

2

Conceitos Preliminares

Este capítulo descreve conceitos básicos relacionados à disseminação de conteúdo adaptado para clientes móveis, com enfoque em adaptação dinâmica de conteúdo, ciência de contexto e comunicação publish/subscribe. Além disso, é apresentada a MoCA, uma arquitetura de provisão de contexto.

O capítulo é organizado como segue. A Seção 2.1 descreve as técnicas de adaptação dinâmica de conteúdo usadas em sistemas sensíveis a contexto. A Seção 2.2 apresenta as definições sobre contexto para ambiente móvel, e descreve as diferentes formas de modelagem das informações de contexto. A Seção 2.3 provê uma visão geral do estilo de interação publish/subscribe e suas características: descreve o padrão de comunicação, define a noção de notificação e de registro de interesse, descreve os esquemas existentes para definição desses registros, discute as arquiteturas de interconexão e o tratamento de mobilidade. Por fim, a Seção 2.4 apresenta a arquitetura MoCA, desenvolvida no grupo de pesquisa do LAC da PUC-Rio com objetivo de auxiliar o desenvolvimento de aplicações sensíveis ao contexto, e que fornece serviços de coleta e processamento de informações de contexto de dispositivos móveis.

2.1

Adaptação

A capacidade de uma aplicação se adaptar ou ser adaptada a mudanças do ambiente é uma importante característica para ciência da computação, notavelmente para ambientes móveis e pervasivos. A adaptação em aplicações sensíveis ao contexto pode ocorrer em diversos domínios: *adaptação de conteúdo*, ou dos dados de saída, às necessidades e capacidades do cliente; *adaptação de comportamento* da aplicação em função de mudanças no ambiente, como por exemplo, reduzir consumo de recursos; *adaptação da interface* com o usuário, na qual a interface humano-computador pode se alterar em função do dispositivo usado (transformações de layout, ou em modos de interação) ou ser transferida de um dispositivo para outro conforme a necessidade; *adaptação da interação* da aplicação com outras entidades (aplicações,

objetos, elementos de rede), por exemplo na decisão de *handoff*; e adaptação da interface de programação (API).

A ênfase desta tese é o suporte ao desenvolvimento de aplicações com adaptação de conteúdo sensível a contexto, e por isso, este tipo de adaptação é detalhado a seguir.

2.1.1

Adaptação de Conteúdo

No âmbito desta tese, adaptação de conteúdo consiste de técnicas que transformam o conteúdo de acordo com as condições do enlace sem fio (qualidade) e restrições dos dispositivos (tipo de saída, poder computacional, etc). Seu objetivo é prover o acesso a informações independente do dispositivo, a transmissão otimizada e a apresentação adequada de conteúdo aos clientes móveis.

Várias tecnologias de adaptação de mídias podem ser usadas para transformar a informação de forma a aumentar a sensibilidade dos dados e melhorar a experiência dos clientes num ambiente de rede heterogênea. A Tabela 2.1 lista algumas das possíveis adaptações de conteúdo para diferentes tipos de mídia.

Texto	Imagem	Áudio	Vídeo
Resumo do texto	Remoção de imagem	Remoção do áudio	Remoção do vídeo
Transformação do texto para áudio	Redução de tamanho	Stereo para mono	Vídeo para imagem ou áudio ou texto
Conversão de formato	Colorido para escala de cinza	Codificação áudio para texto	Conversão de formato
Tabela para lista	Conversão de formato	Conversão de formatos	Redução da taxa de quadros
Redução do tamanho da fonte	Redução da qualidade	Redução taxa de amostragem	Redução da resolução
			Extração de miniaturas

Tabela 2.1: Tipos de mídia e adaptações de conteúdo relacionadas

A maioria das adaptações de conteúdo resultam em redução dos dados, e conseqüentemente, a redução de carga nos componentes do sistema, mas nem toda estratégia de adaptação pode ser usada para todos os tipos de mídia. A estratégia de adaptação é altamente específica para o tipo de mídia e em alguns casos pode ser dependente do conteúdo.

Técnicas de Adaptação

Uma das técnicas mais utilizadas em aplicações com adaptação dinâmica (sob demanda) de conteúdo é a técnica de *transcoding*. Essa técnica engloba várias propostas para adaptação de diferentes tipos de dados, e pode ser dividida em subcategorias: destilação, refinamento, sumarização, dentre outras [30, 31, 32].

A **destilação** (*distillation*) é um processo que utiliza a compressão com perdas, executado em tempo de execução, para tipos de dados específicos, que elimina informações redundantes ou desnecessárias, mas que preserva a semântica do conteúdo. É um termo geral para várias formas de compressão de dados, que podem ou não ser baseadas em padrões de codificação. Um exemplo para imagens é o JPEG, cuja taxa de compressão pode ser controlada de acordo com a qualidade desejada. Exemplo de destilação não baseado na codificação, para imagens, é reescalar uma imagem em cada eixo ou recortá-la (*cropping*), reduzindo seu tamanho e representação binária. Para textos, exemplos de destilação incluem a formatação, compressão e conversão entre codificações (p.ex: PS para Rich-text), e para vídeos, pode-se ter destilações temporais (limitar taxa de quadros, condensar o tempo) ou espaciais (alteração da resolução, requantização, redução de cores) [15].

