



UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



CENTRO DE INFORMÁTICA

APLICAÇÕES MÓVEIS CIENTES DE CONTEXTO

UM GERENCIADOR DE GRUPOS DE USUÁRIOS PARA O MOBILE COLLABORATION ARCHITECTURE

Recife, 05 de outubro de 2006.

Aluno: **André Galamba Rodrigues dos Anjos** (agra@cin.ufpe.br)

Orientador: **Carlos André Guimarães Ferraz** (cagf@cin.ufpe.br)

*"Há três coisas na vida que nunca voltam atrás: a flecha lançada, a palavra pronunciada e a oportunidade perdida."
(Provérbio Chinês)*

RESUMO

Computação ciente de contexto é um paradigma computacional no qual aplicações podem descobrir e se aproveitar de informações contextuais, tal como a localização de um usuário, hora do dia, pessoas ou dispositivos próximos e a atual atividade do usuário. Um dos objetivos de uma aplicação ciente de contexto é obter e utilizar informações do contexto de um dispositivo, para prover serviços apropriados para uma determinada pessoa, num determinado local, em um dado momento. Este trabalho irá apresentar um estudo sobre aplicações móveis cientes de contexto, sendo este estudo seguido da implementação de uma aplicação móvel ciente de contexto utilizando o middleware MoCA (Mobile Collaboration Architecture). Será especificado um serviço de gerenciamento de grupos de usuários para o MoCA, com o objetivo de permitir a criação de aplicações inovadoras utilizando este middleware.

Palavras-chave: Contexto, Ciência de Contexto, Computação Móvel, MoCA, Middleware, Colaboração Móvel.

ABSTRACT

Context aware computing is a computational paradigm in which applications can discover and take advantage of contextual information, such as the location of a user, time of the day, people or devices nearby and the user's current activity. One of the objectives of a context aware application is to obtain and utilize context information from a device, to provide appropriate services to a person, in a specific place, in a specific moment. This work will present a study about mobile context aware applications, the study will be followed by the development of a mobile context aware application using the MoCA (Mobile Collaboration Architecture) middleware. A user group management service will be specified for MoCA, with the goal of allowing the creation of innovative applications using this middleware.

Keywords: Context, Context Awareness, Mobile Computing, MoCA, Middleware, Mobile Collaboration.

Dedico este trabalho à minha família e amigos.

AGRADECIMENTOS

À minha mãe e meus irmãos por todo o apoio.

Aos meus amigos de fora da faculdade, por me agüentar falando sobre esse trabalho e pelos momentos de diversão.

A todos os amigos da faculdade, em especial ao eixo, que me deram forças pra seguir essa caminhada até o fim.

Ao meu orientador, Carlos Ferraz, que conseguiu achar uma brecha na sua agenda para me orientar neste trabalho.

Ao Samsung Innovation Group, pela força e motivação na conclusão deste trabalho.

SUMÁRIO

<u>1</u>	<u>INTRODUÇÃO</u>	12
1.1	OBJETIVOS DA MONOGRAFIA	13
1.2	RELEVÂNCIA	13
1.3	METODOLOGIA DE TRABALHO	14
1.4	ORGANIZAÇÃO DA MONOGRAFIA	15
<u>2</u>	<u>APLICAÇÕES CIENTES DE CONTEXTO</u>	16
2.1	INTRODUÇÃO	16
2.2	DEFINIÇÃO DE CONTEXTO	16
2.3	COMPUTAÇÃO CIENTE DE CONTEXTO	18
2.4	CAPTURANDO E MANTENDO CONTEXTO	19
2.5	TIPOS DE APLICAÇÕES CIENTES DE CONTEXTO	21
	TIPO 1 - PROACTIVE TRIGGERING	21
	TIPO 2 - STREAMLINING INTERACTION	21
	TIPO 3 - MEMORY FOR PAST EVENTS	22
	TIPO 4 - REMINDERS FOR FUTURE CONTEXTS	23
	TIPO 5 - OPTIMIZING PATTERNS OF BEHAVIOUR	23
	TIPO 6 - SHARING EXPERIENCES	24
2.6	PRIVACIDADE	24
2.7	EXEMPLOS DE APLICAÇÕES CIENTES DE CONTEXTO	26
<u>3</u>	<u>MOBILE COLLABORATION ARCHITECTURE</u>	30
3.1	VISÃO GERAL	30
3.2	SERVIÇOS	31
3.3	ARQUITETURA	33
3.4	FUNCIONAMENTO	34
3.5	APLICAÇÕES	36
<u>4</u>	<u>A APLICAÇÃO MYOFFICESPACE</u>	38
4.1	MOTIVAÇÃO	38
4.2	DEFINIÇÃO	38

4.3	INTERAÇÃO COM O MOCA	39
4.4	REQUISITOS DA PROVA DE CONCEITO	40
	REQUISITOS FUNCIONAIS	40
	REQUISITOS NÃO FUNCIONAIS	41
4.5	CASOS DE USO	41
4.6	DIAGRAMA DE CLASSES	43
	CLIENTE	44
	SERVIDOR	45
4.7	FUNCIONAMENTO	45
5	<u>GERENCIADOR DE GRUPOS DE USUÁRIOS</u>	49
5.1	MOTIVAÇÃO	49
5.2	DEFINIÇÃO	49
5.3	SERVIÇOS	50
	SERVIÇOS DO GERENCIADOR DE GRUPOS DE USUÁRIOS	51
	SERVIÇOS DE PERFIL DOS USUÁRIOS	56
5.4	COMUNICAÇÃO COM O MOCA	57
5.5	APLICAÇÕES	58
6	<u>CONCLUSÕES</u>	60
6.1	TRABALHOS FUTUROS	61
6.2	CONSIDERAÇÕES FINAIS	61
7	<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	62

LISTA DE IMAGENS

FIGURA 1. A ARQUITETURA DO MOCA..	34
FIGURA 2. SEQUÊNCIA DE INTERAÇÃO ENTRE UMA APLICAÇÃO E OS SERVIÇOS DO MOCA.	34
FIGURA 3. DIAGRAMA DE CLASSES DO CLIENTE DA APLICAÇÃO.	44
FIGURA 4. DIAGRAMA DE CLASSES DO SERVIDOR DA APLICAÇÃO.	45
FIGURA 5. MYOFFICESPACE: LOGIN.	46
FIGURA 6. MYOFFICESPACE: TELA PRINCIPAL.	46
FIGURA 7. MYOFFICESPACE: FIND USERS.....	48

LISTA DE TABELAS

TABELA 1. ALGUMAS APLICAÇÕES DESENVOLVIDAS UTILIZANDO O MOCA..37

1 Introdução

Vivemos um período de popularização e massificação cada vez maior de dispositivos computacionais móveis. Temos atualmente notebooks, cada vez menores e mais poderosos, PDA's (Personal Digital Assistant), smartphones e aparelhos celulares, que permitem aos seus usuários estarem sempre em contato com informações pessoais, informações corporativas, e o mais importante, estarem sempre conectados. A distinção entre comunicação e computação está cada vez menos evidente, não somente nos aparelhos, como também na variedade de formas pelas quais a computação torna possível a comunicação (e-mail, Chat, voz, vídeo).

Os usuários de tecnologia móvel, que formam uma grande parcela da população mundial graças à popularização dos celulares, querem informação acessível de qualquer lugar, a qualquer momento. Uma característica fundamental desse novo estilo de vida conectado é que as pessoas desejam informações que estejam relacionadas de alguma maneira com o contexto onde elas estejam inseridas.

Esta noção de contexto está bastante difundida hoje em dia, ela se refere à situação física e social na qual um dispositivo computacional está inserido. Computação ciente de contexto é um paradigma computacional no qual aplicações podem descobrir e se aproveitar de informações contextuais, tal como a localização do usuário, hora do dia, pessoas ou dispositivos próximos e a atual atividade do usuário.

Um dos objetivos de uma aplicação ciente de contexto é obter e utilizar informação do contexto de um dispositivo para prover serviços apropriados para uma determinada pessoa, num determinado local, em um dado momento. Por exemplo, um celular pode automaticamente entrar no modo silencioso quando o usuário entra numa sala de reunião, isso se o sistema conseguir identificar a localização do aparelho e a hora de uma reunião agendada. É importante que a informação levada ao usuário possa ser interpretada e seja uma informação útil naquele instante, no contexto em que está inserido o usuário.

Neste cenário de aplicações cientes de contexto, já existem implementações de middlewares que possibilitam a criação de aplicações móveis distribuídas cientes de contexto. Uma dessas implementações é o MoCA (Mobile Collaboration Architecture),

que é um projeto cujo objetivo é o desenvolvimento de um middleware flexível e extensível baseado em serviços, destinado a dar suporte ao desenvolvimento de aplicações distribuídas cientes de contexto para redes *mobile*.

Temos atualmente o aparecimento de diversas aplicações móveis cientes de contexto, sendo este tema considerado um forte candidato a abarcar futuras aplicações móveis “matadoras”. Inovações e aumento da capacidade de processamento nos dispositivos móveis permitem a criação de poderosas aplicações, que podem tirar vantagem da sensibilidade ao contexto.

A ciência de contexto é uma característica interessante para tornar novas aplicações móveis ainda mais atraentes. Neste campo de ciência de contexto, temos desde simples aplicações, como a da sala de reunião descrita anteriormente, até sofisticadas aplicações na área de medicina pervasiva.

1.1 Objetivos da Monografia

Este trabalho possui três objetivos principais. Primeiramente, será realizado um estudo sobre aplicações móveis cientes de contexto e sobre o MoCA, o middleware que será utilizado neste trabalho. Em segundo lugar, será implementada uma aplicação móvel ciente de contexto, utilizando o MoCA. Esta implementação tem como objetivo servir como uma prova do conceito que está sendo estudado. O último objetivo do trabalho, consiste na proposição de um novo serviço para compor a plataforma MoCA, sendo este serviço interessante para possibilitar o desenvolvimento de novas aplicações, com novas funcionalidades, utilizando o MoCA.

1.2 Relevância

A relevância deste trabalho é fundamentada no fato de se estar discutindo um assunto muito pertinente e muito em evidência nessa era da computação móvel que é vivida no momento. Torna-se cada vez mais claro que aplicações móveis devem se adaptar ao contexto do usuário, fornecendo informações e serviços úteis nos momentos e situações corretas.

Dessa maneira, esta é um tema importantíssimo a ser estudado, pois como foi dito anteriormente, um dos fortes diferenciais de aplicações móveis daqui em diante

vai ser a utilização de informações do contexto do usuário. Esse trabalho traz a proposta de um novo serviço para o MoCA, contribuindo para a discussão sobre esse tema, e apontando novas possibilidades na elaboração de aplicações e serviços cientes de contexto.

