arquitetura para uma plataforma de prototipação, considera-se diferentes interfaces entre consumidores e provedores de contexto para possibilitar a montagem de cenários e experimentos variados. Dessa forma é possível que uma gama maior de protótipos se *beneficie* da implementação do *ContextTV*.

4.2. Context Toolkit

O *Context Toolkit* (Dey, 2001) 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.

4.2.1. 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:

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

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: 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.

4.2.2. 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:

- a) 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;
- b) Identificar em cada entidade quais informações de contexto serão necessárias.
- c) Para cada informação de contexto de cada entidade, deve-se criar um *widget* para aquisição da informação;
- d) Para cada entidade, deve-se criar um *aggregator* para agrupar as informações coletadas pelos *widgets*;
- e) A aplicação deve obter as informações de contexto se comunicando com os *aggregators* de todas as entidades envolvidas;
- f) 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. A figura 5, detalha a arquitetura do *Context Toolkit*.

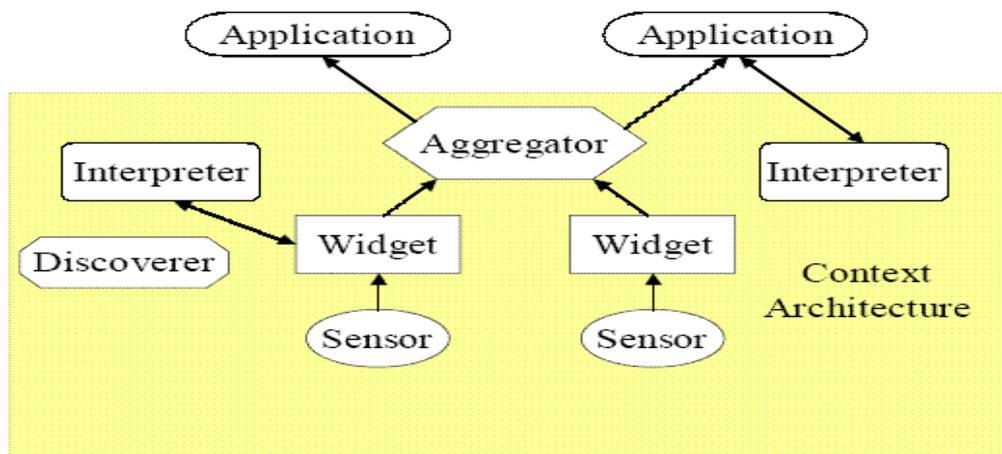


Figura 5 - Arquitetura comum ao se usar o *Context Toolkit*

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. Trabalhos relacionados

Outras propostas que elaboram sobre arquiteturas e *middlewares* com função similar ao *ContexTV* e *Context Toolkit* foram desenvolvidas nos últimos anos.

- **Middleware CASS (Fahy, 2004):** Demonstra a importância de distribuir armazenamento e processamento de contexto entre componentes de um cliente móvel e o servidor.
- **A proposta de (da Rocha e Endler, 2005):** Discute a necessidade do desenvolvedor trabalhar com uma modelagem genérica e evolutiva de contexto e propõe uma plataforma para gerenciamento de contexto em um ambiente heterogêneo.
- **ContextPhone (Raento, 2005) e o MUPE (Suomela et AL, 2004):** Apresentam vantagens de se utilizar plataformas de prototipação para construção de aplicações móveis, promovendo o desenvolvimento ágil e orientado ao usuário.

6. Considerações Finais

Como contribuições deste trabalho, pode-se citar que inicialmente foi mostrado conceitos de Computação Ubíqua, Computação Sensível ao Contexto e como pode-se explorar a Sensibilidade ao Contexto na TV Digital.

O trabalho também descreveu um modelo de Contexto para TV Digital Interativa. Esse modelo, demonstra as principais entidades envolvidas na TV Digital com a Computação Sensível ao Contexto.

Foi descrito também duas arquiteturas: *ContexTV* e *Context Toolkit*. O *ContexTV* não se propõe a ser uma solução definitiva como infra-estrutura de distribuição de contexto, apenas utiliza idéias presentes em vários trabalhos anteriores para a criação de uma plataforma de prototipação no contexto de TV Digital. O *Context Toolkit* é bastante encontrado na literatura sobre Computação Sensível ao Contexto, onde foi mostrado os problemas que ele tenta solucionar, seus componentes e como utilizá-lo.

Espera-se que este trabalho tenha servido para mostrar como a sensibilidade ao contexto pode ser um grande diferencial para as aplicações do futuro, tornando-as mais atrativas e intuitivas aos usuários.

7. Futuros Trabalhos

Com a apresentação das arquiteturas presentes neste trabalho, diversas novas pesquisas podem se beneficiar para darem continuidade na exploração da Sensibilidade ao Contexto em aplicações para TV Digital.

Este trabalho fez uma relação a tais arquiteturas, do mesmo modo, uma sugestão de trabalho futuro inclui pesquisas no *middleware* Ginga, padrão utilizado pela TV Digital Brasileira.

Referências Bibliográficas

da Rocha, R. C. and Endler, M. 2005. Evolutionary and efficient context management in heterogeneous environments. In Proceedings of the 3rd international Workshop on Middleware For Pervasive and Ad-Hoc Computing (Grenoble, France, November 28 - December 02, 2005). MPAC '05, vol. 115. ACM Press, New York, NY, 1-7.

da C.A. NETO, F.. Desenvolvimento de aplicações de tv digital interativa sensíveis a contexto auxiliado por uma infra-estrutura de middleware. M.sc. thesis, Centro de Informática, UFPE (Brazil), August 2006.

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

Dey, A.K. et al, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", HCI Journal, 16(2-4), 2001, 97-166.

Fahy, P., Clarke, S. CASS - Middleware for Mobile Context-Aware Applications. MobiSys 2004 Workshop on Context Awareness, June 2004.

JSR 272: Mobile Broadcast Service API for Handheld Terminals, disponível em <http://www.jcp.org/en/jsr/detail?id=272>, última visita em março de 2006.

K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI-00), pages 17–24, New York, 2000. ACM Press

Matsumura, K., Usui, K., Kai, K., and Ishikawa, K. 2003. Locationaware data broadcasting: an application for digital mobile broadcasting in Japan. In Proceedings of the Eleventh ACM international Conference on Multimedia (Berkeley, CA, USA, November 02 - 08, 2003). MULTIMEDIA '03. ACM Press, New York, NY, 271-274.

Ortiz C. E., The Generic Connection Framework. <http://developers.sun.com/techttopics/mobility/midp/articles/genericframework/> (última visita em março de 2006)

Raento M., Oulasvirta A., Petit R., Toivonen H., "ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications," IEEE Pervasive Computing, vol. 04, no. 2, pp. 51-59, Apr-Jun, 2005.

Suomela, R., Räsänen, E., Koivisto, A., Mattila, J.: Open-Source Game Development with the Multi-user Publishing Environment (MUPE) Application Platform. In Rauterberg, M., ed.: Entertainment Computing ICEC 2004: Third International Conference, Eindhoven, The Netherlands, September 1-3, 2004. Proceedings. (2004) 308–320

S. Tamminen, A. Oulasvirta, K. Toiskallio, A. Kankainen, "Understanding mobile contexts", Personal and ubiquitous computing, Volume 8, Issue 2, May 2004, Pages 135 – 143.

Weiser, Mark. The Computer for the Twenty-First Century. Scientific American, 1991.