Alternativamente, pode-se desejar ver parte do conteúdo original com alta precisão, por exemplo, zoom em parte de um gráfico ou mapa, ou renderizar apenas uma página particular de um documento PostScript. O termo **refinamento** (*refinement*) é usado para se referir ao processo de selecionar partes de um documento em sua qualidade original. Assim como a destilação, é específico para cada tipo de dado, e pode ser aplicado ortogonalmente à destilação. Pode-se utilizar também um refinamento progressivo, de forma que diferentes níveis de qualidade possam ser providos durante a entrega dos dados. Isso pode ser obtido através da utilização de esquemas especiais de codificação, tais como codificação em camadas [33], [34] e compressão em multi-resolução.

A **sumarização** (*summarization*) é um tipo de compressão onde partes específicas do conteúdo original contendo as informações mais relevantes são selecionadas. Esta técnica propicia uma alta remoção de informação, mas com a mínima perda possível da compreensão da mensagem, e é mais utilizada para textos e vídeos. A sumarização de texto é uma técnica com origem nos anos 60, e que teve um novo impulso recentemente com as tecnologias Web para dispositivos móveis. Ela visa a extração de pequenas e não-redundantes porções de um texto, e utiliza para isso métodos estatísticos, lingüísticos e heurísticos [35]. A sumarização de vídeo consiste na apresentação de uma seqüência de imagens, com ou sem áudio, com versão concisa do conteúdo, e mantendo a

mensagem essencial, que permita uma rápida obtenção de conteúdo de forma compreensível, mesmo em conexões de baixa vazão. O sumário do vídeo pode ser uma versão curta de um vídeo montado com segmentos importantes, ou uma série de pequenos cliques, ou sua sinopse (*trailer*) [36].

A **trans-codificação** (*data transcoding*) é o processo geral de conversão do conteúdo entre diferentes formatos para adequá-lo às capacidades do dispositivo, e otimizar sua transmissão sobre o enlace sem fio; frequentemente com perda de fidelidade. Pode ser usada para adaptar conteúdo de áudio, vídeo e texto, estaticamente ou *on-the-fly*, através de métodos de compressão, conversão, transformação, re-formatação, filtragem, redução, re-codificação de conteúdo. Alguns exemplos de trans-codificação de dados são: conversão de formatos de vídeo (MPEG para H.263), conversão de formato de áudio (WAV para MP3), conversão de imagens (BMP para WBMP, GIF para JPEG), transformação de textos (PDF para HTML), ajuste de HTML para restrições do dispositivo (HTML para WML).

Um outro tipo de adaptação é a **transformação cross-media** [12]. Esse tipo de adaptação permite a substituição dos conteúdos por diferentes alternativas, variando o tipo de mídia. Por exemplo, alguns computadores móveis não são muito adequados para receberem transmissões de vídeo, dadas as suas restrições de hardware e software. Assim, para tornar a informação contida nesse vídeo acessível a esses dispositivos, o vídeo pode ser transformado em (ou substituído por) um conjunto de imagens sem áudio (e com legenda).

2.2

Ciência de Contexto

A ciência de contexto está relacionada com a habilidade dos sistemas computacionais obter vantagem das informações ou condições existentes em um ambiente dinâmico para adicionar valor aos serviços ou executar tarefas mais complexas. O primeiro trabalho em computação sensível a contexto foi o *Active Badge* [37], que utilizava informações de localização (baseado na tecnologia de infravermelho) de usuários para realizar transferências automáticas de chamadas telefônicas. Entretanto, o termo *context-aware* (ciência de contexto ou sensibilidade ao contexto) apareceu pela primeira vez em [38]. Nos últimos anos, a computação sensível ao contexto tem recebido maior atenção, principalmente em função do desenvolvimento da computação móvel e do aparecimento de uma nova geração de dispositivos móveis. Porém, a construção do suporte à sensibilidade ao contexto para as aplicações apresenta inúmeros desafios, os quais se relacionam especialmente a obtenção, modelagem, armazenamento, distribuição e monitoramento do contexto.

2.2.1

Definição de contexto

Na literatura é possível encontrar uma variedade de trabalhos baseados em contexto, cada qual com sua definição para este termo. Schilit et al. [10] refere-se a contexto como localização, identificação de pessoas e objetos, situações sociais e condições ambientais. Eles definem três aspectos importantes de contexto: onde o usuário está, com quem ele está e quais os recursos próximos a ele. Pascoe [39] define contexto como o subconjunto de estados físico e conceitual de interesse a uma entidade particular. Chen e Kotz [9] definem contexto como o conjunto de estados e configurações do ambiente que: ou determinam o comportamento da aplicação, ou causam a ocorrência de um evento específico da aplicação que é relevante para o usuário. Dey [8] elaborou uma definição mais genérica, a qual vem sendo utilizada em várias pesquisas recentes:

“Contexto é toda a informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto que é considerado relevante à interação entre um usuário e uma aplicação, incluindo o usuário e suas próprias aplicações.”