1.3 Metodologia de Trabalho

A metodologia utilizada no desenvolvimento deste trabalho constitui das seguintes fases:

- 1. Revisão bibliográfica do conceito de computação ciente de contexto** – primeiramente, procurou-se entender os conceitos que formam a base do tema estudado. Foi necessário entender as diversas definições de computação ciente de contexto e os vários fatores que cercam o tema.
- 2. Revisão bibliográfica de aplicações cientes de contexto já existentes** – esta atividade contemplou uma pesquisa sobre algumas aplicações cientes de contexto que já foram desenvolvidas. Essa pesquisa serviu tanto para uma documentação e estudo mais aprofundado do tema, como serviu de inspiração para a definição da aplicação a ser desenvolvida nesse trabalho.
- 3. Estudo da arquitetura e funcionamento do MoCA** – consistiu no estudo teórico (arquitetura, componentes) e prático (funcionamento, aplicações de exemplo) do MoCA.
- 4. Definição e implementação da aplicação** – após os estudos realizados, a aplicação a ser desenvolvida foi definida. Partiu-se então para o desenvolvimento da aplicação, que serviu como uma prova de conceito para o tema estudado.
- 5. Definição e especificação do serviço proposto** – por fim, com base em todo o estudo realizado e na experiência prática adquirida com o desenvolvimento da aplicação, o novo serviço para o MoCA foi definido e posteriormente, especificado.

1.4 Organização da Monografia

A monografia encontra-se organizada em 6 (seis) capítulos:

O Capítulo 2 apresenta ao leitor um panorama sobre aplicações móveis cientes de contexto. Os conceitos fundamentais relacionados ao tema são explicitados, bem como assuntos relacionados ao tema principal, como questões envolvendo privacidade. No final desse capítulo apresentamos algumas aplicações móveis já desenvolvidas para exemplificar os conceitos explicados.

O Capítulo 3 traz informações sobre o MoCA, apresentando uma visão geral do middleware e detalhes sobre os componentes que o formam e a arquitetura do sistema. Também será explicado o funcionamento do MoCA e ao final do capítulo será apresentada uma pequena lista das aplicações já desenvolvidas utilizando o MoCA.

No Capítulo 4 são fornecidas informações sobre a aplicação desenvolvida para este trabalho, a aplicação MyOfficeSpace. Será apresentada inicialmente do que se trata a aplicação, qual seu propósito e como ela foi implementada (quais componentes do MoCA foram utilizados, funcionamento de aplicação, etc.).

No Capítulo 5 é apresentada a especificação do novo serviço proposto para o MoCA, um serviço de Gerenciador de Grupos de Usuários. Primeiramente será apresentada a motivação para o serviço proposto, e em seguida a definição e o objetivo desse serviço. Serão detalhadas as funcionalidades básicas que o serviço deve implementar e como essas funcionalidades irão interagir com os serviços existentes no MoCA. Ao final do capítulo, apresentaremos algumas aplicações para o serviço proposto.

No Capítulo 6 são apresentadas as conclusões deste trabalho, contribuições providas pelo mesmo, bem como as atividades futuras sugeridas.

2 Aplicações Cientes de Contexto

2.1 Introdução

Duas tecnologias permitem que ao mesmo tempo, um usuário se mova com poder computacional e recursos de rede na mão. Essas duas tecnologias são computadores portáteis e comunicação sem fio. Os computadores estão cada vez menores, existem atualmente computadores poderosíssimos que podem ser carregados na mão pelo usuário. Ao mesmo tempo, a largura de banda de redes sem fio aumenta constantemente, permitindo um tráfego de dados cada vez maior na rede. Essas mudanças permitem às pessoas acesso ilimitado a informações em qualquer lugar, a qualquer hora.

À medida que os computadores se tornam mais portáteis, as pessoas sentem cada vez mais a necessidade de acessar informação em qualquer momento, em qualquer lugar, com seus dispositivos móveis. Sistemas distribuídos tradicionais, que assumem um ambiente de execução estático, já não são mais suficientes para suprir esta nova demanda. Ao invés de se tentar adaptar os sistemas atuais, deve ser explorada a infra-estrutura existente para suportar a criação de novas aplicações móveis. Aplicações estas que consigam se adaptar com facilidade às necessidades dos usuários, se aproveitar de informações como a localização do dispositivo, pessoas próximas, hora do dia, fazendo tudo isso sem consumir muito da atenção do usuário [Chen & Kotz, 2000].

2.2 Definição de Contexto

De acordo com o Merriam-Webster, a palavra contexto é definida, numa tradução livre, como as condições inter-relacionadas onde alguma coisa existe ou ocorre. Essa definição não ajuda muito a entender o significado de contexto num ambiente computacional. O uso da palavra contexto tende a ser usado de uma maneira muito vaga, visto que tudo no mundo acontece em um determinado contexto. É uma palavra usada recorrentemente em diferentes áreas da computação, onde existem expressões como “ajuda sensível ao contexto”, “busca contextual”, “percepção psicológica contextual”, etc. Neste documento o foco será dado exclusivamente às aplicações existentes num cenário de mobilidade computacional.

Vários pesquisadores expandiram esta definição de contexto, procurando enumerar as características que envolvem a definição do termo. [Schilit et al., 1994] dividiu contexto em três categorias:

- Contexto computacional, tal como conectividade de rede, custos de comunicação, recursos próximos (como impressoras), estações de trabalho, etc.
- Contexto do usuário, tal como o perfil do usuário, localização, pessoas próximas, etc.
- Contexto físico, como a iluminação, nível de ruído, condições do tráfego, temperatura, etc.

O tempo também é uma importante informação de contexto para diversas aplicações. [Chen & Kotz, 2000] propôs a adição do tempo nessa categorização proposta por Schilit. Dessa maneira, teríamos mais uma categoria:

- Contexto temporal, como hora do dia, semana, mês do ano, etc.

Uma característica importante de aplicações cientes de contexto, é que através do armazenamento de informações colhidas, é possível obter um histórico do contexto do usuário. Para certas aplicações este pode ser um dado importantíssimo.

Outros pesquisadores tentaram definir formalmente o conceito de contexto. [Schmidt et al., 1999] definiu contexto como “conhecimento sobre o estado dos dispositivos de informação do usuário, incluindo proximidades, situação, e num menor grau, localização”. [Dey & Abowd, 1999] define contexto como “qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, local, ou objeto que seja considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação.”

De acordo com [Chen & Kotz, 2000], “contexto é o conjunto de estados e configurações ambientais que determinam o comportamento de uma aplicação,

ou, determinam quando um evento da aplicação ocorre e é interessante para o usuário”.

Essa definição de [Chen & Kotz, 2000] pode ser quebrada em duas partes. A primeira parte da definição pode ser caracterizada como contexto ativo, que influencia diretamente o comportamento de uma aplicação. A segunda parte da definição caracteriza um contexto passivo, onde a informação de contexto é relevante, mas não é crítica para a aplicação. Logicamente, o contexto pode ser considerado ativo ou passivo, dependendo do uso que é feito de suas informações em cada aplicação.

2.3 Computação Ciente de Contexto

De nada adianta termos informações de contexto dentro de uma aplicação móvel e não usá-las com eficiência para uma finalidade bem definida. Este ainda é um desafio para desenvolvedores de aplicações, descobrir quais “*killer applications*” podem surgir a partir da utilização de informações de contexto. [Schilit et al., 1994] define computação ciente de contexto categorizando suas aplicações da seguinte forma:

1. Seleção por proximidade - técnica de interface do usuário através da qual objetos localizados próximos à aplicação são enfatizados de alguma maneira.
2. Re-configuração contextual automática - um processo que envolve a adição de novos componentes, a remoção de componentes existentes, ou a alteração de conexões entre componentes geradas por uma mudança de contexto.
3. Informação contextual e comandos - apresentação de resultados diferentes de acordo com o contexto presente.
4. Ações ativadas por contexto - simplesmente uma regra do tipo IF-THEN usada para especificar como sistemas cientes de contexto devem se adaptar.

[Dey & Abowd, 1999] identifica três funcionalidades básicas que aplicações cientes de contexto podem apresentar:

1. Apresentação de informações e serviços para um usuário.
2. Execução de um serviço.
3. Associação de um contexto a uma informação para futura utilização.

De acordo com a definição de contexto apresentada por [Chen & Kotz, 2000], podem ser definidas duas maneiras de se usar o contexto nas aplicações: automaticamente adaptar comportamentos de acordo com o contexto descoberto (utilizando a idéia de contexto ativo discutida anteriormente), ou, apresentar o contexto ao usuário em um dado momento e/ou armazenar o contexto para que seja consultado pelo usuário mais tarde (utilizando a idéia de contexto passivo). A partir daí, podem ser estabelecidas duas definições de computação ciente de contexto:

- Ciência de contexto ativa: uma aplicação se adapta automaticamente ao contexto descoberto, através da mudança do comportamento da aplicação.
- Ciência de contexto passiva: uma aplicação apresenta o contexto novo ou recém atualizado ao usuário interessado, ou torna o contexto persistente para futura consulta.

Ciência de contexto ativa é provavelmente a forma mais interessante de ciência de contexto das formas apresentadas. Ela permite a criação de aplicações móveis inovadoras que podem eliminar inputs desnecessários do usuário, tornando a aplicação o mais discreta possível.

2.4 Capturando e Mantendo Contexto

Aplicações cientes de contexto requerem uma infra-estrutura básica para manter e manipular contextos. Existem então dois elementos envolvidos, captura do contexto atual e a chamada memória de contexto.

O contexto atual é tipicamente capturado através de uma série de equipamentos e técnicas. Informação de contexto pode ser extraída a partir de: sensores, informações já existentes (diários, listas de tarefas, previsões de tempo, valor de ações), estado do dispositivo computacional do usuário e a interação do usuário com este equipamento. Além disso, informações de contexto podem também ser capturadas a partir da entrada explícita de dados pelo usuário. Como um exemplo,

um usuário pode indicar qual seu interesse/necessidade/estado, como por exemplo, “Eu preciso de um restaurante”, “Estou de férias”, “Quero saber sobre prédios históricos”. O contexto atual desse usuário pode então ser representado pela junção desse tipo de informação com informações físicas do usuário (como sua localização), e até mesmo informações virtuais (no caso do usuário desejar intencionalmente fingir alguma situação, por exemplo) [Brown et al., 2000].