Essa definição deixa a cargo do projetista da aplicação decidir quais informações podem ser contexto. A literatura ([9, 8, 10, 11]) sugere que existem duas categorias principais de informações de contexto: fatores físico e humano. A primeira foi dividida dentro de três sub-grupos: contexto do ambiente (por exemplo, luz, temperatura, hora, etc.), infra-estrutura computacional (por exemplo, recursos do dispositivo e da comunicação, etc.), e localização (por exemplo, coordenadas relativas e absolutas, etc.). Fatores humanos também foram classificados em sub-grupos: informação pessoal (por exemplo, preferências, interesses, habilidades, etc.), relacionamento social (grupo, família), e atividade (por exemplo, ocupado, descansando, trabalhando, etc.).

Nesta tese, são considerados principalmente aspectos de infra-estrutura computacional, localização e informações dos usuários. O *contexto* (ou situação de contexto) é definido como uma combinação de *atributos de contexto*, que são as unidades de informação de contexto relativas ao enlace sem fio, dispositivo, etc., tais como, qualidade do canal de comunicação, CPU, largura da tela.

2.2.2

Modelagem de Contexto

Um modelo eficiente para manipulação, compartilhamento e armazenamento de dados de contexto é essencial para os sistemas sensíveis ao contexto. As principais abordagens para definição e modelagem de contexto, baseadas nas estruturas de dados utilizadas para representação e troca das informações contextuais, segundo Strang [40] e Balakrishnan [41], são:

- ▷ **Modelos de pares objeto-valor.** Estes modelos representam a estrutura de dados mais simples para modelar o contexto, e foi primeiramente proposto por Schilit [38]. Os pares objeto-valor são usados para descrever as características dos dados de contexto, e pode-se usar algoritmos recursivos de casamento de padrões para se consultar as informações desejadas.
- ▷ **Modelos de esquemas Markup.** Todos os modelos baseados em linguagem markup usam estruturas de dados hierárquicas consistindo de *tags* com atributos e conteúdo. Adicionalmente, é possível gerar gatilhos (*triggers*) baseados em condições especificadas nas *tags*. Esta abordagem é tipicamente utilizada para modelar perfis já que ela reflete naturalmente a estrutura das informações de um perfil. Os exemplos típicos para tais perfis são o CC/PP (*Composite Capabilities/Preference Profile*) [42, 43], o UAProf (*User Agent Profile*) [2] e o CSCP (*Comprehensive Structured Context Profiles*) [44].
- ▷ **Modelos de Linguagens de Modelagem.** Um instrumento de modelagem gráfica popular é a UML (*Unified Modeling Language*), que devido sua estrutura genérica, pode também ser utilizada para modelar contextos, como em [45]. Um outro exemplo é a CML (*Context Modeling Language*) [46], que provê uma notação gráfica para descrever tipos de informação, suas fontes, especificar dependências e restrições, bem como qualidade. Modelos desenvolvidos usando esta abordagem podem ser facilmente mapeados para banco de dados relacionais.
- ▷ **Modelos Orientados a Objetos.** Esses modelos herdam as idéias do paradigma de orientação a objetos, tais como encapsulamento, herança e reusabilidade. Nestes modelos, os detalhes da coleta, do processamento e da representação dos dados são encapsulados e, portanto, escondidos de outros componentes. O acesso à informação de contexto é fornecido através de interfaces bem-definidas. Exemplos de projetos que utilizam esta abordagem são GUIDE [47] e Hydrogen [48].

- ▷ **Modelos Baseados em Regras.** São modelos formais que expressam a informação contextual como fatos em um sistema baseado em regras. Nestes sistemas, um conjunto de regras define as condições que devem ser satisfeitas para que uma certa conclusão seja atingida. O processo de inferência (ou *reasoning*) pode ser usado derivar novos fatos baseados em regras existentes nos sistemas. Apesar de lógica ser às vezes usada para modelagem de contexto (como em [49]), ela é principalmente utilizada em conjunto com outras abordagens, como um mecanismo de inferência de contexto.
- ▷ **Modelos Baseados em Ontologias.** Ontologias representam uma descrição dos conceitos, suas propriedades e seus relacionamentos. Conseqüentemente, as ontologias são um instrumento muito promissor para modelar a informação de contexto devido a sua elevada expressividade e à da possibilidade de aplicar técnicas sofisticadas de inferência de ontologias. O Context Ontology (CONON) [50] é um exemplo de modelo de contexto baseado em ontologias.

Alguns atributos utilizados por Strang e Linnhoff-Popien [40] para avaliação dos modelos de contexto incluem: possibilidade de composição distribuída (cd), capacidade para validar conhecimento parcial (vp), suporte à informação com distintos níveis de qualidade (qua), tratamento de informação incompleta e ambígua (inc), nível de formalidade (for), e aplicabilidade em ambientes já existentes (app). A Tabela 2.2 apresenta um resumo dos modelos de contexto em relação a estes atributos.

Modelos	cd	vp	qua	inc	for	app
Modelos de pares objeto-valor	-	-	-	-	-	+
Modelos de esquemas Markup	+	++	-	-	+	++
Modelos de Linguagens de Modelagem	-	-	+	-	+	+
Modelos Orientados a Objetos	++	+	+	+	+	+
Modelos Baseados em Regras	++	-	-	-	++	-
Modelos Baseados em Ontologias	++	++	+	+	++	+

Tabela 2.2: Modelos de Contexto

2.2.3

Sistemas sensíveis a contexto

“Um sistema é sensível a contexto se ele utiliza contexto para fornecer informação e/ou serviço relevante para o usuário, onde a relevância depende da tarefa do usuário.” [8]

Encontram-se na literatura diversas propostas de categorização desses sistemas. Entretanto, pode-se agrupar as aplicações sensíveis a contexto de acordo com suas características, nas seguintes categorias: apresentação das informações de contexto ao usuário, associação de informação de contexto a dados, descoberta de recursos ou serviços, e adaptação de serviços ou ativação de ações acionadas por mudanças de contexto. Essa última categoria de aplicações é o foco desta tese.

Além disso, sistemas sensíveis a contexto podem ser implementados de diversas formas, mas três principais abordagens para arquiteturas desse tipo de aplicações podem ser destacadas [51, 6]. Na abordagem do tipo *acesso direto aos sensores*, como o próprio nome diz, as aplicações clientes coletam a informação desejada diretamente dos sensores. Nos sistemas *baseados em middleware*, uma arquitetura em camadas esconde das aplicações clientes os detalhes de baixo nível da interação com os sensores. Finalmente, na abordagem baseada em *servidores de contexto*, serviços de notificação estendem o middleware para permitir acesso múltiplo e concorrente às informações de contexto. Os trabalhos mais recentes na área de aplicações sensíveis a contexto têm adotado justamente uma abordagem que assume o uso de uma infra-estrutura capaz de coletar, gerenciar e disseminar informações de contexto, para assim reduzir a complexidade exigida no desenvolvimento dessas aplicações [46].

2.3

Comunicação Publish/Subscribe

2.3.1

Definição

A comunicação publish/subscribe permite um modelo de interação assíncrono entre as partes, o que a torna adequada a aplicações de disseminação de conteúdo. Esse tipo de comunicação é baseado na troca de *notificações*, conhecidas também como *eventos*, que representam os itens de informação e contêm o conteúdo publicado. O modelo básico de interação publish/subscribe, apresentado na Figura 2.1¹, se baseia em um serviço de notificação de eventos (infra-estrutura Pub/Sub) que representa um mediador entre os publicadores (*publishers*), que atuam como produtores de eventos, e os assinantes (*subscribers*), que são os consumidores de eventos. Os assinantes registram seu interesse em receber notificações de eventos invocando o método **subscribe** no serviço de eventos. O método **unsubscribe** é usado para finalizar o registro de inte-

¹Ilustração obtida em [52]

resse. O serviço de eventos armazena os registros de interesse (*subscription*) dos assinantes, e os utiliza para comparação com eventos publicados. Para gerar um evento, um publicador invoca o método **publish**. O serviço de eventos compara o evento com os registros existentes e propaga o evento para todos os assinantes interessados, usando o método **notify**. Se um assinante não puder ser contactado, o evento notificado ou é descartado, ou é armazenado para ser entregue quando o assinante estiver disponível, dependendo da política de persistência de publicação adotada. O objetivo principal deste tipo de sistema é, conseqüentemente, suportar a filtragem e a entrega de eventos de uma maneira eficiente e confiável.

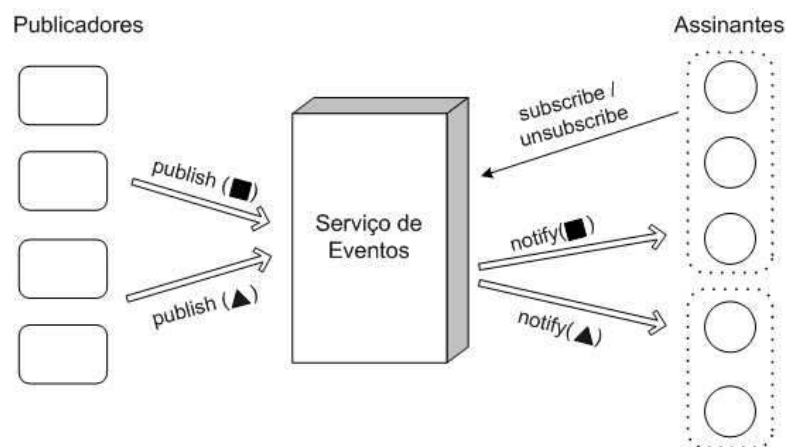


Figura 2.1: Modelo de interação publish/subscribe

A principal característica da interação publish/subscribe é o desacoplamento espacial e temporal dos participantes da comunicação. Como eles interagem através de um intermediário, não necessitam se conhecer, e nem estar ativos simultaneamente para a troca de mensagens, ficando a cargo da infraestrutura Pub/Sub o armazenamento e distribuição das mensagens. Essa comunicação assíncrona, desacoplada e persistente é importante para os dispositivos móveis que estão freqüentemente indisponíveis devido à conexão intermitente ou à potência limitada da bateria, e à mobilidade [25, 26]. Além disso, o modelo Pub/Sub suporta extensibilidade do sistema. A adição de um novo publicador ou de um assinante não afeta a funcionalidade do sistema, característica que é desejável ao lidar com as mudanças freqüentes no ambiente móvel.