Sensores são usados para capturar informações de contexto de baixo nível, como por exemplo, sensores em um escritório podem detectar se uma porta está aberta ou se uma determinada cadeira está sendo usada. Geralmente, seres humanos se interessam mais em elementos contextuais de alto nível, como saber se uma pessoa está ocupada, ou se uma reunião já começou. Existem pesquisas relacionadas à síntese de informações de contexto de alto nível a partir de informações de baixo nível. Com aplicações que conseguem “entender” o contexto de alto nível do usuário, será cada vez menor a quantidade de informação que o usuário terá que prover para que o sistema entenda seu contexto. Por exemplo, a síntese de contexto pode deduzir a partir de uma série de informações de baixo nível que o usuário está de férias.

Síntese também é importante quando é necessário fazer um julgamento baseado em leituras de sensores. Isso pode acontecer quando existem várias alternativas de sensores para uma mesma finalidade, por exemplo, a localização de um usuário pode ser feita através de um GPS (Global Positioning System), ou através do celular do usuário ou ainda através de um reconhecimento de imagens de câmeras. Qualquer um desses dispositivos pode gerar leituras imprecisas, ou pode estar fora do ar, dessa maneira a aplicação deve ser provida com a localização mais provável baseada na informação entregue por esses três dispositivos [Brown et al., 2000].

Outra característica fundamental, além da captura e representação atual do contexto, é a chamada memória de contexto (isso também poderia ser chamado de histórico ou log de contexto, mas o termo memória implica um nível de organização e inteligência adequado para o que será descrito). No nível mais baixo a memória é importante em aplicações onde a informação de contexto por si só não é importante, o que importa é a mudança nessas informações, por exemplo, uma pessoa se movendo. Para detectar a mudança a aplicação deve memorizar

informações passadas. Num nível mais alto, memória de contexto anda em conjunto com a síntese: a análise de contextos passados e, particularmente, a análise das mudanças que estão acontecendo podem levar a intuições valiosas. Por exemplo: “o usuário passou os últimos dez minutos visualizando páginas sobre prédios antigos, logo...”, ou então, “o usuário está escrevendo mensagens para companheiros de trabalho, logo...” [Brown et al., 2000].

2.5 Tipos de Aplicações Cientes de Contexto

No cenário de aplicações cientes de contexto, de acordo com [Brown et al., 2000], existem seis tipos de aplicações interessantes que podem ser exploradas. Será discutido brevemente sobre cada um desses tipos de aplicação, a fim de apontar as possibilidades do uso da ciência de contexto.

Tipo 1 - Proactive Triggering

As aplicações *Proactive Triggering* são aquelas que guardam o atual contexto do usuário e disparam informações para serem apresentadas, ou ações a serem tomadas, quando isso parecer relevante. Um exemplo desse tipo de aplicação é a aplicação clássica do guia turístico. O autor da aplicação pode ter definido que quando o usuário estiver perto de um parque na primavera deverão ser disparadas informações sobre esse parque para o usuário. A aplicação pode também detectar contextos passados relacionados ao contexto presente e apresentar uma informação que foi automaticamente capturada naquele momento. Por exemplo: “Seu avô esteve aqui em 1999, e ele viu esta vista...”. Outro exemplo desse tipo de aplicação seria: “Quando você usou a copiadora pela última vez você apertou tais botões”, em seguida, a aplicação poderia mostrar o passo a passo do uso da copiadora no dispositivo móvel do usuário.

Tipo 2 - Streamlining Interaction

Segundo [Dey & Abowd, 1999], quando humanos conversam com humanos, eles têm a capacidade de usar informações situacionais implícitas, ou contexto, para melhorar a conversação. Infelizmente, essa habilidade de capturar idéias não se transfere para a interação entre homens e computadores.

As aplicações do tipo *Streamlining Interaction* são um híbrido entre comunicação humano-humano e interação humano-computador. Quando as pessoas usam essa

forma de comunicação híbrida, elas desejam conversar e trocar documentos. Para dar suporte a isso deve existir algum tipo de comunicação entre os dispositivos. O maior desafio é criar interfaces amigáveis que permitam a troca de documentos e outros materiais.

Um exemplo desse tipo de aplicação é o projeto Satchel [Lamming et al., 2000], que simplifica as interfaces através do uso de informações de contexto, para que seja possível identificar o que o usuário deve estar fazendo, que dispositivos estão próximos dele, etc.

Tipo 3 - Memory For Past Events

As aplicações desse tipo automaticamente capturam informações de contexto do usuário e armazenam essas informações para uma utilização futura. Posteriormente o usuário pode consultar informações fazendo buscas relacionadas a contextos específicos, como por exemplo: “Encontre as notas que eu fiz durante uma reunião, um ano atrás na sala de conferência. Eu me lembro que João e Maria estavam na reunião e estava chovendo na hora”.

Esse tipo de aplicação, que ajuda a resolver problemas relacionados à memória, tem um grande potencial, principalmente se imaginarmos aplicações desenvolvidas especificamente para pessoas com deficiência de memória.

Um fato interessante sobre esse tipo de aplicações é que ela pode funcionar como um sistema de arquivos, onde os usuários usam a aplicação como fonte e base de dados. No nosso exemplo anterior, o usuário não precisaria mais se preocupar em salvar as notas de uma reunião em um arquivo específico, ao invés disso, o contexto onde o arquivo foi salvo poderia ser usado como chave para sua recuperação.

Pesquisas anteriores desenvolveram um aparelho chamado de Factoid [Mayo, 2000], que pode ser usado para continuamente capturar fatos sobre a vida de um usuário. Os dados coletados podem ser usados como uma memória auxiliar do usuário.

Uma outra aplicação desse tipo é lidar com interrupções que sofremos no dia-a-dia profissional. Após alguma interrupção, o usuário poderia resgatar o contexto em que estava inserido no momento da interrupção, para que possa assim retornar facilmente à atividade que estava sendo desenvolvida.

Tipo 4 - Reminders For Future Contexts

Esse tipo envolve aplicações onde o usuário configura alguma informação com um contexto futuro atrelado a ela, para que, quando esse contexto futuro acontecer a informação seja ativada. Por exemplo: “próxima vez que eu encontrar João, dizer a ele que...”, “quando eu estiver junto de um supermercado, comprar...”, “haverá uma reunião do grupo de projeto esta quinta às 4 da tarde na sala 219”. Este último exemplo é um pouco diferente dos outros, ele indica que um usuário, se fizer parte do grupo de projetos, deve estar num determinado contexto futuro. Ele impõe uma restrição na definição de contexto do usuário, para que a informação atrelada seja disparada.

Sem um contexto necessário, sem a restrição de que uma informação deve ser disparada num dado contexto definido, esse tipo de aplicações é basicamente igual ao tipo 1, *proactive triggering*. Quando existe uma restrição de contexto necessário, a aplicação deve se comportar de maneira diferente, e por causa dessa diferença temos um novo tipo de aplicação. Um outro exemplo desse tipo seria uma aplicação que lembrasse ao usuário quando um determinado compromisso esteja se aproximando. Para isso, a aplicação pode utilizar informações de contexto para conseguir saber a hora correta de indicar o compromisso para o usuário, por exemplo: se o compromisso é às 16:00 e o usuário está num prédio diferente (a 10 minutos de caminhada do local do compromisso), o aviso deve ser dado às 15:45.

Tipo 5 - Optimizing Patterns of Behaviour

Existem aplicações onde a captura do contexto é a principal razão da aplicação. Por exemplo, podemos ter aplicações que gravam o comportamento de animais numa reserva ecológica, ou que auxiliam arqueologistas a documentar os resultados de uma escavação [Pascoe et al., 2000]. Esse tipo de aplicação torna a captura de dados mais fácil, mais confiável e menos dispendiosa.

As aplicações desse tipo podem ir mais além através da análise dos padrões de comportamento dos usuários. Isso é feito através da análise da informação de contexto capturada e da utilização dessa informação para realizar inferências sobre o comportamento do usuário. Por exemplo: “Maria está trabalhando na mesma tarefa que você, você pode falar com ela no número 3327”, “você normalmente utiliza os ônibus X e Y para ir ao trabalho, você sabia que o ônibus Z faz o mesmo trajeto em uma só viagem?”, “agora é um bom momento para o almoço, os restaurantes ainda não estão cheios”, etc.

Tipo 6 - Sharing Experiences

Existe uma tendência atual de desenvolver aplicações colaborativas, onde remotamente usuários trocam experiências. Essas aplicações se baseiam no compartilhamento de um contexto comum: o contexto de um usuário (ou um grupo) é capturado e outros usuários recebem uma simulação desse contexto.

O elemento básico necessário para implementar experiências compartilhadas é a troca de informações de contexto entre usuários. Logicamente, este tipo de aplicação necessita de uma maneira padrão de representar as informações de contexto, e também há a necessidade de um protocolo de descoberta de contexto.

Um exemplo desse tipo de aplicação seria o compartilhamento das informações de contexto de um funcionário de uma fábrica, trabalhando numa pesquisa de campo sobre algum tema. Outros funcionários da fábrica poderiam acompanhar os detalhes da pesquisa através de um consulta sobre as informações de contexto compartilhadas pelo pesquisador.

2.6 Privacidade

Na discussão sobre aplicações móveis cientes de contexto, sempre são levantadas questões relacionadas à segurança e privacidade. Do ponto de vista da aplicação, quanto mais informações são disponibilizadas para ela, melhor ela pode operar, e ao mesmo tempo, mais invasiva ela é. De acordo com [Brown et al., 2000], existem alguns princípios que devem ser seguidos para que uma aplicação ciente de contexto seja aceitável:

- Cada usuário deve ter controle sobre suas informações, e deve ter liberdade, para a qualquer momento, mudar a acessibilidade sobre essas informações, ou até mesmo, modificar a própria informação. Neste último caso, o usuário poderá mentir, mas esse controle é necessário para que o usuário se sinta no controle da aplicação e não se sinta invadido por ela.
- As maneiras pelas quais as aplicações usam a informação devem ser claras para o usuário. Particularmente, o usuário deve entender e controlar como a informação sobre seu contexto é distribuída para outras pessoas. Este tópico é um desafio, já que a maioria das aplicações reais usa informações de maneira muito discreta.
- O usuário pode somente obter informações que sejam do conhecimento dele em um dado momento. Ou seja, um usuário deve poder obter informações sobre quem estava presente numa reunião da qual ele participou, mas o usuário não deve ser capaz de descobrir quem chegou depois que ele deixou a sala, por exemplo.

De acordo com [Chen & Kotz, 2000], existem dois problemas principais na segurança de sistemas cientes de contexto: garantir a acurácia das informações de localização e identidades, e estabelecer comunicações secretas. Não é surpreendente o fato de que a maioria das pessoas não gosta da idéia de ser localizado precisamente, a qualquer momento, por qualquer um, principalmente quando essa informação está sendo gravada.