Os sistemas publish/subscribe podem ser implementados de várias maneiras, cada uma delas mais apropriada a um tipo de aplicação. Duas características principais a se observar são: a forma como os assinantes especificam os eventos em que estão interessados (esquema de registro de interesse) e a ar-

quietura do serviço de eventos ou infra-estrutura de envio (ou seja, a topologia de interconexão). Estes dois aspectos são discutidos a seguir.

2.3.2

Esquemas de definição de registros de interesse

O modelo de interação Pub/Sub permite que clientes especifiquem as características dos eventos que desejam receber, através de registros de interesse que são usados para filtrar os eventos de acordo com a necessidade dos assinantes. Os registros contém regras que permitem ao serviço Pub/Sub comparar cada evento publicado com o interesse de cada assinante, e entregar apenas as notificações de eventos que obedeçam ao critério definido no registro. Os principais esquemas de definição de registros de interesse (e filtragem de eventos) utilizados pelos sistemas Pub/Sub são baseados em tópicos, em conteúdo ou em tipos.

O **esquema baseado em tópicos** foi o primeiro esquema utilizado por sistemas Pub/Sub, e é baseado na noção de *tópicos* ou *assuntos*. Os participantes podem publicar eventos e registrar interesse em tópicos, identificados por palavras-chave. Na prática, o esquema provê abstrações que mapeiam tópicos em canais de comunicação distintos. Os tópicos podem também ser organizados de forma hierárquica, de tal maneira que clientes que se registram em um certo o tópico são notificados de eventos que pertencem àquele tópico e a todos os tópicos abaixo na hierarquia, como no sistema TIBCO Rendezvous [53].

O **esquema baseado em conteúdo** utiliza uma abordagem em que os registros de interesse são descritos em termos do conteúdo ou das propriedades dos eventos publicados. Estas propriedades podem ser atributos internos das estruturas de dados dos eventos, como em Siena [54], Elvin [55, 56] e Jedi [57], ou meta-dados associados aos eventos, como em JMS [58]. O registro de interesse especifica filtros, através de uma linguagem própria. Os filtros definem restrições, geralmente na forma de pares *nome-valor* das propriedades e operadores de comparação ($=$, $<$, \leq , $>$, \geq), que identificam os eventos válidos. Estas restrições podem ser logicamente combinadas (**or**, **and**, etc) para formar um padrões complexos de registros de interesse. A Figura 2.2 apresenta um exemplo de registro de interesse e de um evento que o satisfaz, para um sistema baseado em conteúdo.

O esquema baseado em conteúdo provê maior expressividade aos registros de interesse do que o esquema baseado em tópicos, pois permite definir qualquer consulta sobre o conteúdo dos eventos. Mas, isso implica em uma maior complexidade para 'combinar' eventos e registros. Assim, um desafio é

manufacturer = "Maserati"	manufacturer = "Maserati"
model = "A6 GCS" OR "Mexico 3300"	model = "A6 GCS"
price < 500.000	price = 499.000

2.2(a): Registro de interesse

2.2(b): Evento válido

Figura 2.2: Registro de interesse em um sistema Pub/Sub baseado em conteúdo e um evento que o satisfaz.

o compromisso entre expressividade e eficiência e escalabilidade.

O **esquema baseado em tipos** é um esquema estático de classificação que se assemelha à abordagem baseada em tópicos, mas utiliza tipos das linguagens orientadas a objetos para filtrar os eventos de interesse [59]. O uso de tipos permite uma integração mais próxima entre o middleware e a linguagem de programação. Os eventos são modelados como objetos, e cada objeto tem um tipo associado. A noção de herança da orientação a objetos, também é utilizada. Assim, assinar a um tipo implica que o assinante recebe todos os eventos da classe que implementa o tipo, bem como os eventos de suas subclasses. O processo de *matching* entre um evento e um tipo apenas verifica se o objeto do evento é uma instância de um tipo definido. Quando comparado ao esquema baseado em conteúdo, a abordagem baseada em tipo garante o encapsulamento dos dados dentro dos objetos de eventos o que é violado em sistemas baseados em conteúdo.

É possível também combinar o tanto o esquema de registro baseado em tópicos como em tipos com o esquema baseado em conteúdo: a classificação de estática dos esquemas baseados em tópicos ou tipos é expandida (aumento em expressividade) pela natureza dinâmica do esquema baseado em conteúdo.

2.3.3

Arquitetura

A implementação de um serviço de notificação de eventos utilizando publish/subscribe pode seguir um abordagem *centralizada* [55] onde um único elemento é responsável por manter o registro de interesse dos assinantes e por disseminar os eventos que são gerados. Essa arquitetura, apesar da simplicidade, apresenta problemas de escalabilidade.

Em outra abordagem, adotada em [60, 61], a comunicação assíncrona é realizada através de primitivas que implementam mecanismos de envio não bloqueantes, tanto em publicadores como em assinantes, sem a necessidade de um intermediário. Nesta arquitetura *descentralizada*, todos os nós são homogêneos, têm um poder de processamento razoável, e estão aptos a

gerenciar os registros de interesse e disponibilizar funções Pub/Sub.