Fazer uma garantia de privacidade perfeita é um serviço geralmente muito difícil (e caro) para se prover [Spreitzer & Theimer, 1993]. Segundo [Chen & Kotz, 2000], o usuário deve ter controle sobre sua informação contextual, e deve ter controle sobre quem acessa essas informações. A arquitetura do sistema deve prover a possibilidade de balanceamento entre garantia de privacidade e funcionalidade e eficiência. Mas a verdade é que é difícil ser específico sobre que informações devem ser visíveis para quem e quando.

De fato, o problema é que a tecnologia por si mesmo raramente tem um fim maldoso, no entanto ela pode ser usada para bons ou maus propósitos [Want et al., 1992]. Muitos sistemas existentes podem ser usados para comprometer a privacidade das pessoas, graças principalmente a políticas de gerenciamento ruins.

Quando tecnologias são vulneráveis, leis podem ser criadas para garantir que as informações de contexto de um usuário não serão usadas para fins ilícitos.

2.7 Exemplos de Aplicações Cientes de Contexto

Nesta seção será apresentada uma breve pesquisa sobre aplicações cientes de contexto. Esta pesquisa não tem como objetivo ser completa, ela apresenta algumas aplicações interessantes que já foram implementadas, com a intenção de analisar que informações de contexto são mais utilizadas e como elas são utilizadas.

Call Forwarding [Want et al., 1992]

Grupo: Olivetti Research Ltd (ORL).

Informações de contexto: Localização do usuário.

Descrição: A localização dos usuários é mostrada para a recepcionista, que pode então redirecionar um telefonema para o telefone mais próximo de quem deve receber a chamada. Posteriormente, foi desenvolvido um sistema que automaticamente redireciona as chamadas. Foi observado, no entanto, que as pessoas gostariam de ter mais controle sobre a aplicação, para que os telefonemas fossem redirecionados para eles somente em determinados contextos. Por exemplo, as pessoas preferiam não receber ligações redirecionadas durante uma reunião com seus chefes. Esse foi um dos trabalhos pioneiros utilizando conceitos de ciência de contexto.

Teleporting [Benett et al., 1994]

Grupo: Olivetti Research Ltd (ORL).

Informações de contexto: Localização do usuário e a localização de *workstations*.

Descrição: Esta ferramenta tinha como objetivo mapear dinamicamente a interface gráfica de uma aplicação em computadores próximos do usuário. Esta aplicação usa a idéia do “Follow Me”, mapeando a localização das pessoas para que a aplicação “siga” o usuário enquanto ele se movimenta no trabalho.

Active Map [Weiser, 1993] [Want et al., 1995] [Want et al., 1996]

Grupo: Xerox Parc.

Informações de contexto: Localização do usuário.

Descrição: A localização dos usuários é coletada e uma foto dos usuários é mostrada num mapa, que é atualizado em curtos períodos de tempo. A aplicação permite que pessoas se encontrem facilmente, ou percebam o acontecimento de uma reunião da qual desejam participar.

Shopping Assistant [Asthana et al., 1994]

Grupo: AT&T Bell Laboratories.

Informações de contexto: Localização do usuário dentro da loja.

Descrição: O dispositivo guia usuários pela loja, provê detalhes sobre itens, ajuda a localizar itens, apresenta promoções, faz uma comparação de preços, etc. Nesse tipo de aplicação há uma preocupação sobre a privacidade dos usuários. Nesta aplicação, os usuários se dividem em duas categorias: os que usam o serviço anonimamente, sem fornecer informações de perfil, e os que são clientes da loja, fornecem informações pessoais ao sistema e recebem descontos em troca dessas informações.

Cyberguide [Long et al., 1996] [Abowd et al., 1997]

Grupo: Future Computing Environments (FCE) - Georgia Institute of Technology.

Informações de contexto: Localização do turista.

Descrição: Esta talvez seja o exemplo mais clássico de aplicação ciente de contexto. O sistema oferece informações para o turista sobre sua localização atual. O usuário pode acessar informações de direções, obter informações históricas sobre o local e deixar comentários num mapa interativo. O diário do turista é automaticamente compilado usando o histórico dos locais por onde ele passou. Esse histórico também pode ser utilizado pelo sistema para propor sugestões de locais turísticos interessantes que o turista ainda não visitou.

Conference Assistant [Dey et al., 1999]

Grupo: Future Computing Environments (FCE) - Georgia Institute of Technology.

Informações de contexto: Apresentação atual, localização do usuário, hora atual e agenda das apresentações.

Descrição: Sistema para ser usado em conferências. O sistema examina a agenda das apresentações, tópicos das apresentações, localização do usuário e interesses do usuário para sugerir a presença em algumas palestras. Ao entrar numa sala de

apresentação, o sistema disponibiliza o nome do apresentador, o título da apresentação e outras informações relacionadas.

Fieldwork [Pascoe, 1998] [Pascoe et al., 1998]

Grupo: University of Kent at Canterbury.

Informações de contexto: Localização do usuário e hora atual.

Descrição: O sistema provê uma série de ferramentas para auxiliar na observação de campo e atividades de coleta de dados, ajudando o usuário a guardar informações sobre o ambiente onde ele se encontra. O sistema facilita a coleta de dados através da captura automática de informações de contexto do usuário.

Adaptive GSM phone and PDA [Schmidt et al., 1999]

Grupo: TEA1 (Technology for Enabling Awareness) - Starlab.

Informações de contexto: Atividade do usuário, nível de luz, pressão e proximidade com outras pessoas.

Descrição: A aplicação se comporta de maneiras diferentes quando usada em PDA's ou celulares. No cenário do PDA, uma aplicação de bloco de notas é modificada para adaptar o tamanho da fonte à atividade do usuário (fonte grande quando o usuário está caminhando, fonte pequena quando o usuário está parado). Informações do ambiente, como nível de luz, também são analisadas para adaptar a fonte. No cenário de celulares, os perfis do celular são selecionados automaticamente baseados no contexto do usuário. O celular decide entre tocar ou vibrar, ajusta o volume do toque ou entrar no modo silencioso, dependendo se o telefone está na mão do usuário, numa mesa, numa bolsa ou em um ambiente externo.

Location-aware Information Delivery [Marmasse and Schmandt, 2000]

Grupo: MIT Media Laboratory.

Informações de contexto: Localização do usuário e hora atual.

Descrição: O sistema permite a criação de lembretes associados a um local. Quando o destinatário do lembrete entra neste local, a mensagem é entregue a ele através de síntese de voz, evitando que o usuário tenha que segurar o aparelho e ler a mensagem na tela.

Location-aware Applications for the Elderly [Helal et al., 2003]

Grupo: The Pervasive Computing Laboratory (University of Florida).

Informações de contexto: Localização do usuário e hora atual.

Descrição: Este projeto teve como objetivo construir lares inteligentes para auxiliar o dia-a-dia de idosos. Através de um sistema de localização dentro da casa do idoso, o sistema consegue realizar sugestões pró-ativas para servir ao usuário, utilizando suas informações de contexto.

AccesSights [Klante et al., 2004]

Grupo: Oldenburg Research and Development Institute for Information Technology Tools and Systems.

Informações de contexto: Localização do usuário.

Descrição: Este é outro sistema de guia turístico, com o diferencial que ele foca em prover informações para grupos de usuários portadores de deficiência visual. O sistema definiu uma interface multimodal para suportar tanto usuários deficientes visuais, como usuários não deficientes, cada um de uma maneira adequada.

Applications of context-aware computing in hospital work [Bardram, 2004]

Grupo: University of Aarhus.

Informações de contexto: Localização do aparelho, informações sobre o paciente, hora do dia.

Descrição: Esse sistema aplica o conceito de ciência de contexto para trabalhos médicos em hospitais. Foram implementados um recipiente para remédios e uma cama de hospital cientes de contexto, que reagem e se adaptam ao que está acontecendo ao seu redor.

3 Mobile Collaboration Architecture

3.1 Visão Geral

O MoCA (Mobile Collaboration Architecture) é um middleware que suporta o desenvolvimento e implantação de aplicações distribuídas cientes de contexto para redes locais sem fio, isso é, IEEE 802.11b/g WLANs. O MoCA provê meios para a coleta, armazenamento e processamento de informações de contexto de dispositivos móveis (notebooks, por exemplo) [MoCA website, 2006].

Essencialmente, o MoCA consiste num conjunto de API's que permitem o desenvolvimento de aplicações de maneira mais simples e efetiva, provendo fácil acesso a serviços genéricos úteis para uma grande variedade de aplicações. O MoCA não assume nenhuma arquitetura específica para a aplicação (por exemplo, cliente/servidor ou *Peer-to-Peer*). No entanto, requer que os dispositivos móveis tenham interfaces IEEE 802.11, e sejam capaz de se conectar a *access points* IEEE 802.11 [MoCA website, 2006].

O MoCA também envolve uma série de serviços que suportam a execução de aplicações cientes de contexto. Até o momento, o MoCA pode ser usado em dispositivos utilizando Windows XP, Windows CE ou PalmOS.

A infra-estrutura do MoCA consiste de API's cliente e servidor, serviços básicos para suportar as aplicações colaborativas e um *framework* para a implementação de *proxies* de aplicação (ProxyFramework), que pode ser customizado para atender às necessidades específicas da aplicação colaborativa e que facilita o acesso aos serviços básicos pela aplicação. As API's e os serviços básicos foram concebidos para serem genéricos e flexíveis, úteis para um grande número de aplicações. Por exemplo, deve ser suportado interação síncrona ou assíncrona, baseada em mensagens ou no compartilhamento de artefatos, etc. [Rodrigues et al., 2004].

Em MoCA, cada aplicação tem três partes: um servidor, um *proxy* e um cliente. Os dois primeiros executam em nós da rede, enquanto o cliente roda num dispositivo móvel. Um *proxy* intermedeia toda a comunicação entre a aplicação do servidor e um ou mais dos seus clientes em dispositivos móveis.

Aplicações com requisitos de envolver um grande número de clientes podem ter diversos *proxies* executando em diferentes redes. O *proxy* de uma aplicação pode ser encarregado de várias tarefas, tal como a adaptação de dados transferidos, por exemplo: compressão, conversão de protocolos, encriptação, autenticação de usuários, processamento de contexto, registro e localização de serviços, gerenciamento do *handover* e outros. A maior parte dessas tarefas requerem uma grande quantidade de processamento, e dessa forma, o *proxy* também serve como um meio de distribuir o processamento de uma aplicação entre o servidor e seus *proxies* [Rodrigues et al., 2004].