O problema de escalabilidade do sistema também pode ser resolvido através de uma arquitetura distribuída que compreenda uma rede de servidores especiais, denominados *brokers*. Essa arquitetura de nós heterogêneos utiliza *brokers* fixos de alto poder computacional e clientes (produtores e assinantes) possivelmente móveis e menos poderosos. Cada broker é responsável por prover o serviço Pub/Sub para parte dos clientes, e por executar os protocolos necessários para persistência, confiabilidade, ou alta-disponibilidade, bem como filtragem e roteamento de eventos baseado em conteúdo.

Essa *rede de brokers distribuídos* pode ser interligada sob diferentes topologias. Na topologia hierárquica, como em [57], os clientes podem se conectar a qualquer nó e seus registros de interesse são propagados pela árvore de brokers para os níveis mais altos, até a raiz. Entretanto, essa topologia tende a sobrecarregar o nó raiz, e a falha de um broker pode desconectar toda uma subárvore. Em uma topologia *peer-to-peer* [62, 54] os brokers que fazem parte do serviço de notificação são interligados formando um grafo não dirigido, e são empregados algoritmos eficientes para a filtragem e a propagação (roteamento) dos eventos. A eficiência do modelo distribuído depende de como os brokers são interligados, como os eventos são propagados e a estratégia de processamento dos interesses dos assinantes [54].

Além disso, a transmissão dos dados pode ser implementada de várias formas; em particular, pode-se usar comunicação ponto-a-ponto, ou mecanismos de *multicast*. A escolha do mecanismo de comunicação depende de diversos fatores, como a rede e a arquitetura do sistema.

2.3.4

Suporte à Mobilidade

Em comunicação, mobilidade é a habilidade dos clientes moverem-se através de diferentes redes (domínios, ou redes com tecnologias de comunicação diferentes) durante a utilização de algum serviço. A maioria dos sistemas publish/subscribe existentes foram projetados para ambientes estacionários, e as operações relacionadas a mobilidade são tratadas pela aplicação. Por exemplo, a aplicação deve se encarregar de efetuar o cancelamento de registro no sistema Pub/Sub antes da desconexão e novamente se registrar após a reconexão, mas neste caso o assinante não receberá eventos publicados durante o tempo de desconexão.

Em [29, 62], argumenta-se que o próprio middleware Pub/Sub deve oferecer suporte à mobilidade e assegurar reconexões transparentes aos clientes e que preservem as notificações publicadas durante a desconexão. Os problemas

relacionados à mobilidade têm sido tratados recentemente [63, 64, 62]. Boa parte das soluções propostas para a mobilidade do cliente em middlewares Pub/Sub não são otimizadas para ambientes móveis, mas apenas estendem os sistemas estacionários existentes [63, 64].

A solução mais comum é baseada na abordagem de “enfileiramento” de eventos, na qual um *broker* do sistema (geralmente o último broker que serviu um assinante) atua como um proxy do assinante durante sua desconexão. Este proxy armazena as notificações publicadas durante a desconexão do assinante em uma fila especial, e as entrega após a reconexão deste assinante. Se o assinante se reconectar ao sistema através de um novo broker, é executado um procedimento de *handoff* que transfere as notificações armazenadas ao assinante e atualiza as rotas da rede de *brokers*.

Outra abordagem é o uso de notificações persistentes [29, 58], que ao invés de armazenar as notificações em filas especiais para cada cliente, a rede de *brokers* armazena as notificações persistentes até seu prazo de validade expirar. Os publicadores devem definir o período de validade das notificações, e o sistema deve garantir o armazenamento das notificações durante seu período de validade, e entregar notificações válidas aos assinantes quando estes se reconectam ao sistema publish/subscribe. Se o assinante se reconectar após a expiração da notificação, esta não é entregue pois se considera que ela não possui mais informações válidas.

2.4 MoCA

A arquitetura MoCA (*Mobile Collaboration Architecture*) [65, 66] foi desenvolvida com a finalidade de oferecer suporte ao desenvolvimento e execução de aplicações distribuídas sensíveis ao contexto, particularmente aquelas que envolvem dispositivos móveis interconectados através de redes Wireless LAN infra-estruturadas (IEEE 802.11b/g). Os serviços disponibilizados pela MoCA provêem meios para coletar, armazenar, processar e difundir informações de contexto obtidas a partir dos dispositivos móveis que interagem entre si em uma rede sem fio e com computadores da rede fixa. Além disso, é fornecido um conjunto de API's para o desenvolvimento de aplicações que interagem com esses serviços como consumidores de informações de contexto.

Os serviços da MoCA, ilustrados na Figura 2.3, são brevemente descritos a seguir. Maiores detalhes sobre como esses interagem com os demais serviços da arquitetura podem ser encontrados em [67].