3.2 Serviços

O MoCA oferece uma série de serviços que suportam o desenvolvimento de aplicações colaborativas cientes de contexto. De acordo com informações contidas em [Rodrigues et al., 2004] e em [MoCA website, 2006] serão definidos os papéis dos serviços existentes no MoCA.

Client and Server API's: o servidor e o cliente da aplicação colaborativa devem ser desenvolvidos usando as API's do MoCA, visto que elas escondem do desenvolvedor a maioria dos detalhes envolvendo o uso de serviços providos pela arquitetura.

Monitor: é um *daemon* executando em cada dispositivo móvel. Ele fica encarregado de colher os dados relacionados à execução, estado e ambiente do aparelho, e enviar esses dados ao CIS (Context Information Service) executando em um (ou mais) nó(s) da rede. Os dados coletados incluem a qualidade da conexão sem fio do dispositivo móvel, bateria restante, uso da CPU, memória livre, *Access Point* (AP) atual (que está sendo acessado no momento), lista de todos os AP's e a força dos seus sinais que estão ao alcance do dispositivo móvel.

Configuration Service (CS): este serviço se encarrega de armazenar e gerenciar informações de configuração de todos os dispositivos móveis, para que estes possam utilizar os serviços do MoCA, tal como CIS e o Discovery Service (DS). A informação de configuração é armazenada numa *hash table* persistente, onde cada entrada (indexada pelo endereço MAC de cada dispositivo) guarda as

seguintes informações: os endereços (IP:porta) de um servidor CIS e de um Discovery Server, e a periodicidade que o Monitor deve mandar as informações do dispositivo móvel para o CIS. A indexação de acordo com o endereço MAC é essencial para a implementação de um CIS distribuído, onde cada servidor recebe aproximadamente a mesma carga de processamento de contexto.

Discovery Service (DS): é encarregado de armazenar informações, tal como nome, propriedades, endereços, de qualquer aplicação (seus servidores e *proxies*) ou de qualquer serviço registrado no MoCA. O DS provê uma maneira de realizar descoberta de serviços com base na localização do dispositivo.

Context Information Service (CIS): este é um serviço distribuído, onde cada servidor CIS recebe e processa as informações de estado dos dispositivos enviadas pelos Monitores. O contexto atual do dispositivo é constantemente monitorado pelo Monitor, e é disponibilizado para qualquer aplicação que esteja registrada no CIS para receber notificações sobre mudanças de contexto. As informações de contexto do usuário incluem força do sinal RF (Rádio Frequência), nível de energia, uso da CPU, memória livre, entre outras informações.

Location Inference Service (LIS): infere a posição lógica aproximada de um dispositivo. Isso é realizado através da comparação dos padrões de sinais RF recebidos (de todos os *access points* 802.11 audíveis), com os padrões de sinais previamente medidos em pontos de referência (*reference points*) pré-definidos em um prédio ou campus. Dessa maneira, antes de poder fazer qualquer inferência, o banco de dados do LIS deve ser populado com medições RF em cada *reference point*, e parâmetros de inferência devem ser escolhidos de acordo com a característica da região. O número de *reference points* determina a confiabilidade da inferência. O LIS usa a informação de contexto de qualquer dispositivo coletada pelo CIS, e calcula a mediana da distância euclidiana para cada *reference point*. O LIS usa um modelo hierárquico onde a posição de um dispositivo pode ser dada através de coordenadas ou por um nome simbólico.

Symbolic Region Manager (SRM): Este serviço expõe uma interface para definir (criar, modificar e deletar) e resgatar informações sobre hierarquias de regiões simbólicas. Novas regiões devem se basear no conjunto de regiões

atômicas definidos pelo administrador do LIS. Regiões simbólicas é o nome que se dá para áreas físicas bem definidas, como salas, corredores, prédios, que podem ser do interesse de aplicações baseadas na localização do usuário.

Context Privacy Service (CoPS): Controla quando, como, e para quem informações de contexto coletadas e processadas pelo CIS e LIS serão reveladas. Este serviço implementa um mecanismo que avalia políticas de privacidade e checka se o acesso à informação de contexto de um usuário será permitido a um outro usuário ou aplicação.

Proxy Framework: O Proxy Framework tem como principal objetivo facilitar o desenvolvimento de aplicações *proxy*. O *framework* é responsável por interagir com o MoCA, escondendo os detalhes desta interação e se cadastrando como interessado em receber notificações nos contextos do interesse da aplicação.

Além desses serviços, o MoCA também disponibiliza uma série de API's para auxiliar no desenvolvimento de aplicações. Estas API's serão descritas a seguir.

MoCA Util: Seu objetivo é facilitar o uso e a implementação de tarefas comuns necessárias em vários projetos. Alguns dos serviços genéricos implementados nessa API são: Listener Manager, Thread Pools, etc.

Event-Based Communication Interface: Implementa a funcionalidade de um serviço baseado em eventos. Isso permite que clientes se registrem em eventos específicos e sejam notificados quando esses eventos ocorrerem.

Communication Protocol: Implementa protocolos para comunicação síncrona e assíncrona de objetos pela rede. Um cliente pode enviar mensagens usando TCP ou UDP.

3.3 Arquitetura

Abaixo temos a arquitetura do MoCA, que mostra como os componentes básicos descritos se comunicam.

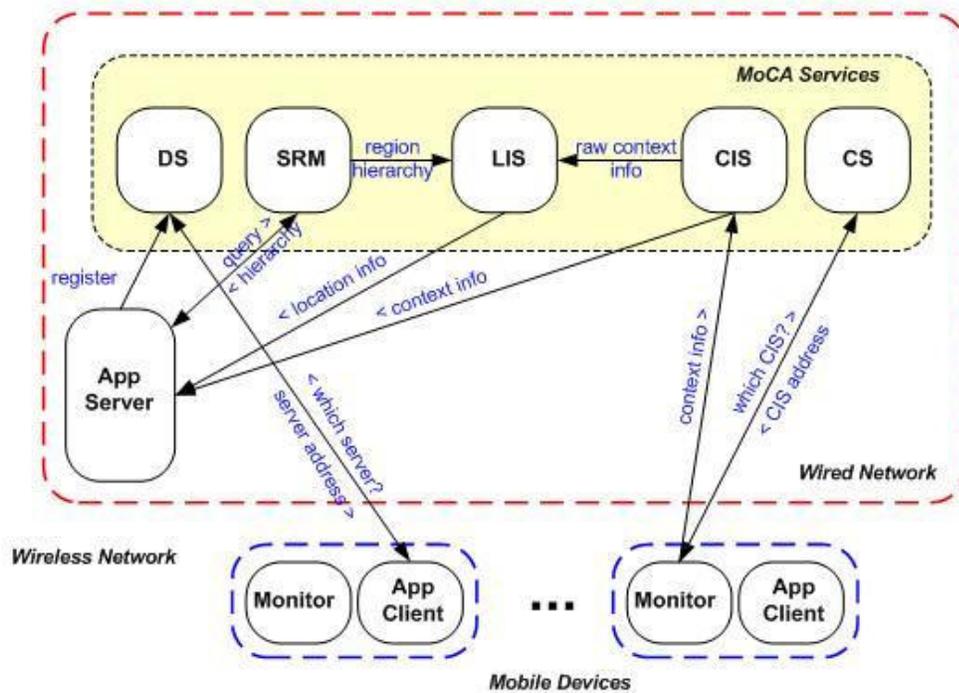


Figura 1. A arquitetura do MoCA..

3.4 Funcionamento

[Rodrigues et al., 2004] explicam como uma aplicação cliente se comunica e interage com a infra-estrutura existente no MoCA.

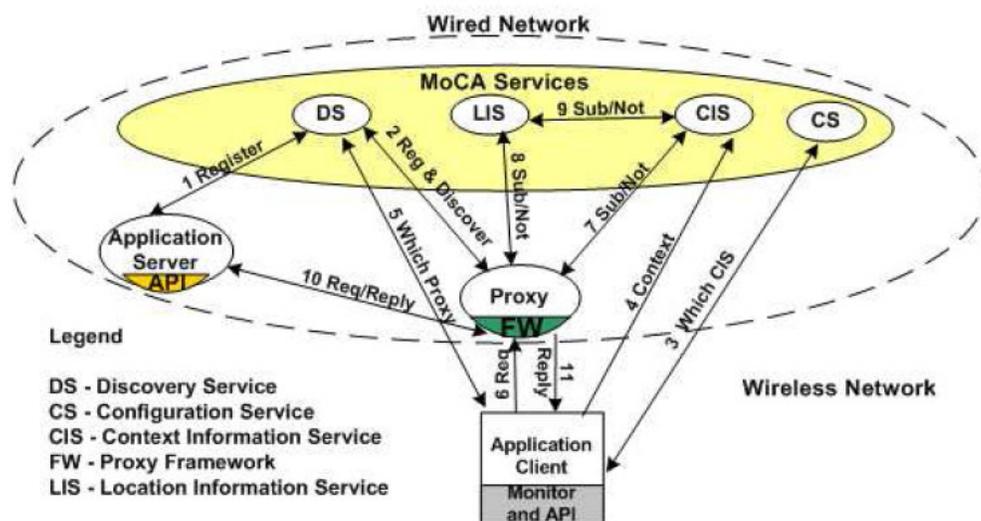


Figura 2. Sequência de interação entre uma aplicação e os serviços do MoCA.

A figura acima mostra uma seqüência típica de interação entre os elementos da arquitetura do MoCA e uma aplicação. O objetivo desta seção é ilustrar o papel de cada um desses elementos durante o registro e a execução de uma aplicação colaborativa, composta de uma (ou mais) instâncias de um Application Server, Proxy(ies) e Application Clients.

Inicialmente, o Application Server se registra no DS (passo 1) informando o nome e as propriedades do serviço colaborativo que ele implementa. Cada *proxy* da aplicação também realizam um registro no DS (passo 2). Dessa maneira, o Application Client pode perguntar ao DS para descobrir como acessar um dado serviço colaborativo na sua rede atual, através do Application Server ou do Proxy. O Monitor que executa em cada dispositivo móvel, captura o estado dos recursos locais e dos sinais RF e envia essa informação de contexto para o CIS.

Como mencionado anteriormente, o endereço do CIS e a periodicidade para envio das informações de contexto são obtidas do CS quando o Monitor é iniciado (passo 3). Dessa maneira, o Monitor passa a enviar periodicamente as informações de contexto do dispositivo móvel para o CIS (passo 4).