O *Monitor* é um *daemon* executado em cada dispositivo móvel que tem o objetivo de coletar dados brutos de contexto do dispositivo (tais como

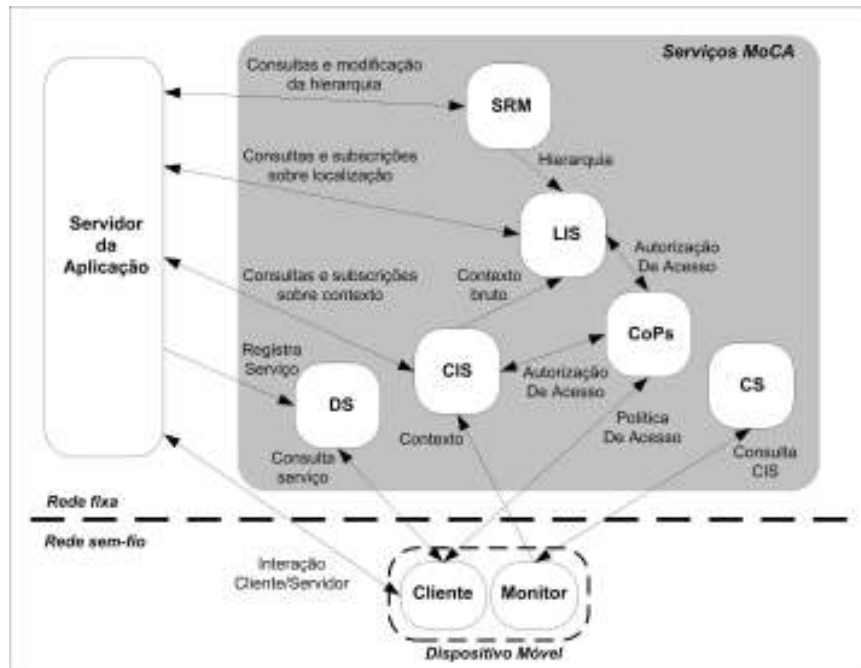


Figura 2.3: Arquitetura MoCA

percentual de uso da CPU, memória disponível, nível de energia da bateria) e da rede sem fio (AP corrente, sinal RF) e, enviá-las periodicamente para o CIS (*Context Information Service*). Até o momento temos uma implementação completa do Monitor para Windows XP e Windows CE [68], e um protótipo para Linux.

A *Event-based Communication Interface* (ECI) é utilizada para implementar comunicação assíncrona do tipo Publish/Subscribe. É baseada em tópicos combinado com um esquema de filtragem baseado em propriedades do evento, o que permite que qualquer aplicação ou serviço torne-se um publicador de eventos ou que aceite subscrições de notificações relativas à mudança de propriedades de contexto associadas a um tópico. Por exemplo, na mudança da localização de algum dispositivo, o tópico definido pode ser o endereço MAC do dispositivo, e a propriedade de interesse a localização.

O *Context Information Service* (CIS) é um serviço distribuído no qual cada servidor CIS recebe e processa dados brutos de contexto enviados por monitores de cada dispositivo móvel. O CIS utiliza as especificações CC/PP [43] e UAProf [2] como fonte de informação das características estáticas dos dispositivos. O CC/PP especifica uma estrutura geral de um documento XML para descrever as características dos dispositivos e dos usuários. O Open Mobile Alliance [3] criou a especificação UAProf com o objetivo de padronizar os vocabulários CC/PP. Para cada dispositivo criado pelos fabricantes, um documento UAProf é criado e armazenado no repositório Web no site do

fabricante. Esse documento contém informações como características da tela do dispositivo, informações sobre sistema operacional e softwares disponíveis. Na MoCA, permite-se também inclusão informações adicionais às definidas no UAProf, através da extensão deste vocabulário, como definido no apêndice A.

A aquisição e manutenção de informações de contexto dos clientes móveis no CIS é ilustrada na Figura 2.4.

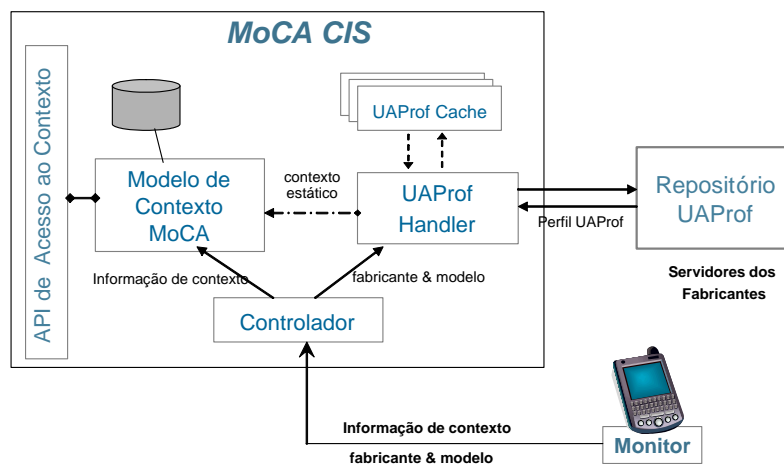


Figura 2.4: Obtenção de contexto no CIS

Como já mencionado, o Monitor coleta as informações dinâmicas sobre os dispositivos e as envia ao CIS. No CIS, estas informações são convertidas para o Modelo de Contexto da MoCA² e armazenadas em um banco de dados local. O Monitor também envia ao CIS as informações sobre o fabricante e o modelo do dispositivo móvel. O Controlador repassa essas informações (modelo e fabricante) ao UAProfHandler, que utiliza o UAProf Cache para checar se o documento UAProf que descreve as propriedades estáticas do dispositivo já se encontra no cache. Caso não seja encontrado, o download do documento é realizado junto ao sítio dos fabricantes. O uso de cache é necessário para evitar downloads repetidos dos documentos UAProf que podem causar atrasos no processo de obtenção do contexto, e sua utilização. O UAProfHandler, de posse do documento UAProf, o interpreta e repassa as informações estáticas ao gerenciador do Modelo de Contexto Moca, que as adequa ao modelo e as armazena no banco local.

As aplicações podem consultar informações de contexto de forma síncrona ou assíncrona, através da API de Acesso ao Contexto do CIS. Por meio de consultas síncronas, aplicações podem solicitar informações atualizadas sobre o contexto de um determinado dispositivo. Através de consultas

²Este modelo organiza as informações de contexto em uma estrutura hierárquica, descrita em [69]

assíncronas, aplicações podem registrar interesse em situações de contexto específicas, descritas por expressões lógicas (de sintaxe semelhante a SQL), envolvendo diversos atributos de contexto de um dispositivo. Por exemplo, para uma aplicação registrar o interesse em ser notificada quando o nível de energia da bateria de um PDA ficar abaixo de 30%, pode-se usar a seguinte expressão de interesse: “`Device.LocalResources.EnergyLevel > 30 AND DeviceProfile.HardwarePlatform.DeviceType = PDA`”. A Tabela 2.4 enumera algumas das informações de contexto providas pelo CIS. O CIS utiliza a ECI para aceitar registros de interesse, e para notificar a ocorrência da condição de contexto especificada.

Tag	Descrição
Device.LocalResources.CpuUsage	Uso da CPU (0 a 100%)
Device.LocalResources.EnergyLevel	Nível de energia disponível (0 a 100%)
Device.LocalResources.FreeMemory	Total de memória disponível (kB)
Device.Connectivity.CurrentApMacAddr	Endereço MAC do ponto de acesso
Device.Connectivity.OnLine	True se o dispositivo está conectado
Device.Connectivity.ApChange	Verdadeiro se o dispositivo muda o AP
DeviceProfile.HardwarePlatform.DeviceType	Tipo do dispositivo, ex: PDA
DeviceProfile.HardwarePlatform.ScreenSizeWidth	Largura da tela do dispositivo

Tabela 2.3: Informações de contexto fornecidas pelo CIS

O *Location Inference Service* (LIS) é responsável por inferir a localização aproximada de um dispositivo móvel comparando o padrão corrente de sinais de radiofrequência observados pelo dispositivo (obtidos de pontos de acesso 802.11 dentro do raio de cobertura) com o padrão de sinais medidos em pontos de referência pré-definidos. O serviço permite ao usuário definir regiões simbólicas, ou seja, associar nomes a regiões físicas bem definidas (por exemplo, salas, prédios, corredores), que são de interesse para aplicações sensíveis a localização. Uma aplicação cliente pode consultar quais são as região simbólicas mapeadas no serviço, em que região simbólica se localiza um dado dispositivo ou quais dispositivos se encontram em uma determinada região simbólica. Ela pode ainda receber notificações sobre a mudança de área de um dispositivo específico ou a entrada/saída de um dispositivo em uma dada região simbólica.

Outros serviços complementares incluem o SRM (*Symbolic Region Manager*), que permite estabelecer uma relação entre as regiões atômicas definidas pelo LIS, descrevendo uma hierarquia em que regiões podem estar contidas em outras; o DS (*Discovery Service*), um serviço de descoberta; e o CS (*Configuration Service*), um serviço de configuração que armazena endereços dos serviços básicos da MoCA.

Um serviço de privacidade, chamado CoPS (*Context Privacy Service*), foi implementado para controlar como, quando e a quem devem ser fornecidas

informações de contexto [70]. O CoPS é um serviço opcional que permite ao usuário de aplicações sensíveis ao contexto ou baseadas em localização definir e gerenciar suas políticas de privacidade em relação às suas informações de contexto. Antes de atender a qualquer solicitação de informações de contexto, os serviços MoCA consultam o CoPS para decidir se o acesso àquelas informações deve ser concedido ou negado.

2.5

Resumo

Este capítulo apresentou alguns conceitos abordados nesta tese, e que são utilizados no projeto de serviços de disseminação de conteúdo para clientes móveis, com adaptação sensível ao contexto. Inicialmente, foram apresentados os tipos de adaptação para sistemas para computação móvel, com maior ênfase à adaptação de conteúdo (mídia). Também foram discutidos a definição de contexto e suas possíveis modelagens, e sistemas sensíveis a contexto. Outro importante conceito apresentado foi o modelo de comunicação publish/subscribe. Este tipo de comunicação é adequado para serviços de disseminação de conteúdo. Entretanto, o suporte a clientes móveis nos sistemas publish/subscribe existentes limita-se a questões de mobilidade e desconexão, sendo que nenhum provê meios para adaptação do conteúdo das notificações para os clientes móveis. Por fim, foi apresentada a arquitetura MoCA, que é utilizada para a coleta e inferência de informações de contextos de dispositivos móveis.