Depois de descobrir um Proxy que implemente os serviços colaborativos desejados a partir do DS (passo 5), o cliente pode começar a enviar *requests* para o Application Server. Cada um desses *requests* é roteado pelo Proxy correspondente (passo 6), que processa o *request* do cliente com relação à necessidade de adaptações específicas da aplicação, e repassa o *request* para o Application Server. Por exemplo, o Proxy pode enviar uma expressão de interesse para o CIS (passo 7) registrando seu interesse em notificações de eventos sobre a mudança de estado de um cliente. A expressão de interesse pode ser por exemplo: “FreeMem < 15%” OR “roaming=True”. A partir desse ponto, quando o CIS receber a informação de contexto de um dispositivo (do Monitor correspondente), ele checa se o novo contexto torna alguma das expressões de interesse verdadeiras. Se esse for o caso, o CIS cria uma mensagem de notificação e envia para todos os Proxies que estão registrados e tenham interesse em tal mudança de estado do dispositivo.

Aplicações que requerem informação de localização registram o seu interesse junto ao LIS (passo 8), que em seguida se registra junto ao CIS (passo 9) para receber atualizações periódicas dos sinais RF do dispositivo. O LIS usa as informações sobre os sinais RF para inferir a localização do dispositivo e enviar as notificações correspondentes ao Proxy da aplicação.

Quando o Application Server recebe um *request* de um cliente, o *request* é processado e uma resposta é enviada para alguns ou todos os Proxies (passo 10), que podem então modificar/processar a resposta de acordo com a notificação recebida do CIS sobre o dispositivo móvel correspondente. O processamento específico de contexto depende dos requisitos específicos de cada aplicação desenvolvida. Por exemplo, se o Proxy é informado que a qualidade da conectividade sem-fio de um dispositivo móvel caiu abaixo de um nível mínimo pré-estabelecido, ele pode temporariamente armazenar os dados das respostas do servidor em um buffer local, ou remover parte dos dados, ou ainda aplicar alguma compressão, etc. O Proxy pode utilizar outras informações de contexto, como a localização do aparelho por exemplo, para determinar que dados, quando, e como serão enviados para o cliente no dispositivo móvel (passo 11).

A arquitetura também implementa transparência de mobilidade para as aplicações. Quando um dispositivo móvel entra em outra rede, o Monitor detecta essa mudança e o CIS notifica o Proxy. O Proxy realiza o *handover* em nível de aplicação, determinando qual o Proxy mais apropriado para o aparelho na nova rede, e se possível, transferindo o estado de sessão para este novo Proxy.

3.5 Aplicações

Uma série de aplicações móveis cientes de contexto já foram desenvolvidas utilizando o MoCA como base. Serão brevemente descritas a seguir algumas aplicações implementadas:

Aplicação	Descrição
comManager	Ferramenta que permite o estabelecimento de comunicação entre usuários numa mesma área física. O usuário pode ter uma lista de amigos e receber informações de status desses amigos, e localização de grupos de amigos (comunidades).
W-CHAT	Programa simples de <i>chat</i> com a peculiaridade de permitir que o usuário possa ler posteriormente mensagens que ele recebeu enquanto estava desconectado.
uGuide: You Guide	Informa ao usuário a sua localização atual e possui uma opção de se acrescentar um <i>website</i> fornecendo informação extra sobre tal localidade.
BuddySpaceLive	É uma extensão para uma aplicação de <i>Instant Messaging</i> (IM). Ela implementa mapas que atualizam dinamicamente a posição dos usuários na lista de amigos.
WMS: Wireless Marketing Service	Permite o envio de cupons de desconto para o dispositivo móvel de clientes de acordo com a localização do usuário.
Virtual Lines	Sistema de controle para filas em parques temáticos. Para entrar na fila o usuário deve passar perto de um brinquedo. Quando chegar próximo da sua vez, o sistema emitirá um aviso ao usuário.
WAY: Who Are You?	Permite o reconhecimento de outros dispositivos móveis próximos ao usuário, e permite a troca de um cartão de identificação com esses vizinhos.

Tabela 1. Algumas aplicações desenvolvidas utilizando o MoCA..

4 A Aplicação MyOfficeSpace

4.1 Motivação

A motivação para a implementação de uma aplicação é utilizar tudo que foi estudado neste trabalho para o desenvolvimento de uma prova de conceito. Através desta implementação, foi possível utilizar os conceitos e a infra-estrutura do MoCA, e pôr em prática os conhecimentos de aplicações cientes de contexto na hora de definir qual seria a aplicação, o que ela deveria fazer e como ela deveria funcionar.

O objetivo desta implementação não é desenvolver uma aplicação de grande apelo comercial ou de grande caráter inovador. O objetivo é pôr em prática tudo que foi aprendido na pesquisa inicial do trabalho e desenvolver uma prova de conceito funcional utilizando o MoCA.

4.2 Definição

O MyOfficeSpace consiste de uma ferramenta para auxiliar o dia-a-dia de funcionários em uma empresa, através da utilização de informações de contexto dos usuários para prover informações úteis. Várias informações diferentes, como a localização da pessoa, agenda de tarefas e a hora do dia, são cruzadas para fornecer informações para a pessoa na situação apropriada. O MyOfficeSpace se trata de uma aplicação móvel ciente de contexto que irá rodar em dispositivos móveis (notebooks ou celulares) dos funcionários.

A aplicação possui três funcionalidades básicas:

- Encontrar usuários: através dessa função, temos acesso a um mapa da empresa e conseguimos localizar onde cada funcionário se encontra no momento.
- Informativos ligados às dependências da empresa: através dessa função, podemos ter informações relacionadas a cada lugar da empresa. Quando um usuário se move de uma dependência para outra, a informação daquele local é atualizada para o usuário. Por exemplo, ao chegar no pátio da empresa, o usuário pode receber informações sobre treinamentos, cursos, quadro de avisos da empresa, etc. Quando o usuário entra na cantina da

empresa ele pode receber informações sobre telefones de pizzarias, lanchonetes, etc.

- Informações para o usuário baseadas em uma agenda de tarefas da empresa: quando o usuário entra na empresa ele recebe informações sobre a agenda de tarefas de seu grupo de trabalho, indicando quais as próximas tarefas de acordo com a hora atual. Algumas tarefas podem ser configuradas para serem mostradas somente quando o usuário se encontra num local específico. Por exemplo, uma tarefa como “Reunião do Projeto às 16:00”, pode ser configurada para ser mostrada de qualquer ponto da empresa, enquanto uma tarefa como “Pesquisar sobre o MoCA”, pode ser mostrada somente no laboratório.

A aplicação consiste de dois módulos principais. Um cliente (que roda em cada dispositivo móvel dos usuários) e um servidor (que roda numa máquina conectada à rede). O dispositivo móvel deve ter, além do cliente da aplicação, o Monitor do MoCA instalado, e a máquina servidora deve ter os serviços básicos do MoCA rodando.

Atualmente, o MoCA oferece Monitores para Windows XP, Windows CE e PalmOS, além de um simulador de monitor. Nesta implementação foi utilizado o simulador de monitor, para facilitar os testes e por se tratar de uma prova de conceito. Como já foi discutido no capítulo sobre MoCA, é necessário que haja uma rede Wi-Fi em funcionamento no local para que uma aplicação do MoCA possa rodar. Por falta de dispositivos móveis disponíveis (notebooks ou Palms Wi-Fi), a aplicação foi testada utilizando o simulador de monitor. A aplicação foi desenvolvida na linguagem de programação Java.

4.3 Interação com o MoCA

No desenvolvimento da aplicação vários serviços do MoCA foram utilizados. O serviço mais utilizado foi o LIS (Location Inference Service). Como já foi discutido no capítulo sobre o MoCA, o LIS oferece métodos que permitem identificar a localização dos dispositivos dentro da rede. Em todas as funcionalidades do MyOfficeSpace, o LIS foi utilizado para buscar a localização do usuário e a partir daí poder prover informações úteis para ele. Para utilizar os serviços do LIS, é necessário desenhar um mapa da área coberta pela rede

utilizando o Mapper Tool. Após a criação do mapa, é necessário colher padrões de sinais RF para cada região, definindo os *reference points*. Esses mapas consistem de um padrão de sinais RF recebidos de todos os *access points* audíveis, e são utilizados pelo LIS para fazer a inferência da localização do usuário.

Além do LIS, o ECI (Event-Based Communication Interface) foi utilizado. O ECI possibilita a implementação de comunicação assíncrona usando a abordagem de *publish/subscribe*. Utilizando esse serviço, a aplicação cliente pode subscrever tópicos do seu interesse e receber informações sobre eventos que ocorram neste tópico. Este serviço foi utilizado para possibilitar as comunicações entre a aplicação cliente e servidor.

4.4 Requisitos da Prova de Conceito

Para auxiliar o desenvolvimento da aplicação, foram definidos os requisitos que devem ser atendidos por ela. Os requisitos foram especificados genericamente, não se fazendo distinção entre a parte servidora e cliente da aplicação.

Requisitos Funcionais

[RF01] Localizar usuários

A aplicação deve ser capaz de localizar os usuários no espaço geográfico coberto pela rede Wi-Fi e definido no mapa da aplicação.

[RF02] Mudança de região

A aplicação deve ser capaz de identificar quando um usuário sair de uma região para outra.

[RF03] Prover informações sobre uma região

A aplicação deve ser capaz de prover informações sobre as regiões definidas no mapa. Essas informações devem ser acessíveis a todos os usuários.

[RF04] Prover informações baseadas na agenda de tarefas do grupo de trabalho

A aplicação deve ser capaz de interpretar a agenda de tarefas do grupo de trabalho e mostrar a tarefa atual e as existentes nas próximas 3 horas.

Requisitos Não Funcionais

[RNF01] Prova de conceito

A aplicação deve utilizar os conceitos de aplicações móveis cientes de contexto estudados, e deve utilizar o MoCA no desenvolvimento da aplicação.

4.5 Casos de Uso

Com a especificação dos requisitos, foram definidos os casos de uso que deveriam ser implementados para a prova de conceito.

[CU01] Realizar login

Caso de uso responsável pela autenticação do usuário. Esse caso de uso foi implementado na prova de conceito para podermos testar a aplicação utilizando usuários diferentes. Num caso prático, onde a aplicação fosse desenvolvida comercialmente, não teríamos esse caso de uso, a aplicação cliente do usuário logaria no sistema automaticamente.

Entradas e pré-condições: A aplicação e infra-estrutura do MoCA devem estar funcionando adequadamente.

Saídas e pós-condições: Usuário logado no sistema.

Fluxo Principal de Eventos

1. O usuário digita seu login (como se trata de uma prova de conceito, basta o login para o usuário entrar no sistema).
2. O sistema verifica o login do usuário e passa para a tela principal da aplicação. Se o login não existir, o sistema apresenta uma mensagem para o usuário e permite que ele tente novamente.

[CU02] Atualizar informações sobre uma região

Caso de uso responsável por disponibilizar as informações (mensagens) existentes na região onde o usuário se encontra.

Entradas e pré-condições: O usuário deve estar logado no sistema.

Saídas e pós-condições: Nenhuma.

Fluxo Principal de Eventos

1. O sistema monitora as mudanças de localização do usuário.
2. Sempre que o usuário entra numa nova região, o servidor é avisado desse evento através de uma função do LIS. O servidor então envia essa informação para o cliente que está rodando no dispositivo móvel do usuário. Além dessa informação de mudança de local, o servidor envia as mensagens associadas à região onde o usuário acabou de entrar.
3. Na tela principal da aplicação uma caixa de texto disponibiliza as mensagens da região atual para o usuário.

[CU03] Atualizar informações de tarefas

Caso de uso responsável por disponibilizar as tarefas pendentes do dia para o usuário. Essas tarefas podem aparecer somente quando o usuário está no local da tarefa, ou podem aparecer de qualquer ponto da rede, dependendo da configuração de cada tarefa existente.

Entradas e pré-condições: O usuário deve estar logado no sistema.

Saídas e pós-condições: Nenhuma.

Fluxo Principal de Eventos

1. A aplicação cliente (rodando no dispositivo móvel) pergunta de tempos em tempos ao servidor se há novas tarefas para aquele usuário.
2. Quando há novas tarefas, o servidor envia a data, hora e local da tarefa para o cliente.
3. Na tela principal da aplicação cliente uma caixa de texto disponibiliza as tarefas pendentes do usuário.

[CU04] Localizar usuários

Caso de uso responsável por localizar os outros usuários que estão logados no sistema. Esse caso de uso apresenta um mapa, com os logins dos usuários nas regiões onde se encontram.

Entradas e pré-condições: O usuário deve estar logado no sistema.

Saídas e pós-condições: Nenhuma.

Fluxo Principal de Eventos

1. O usuário clica no botão “Find Users”.
2. A aplicação cliente envia a requisição para localizar os usuários ao servidor.
3. O servidor, utilizando uma função do LIS, identifica onde cada usuário do sistema está localizado.
4. A mensagem com a localização dos usuários é enviada de volta para a aplicação cliente.
5. A aplicação cliente mostra um mapa, com o login dos usuários nas regiões do mapa onde esses usuários se encontram.

4.6 Diagrama de Classes

Nesta seção, os diagramas de classes do cliente e do servidor da aplicação são apresentados. A maneira como o sistema funciona e as suas classes interagem será discutida em detalhes na próxima seção.

Cliente

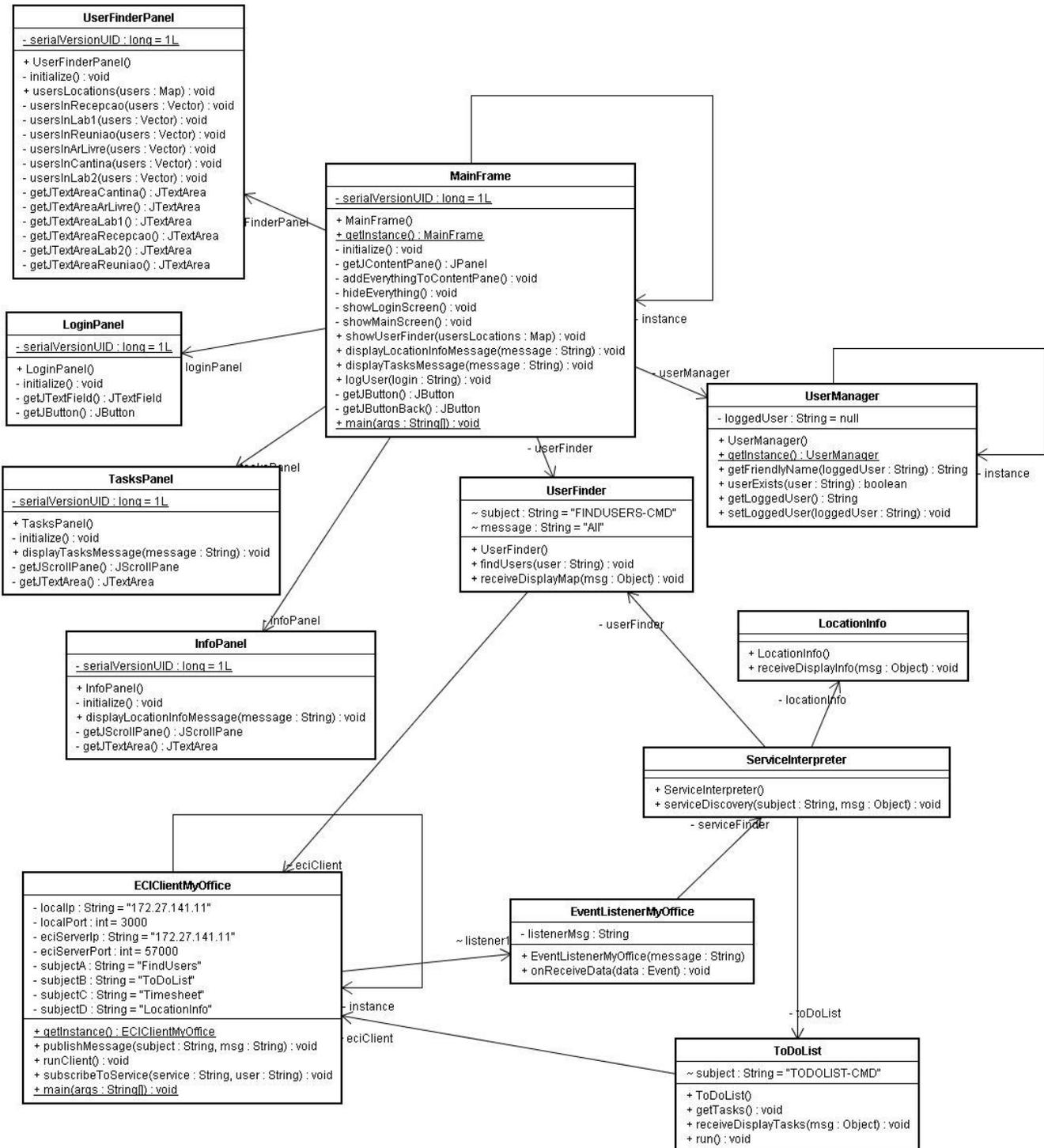


Figura 3. Diagrama de classes do cliente da aplicação.

Servidor

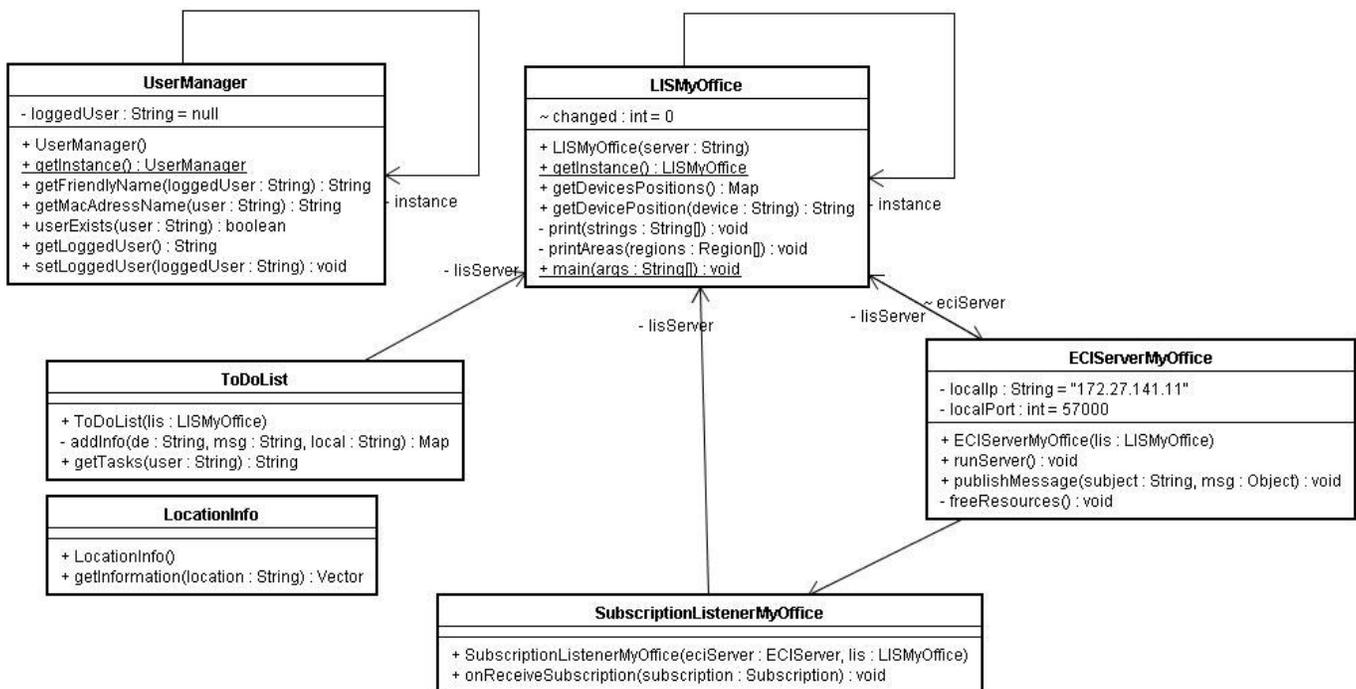


Figura 4. Diagrama de classes do servidor da aplicação.

4.7 Funcionamento

Ao iniciar a aplicação, o usuário deve informar o seu login para ter acesso ao sistema. Como está descrito no Caso de Uso 01, o login foi implementado para podermos testar a aplicação utilizando usuários diferentes. Num caso prático o login seria realizado automaticamente. Ao receber o login, a classe `UserManager` checa se aquele login é válido e então a aplicação é iniciada.



Figura 5. MyOfficeSpace: Login.

Depois de informado o login, o usuário tem acesso à tela principal da aplicação. Nesta tela, temos duas áreas de texto, uma contendo as tarefas presentes na agenda do grupo de trabalho, e outra área contendo as informações da região onde o usuário se encontra.



Figura 6. MyOfficeSpace: Tela Principal.

A área de texto das tarefas pendentes funciona da seguinte maneira: de acordo com o diagrama de classes da aplicação, temos no cliente uma classe chamada

ToDoList. Esta classe cuida da parte das tarefas do usuário. Essa classe (através do ECIClientMyOffice) envia uma mensagem perguntando sobre novas tarefas do usuário de 5 em 5 segundos. O servidor recebe essa mensagem através do SubscriptionListenerMyOffice e redireciona essa chamada para a classe ToDoList do servidor. Essa classe tem acesso à agenda do grupo de trabalho e retorna as tarefas existentes nas próximas 3 horas. Em cada tarefa temos um criador, a data e hora, a mensagem de texto e o local que deve ser atrelado à tarefa. O local pode ser alguma região específica ou pode envolver toda a área da empresa. A mensagem com as tarefas é enviada pelo servidor para o cliente. O EventListenerMyOffice do cliente recebe esta mensagem e a repassa para o ServiceInterpreter. O ServiceInterpreter sabe quem deve cuidar dessa mensagem, e a repassa para a classe ToDoList do cliente. Esta classe trata a mensagem e a envia para a GUI, para ser apresentada na tela para o usuário. Esta comunicação de mensagens está sendo realizada através da API do MoCA chamada ECI (Event-Based Communication Interface). Esta API foi utilizada tanto no cliente quanto no servidor para facilitar a troca de mensagens.

A área de texto das informações das regiões é atualizada a partir de informações enviadas pelo servidor da aplicação. No servidor temos uma classe chamada LISMyOffice, que acessando os métodos do LIS consegue obter informações de localização dos usuários. Dentro do LISMyOffice temos um RegionListener, que dispara um evento sempre que um dispositivo entra numa nova região lógica definida. Ao recebermos esse evento, conseguimos identificar qual dispositivo (através do endereço MAC do dispositivo) entrou em que região (através do ID da região). O servidor busca que informações estão associadas com aquela região usando o LocationInfo, e envia uma mensagem com essas informações ao usuário em questão (que entrou numa nova região) através do ECIServerMyOffice. A aplicação cliente recebe esta mensagem através do EventListenerMyOffice que repassa a mensagem para o ServiceInterpreter. O ServiceInterpreter repassa a mensagem para o LocationInfo, que trata as informações e as envia para a GUI, para serem apresentadas ao usuário.

Temos ainda a funcionalidade de buscar onde estão outros usuários logados no sistema. Quando o usuário clica no botão FindUsers, o sistema apresenta um mapa com o login dos usuários no lugar onde se encontram no momento.

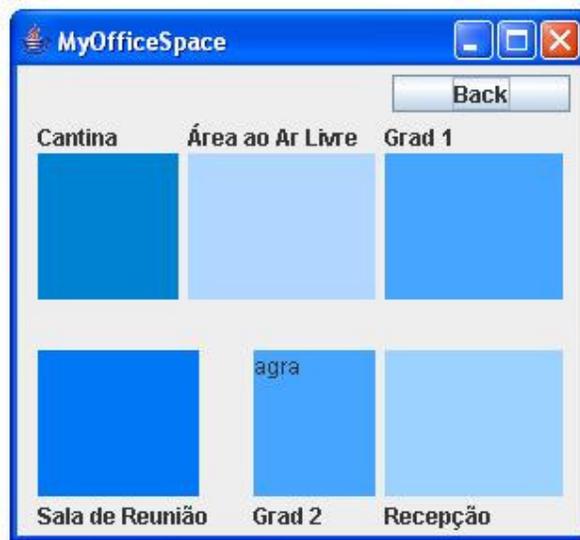


Figura 7. MyOfficeSpace: Find Users.

Quando o botão FindUsers é clicado, a aplicação cliente usa a classe UserFinder que, através do ECIClientMyOffice, envia uma mensagem para o servidor perguntando onde estão os usuários logados no sistema naquele momento. O servidor recebe esta mensagem através do SubscriptionListenerMyOffice e chama um método da classe LISMyOffice. Esse método, através de uma funcionalidade disponibilizada pelo LIS, consegue identificar a localização de todos os dispositivos no sistema. É formatada uma mensagem contendo a localização de todos os usuários, e essa mensagem é enviada de volta pra o cliente da aplicação. Mais uma vez, o cliente recebe a mensagem através do EventListenerMyOffice e repassa essa mensagem para o ServiceInterpreter. Este último repassa a mensagem para o UserFinder, que formata a informação e envia para a GUI. A GUI desenha o mapa do sistema e indica com o login em que região cada usuário se encontra no momento.

5 Gerenciador de Grupos de Usuários

5.1 Motivação

Nesta seção iremos fazer a especificação de um novo serviço para ser incorporado ao MoCA. Como o próprio nome já diz, o MoCA tem como um dos objetivos dar suporte a aplicações móveis colaborativas e atualmente possui uma série de serviços para suportar o desenvolvimento dessas aplicações.

Quando pensamos em colaboração logo vem à mente a idéia de um grupo de pessoas que compartilham algum objetivo. A noção de grupos de pessoas é muito forte em nossas vidas, é muito comum observarmos pessoas atendendo telefones celulares e imediatamente dizendo: “estou na praia com João e Maria”, ou “estou com o pessoal do trabalho”, por exemplo. A informação sobre que pessoas estão próximas a um indivíduo é uma importantíssima informação de contexto, tão importante quanto sua localização, hora, suas atividades atuais, etc. Através da obtenção dessa informação é possível saber com que pessoas um usuário está interagindo fisicamente, e a partir dessa idéia várias aplicações interessantes podem surgir.

No momento o MoCA não possui um componente que trate explicitamente do gerenciamento de grupos de usuários. Através da utilização de métodos do LIS é possível identificar que usuários estão próximos dos outros, no entanto, esse módulo não possui métodos que facilitem a criação abstrata de grupos de usuários e nem oferece métodos que facilitem a colaboração entre essas pessoas. A partir dessa necessidade, surgiu a proposta de um Gerenciador de Grupos de Usuários para o MoCA.

5.2 Definição

A proposta é que seja implementado um serviço no MoCA que dê suporte à criação e ao gerenciamento de grupos de usuários. Utilizando o Gerenciador de Grupos de Usuários, será possível cadastrar usuários de um sistema como pertencentes a um determinado grupo. Este grupo se torna “ativo” quando os usuários deste grupo estiverem reunidos fisicamente. Quando um grupo é ativado, vários serviços poderão ser disponibilizados para os participantes do grupo. Por exemplo, quando o grupo estiver reunido, os participantes poderão ter acesso a

arquivos produzidos por membros daquele grupo, ou um participante poderá compartilhar informações de seu perfil com o restante do grupo.

Uma nova informação de contexto será criada para representar o grupo quando ele estiver ativo, dessa maneira, cada participante terá como uma de suas informações de contexto a informação de que está participando de um grupo no momento. Se o grupo ativo naquele momento for de uma equipe de projeto, por exemplo, essa informação pode ser útil para indicar que os participantes estão numa reunião.

Cada grupo de usuários pode ser configurado para que seus serviços se tornem disponíveis quando todo o grupo estiver reunido, ou para que os serviços fiquem acessíveis mesmo quando somente alguns membros do grupo estiverem reunidos. Também será possível configurar se o grupo pode se ativar em qualquer região ou se o grupo só se ativará quando estiver reunido numa determinada região.

Além dessas funcionalidades, também existirá a possibilidade da criação de grupos dinamicamente, em tempo real, no momento em que pessoas que compartilhem interesses semelhantes estejam reunidas. A idéia é utilizar o dispositivo móvel como um meio de aproximar pessoas e estimular a colaboração entre elas. Quando pessoas que compartilham um mesmo interesse (configurado previamente) estiverem reunidas, o dispositivo móvel pode avisá-las sobre isso, e a partir daí as pessoas podem interagir pessoalmente, ou trocando informações (como arquivos ou dados pessoais) pelos seus dispositivos móveis.

5.3 Serviços

Nesta seção as funcionalidades apresentadas para o Gerenciador de Grupos de Usuários serão mais detalhadas. Para cada funcionalidade serão expostos o seu objetivo, seus parâmetros e retorno. Os parâmetros são os elementos necessários como entrada de uma funcionalidade. O retorno é o resultado da ação de cada função. Essa estruturação é semelhante à utilizada na linguagem de programação Java (linguagem de programação utilizada no MoCA), por isso essa denominação foi utilizada nesta seção.

As funcionalidades estarão divididas em dois grandes grupos. O primeiro grupo envolve as funções do Gerenciador de Grupos de Usuários propriamente dito. O

segundo grupo envolve funções de um serviço de Perfil dos Usuários. Este último é necessário para possibilitar algumas funções propostas para o gerenciador de grupos, já que algumas dessas funcionalidades fazem uso de informações pessoais dos usuários.

Serviços do Gerenciador de Grupos de Usuários

Dentro dessa seção as funções serão divididas em subgrupos, para facilitar o seu entendimento.

Infra-estrutura básica

Dentro da infra-estrutura básica do Gerenciador de Grupos de Usuários temos as funções que formam a base do serviço.

- ***Criar grupo***
 - *Objetivo:* Permitir o cadastro de um grupo no sistema. O grupo será formado por um conjunto de usuários, informações de configuração e irá possuir serviços que serão disponibilizados aos participantes do grupo.
 - *Parâmetros:*
 - Nome do grupo.
 - Mensagem de exibição que descreva o grupo. Esta mensagem servirá como informação de contexto dos participantes do grupo quando ele estiver ativo.
 - Usuário moderador do grupo.
 - Usuários cadastrados no grupo.
 - Informações de configuração:
 - Que indiquem se o grupo deve se tornar ativo com a presença de alguns participantes ou somente quando todos os participantes do grupo estiverem reunidos. O grupo se tornar ativo significa que os serviços implementados por aquele grupo se tornarão disponíveis para seus participantes.

- Que indiquem se o grupo se ativa sempre que estiver reunido ou somente quando o moderador (ou algum participante) acionar o grupo.
 - Que indiquem se o grupo pode se ativar em qualquer região, ou somente numa localização específica.
 - Que indiquem o raio de atuação do grupo: quão próximos os participantes de um grupo devem estar para que o grupo seja ativado. Devem estar todos na mesma região? A alguns metros de distância?
- *Retorno:* O grupo criado.
- ***Adicionar usuário***
 - *Objetivo:* Permitir que novos usuários sejam adicionados em grupos já existentes.
 - *Parâmetros:*
 - Usuário a ser adicionado no grupo.
 - Grupo onde o usuário será adicionado.
 - *Retorno:* O grupo com um novo usuário cadastrado.
- ***Remover usuário***
 - *Objetivo:* Permitir que usuários sejam retirados de grupos já existentes.
 - *Parâmetros:*
 - Usuário a ser removido do grupo.
 - Grupo de onde o usuário será removido.
 - *Retorno:* O grupo com o usuário removido.
- ***Grupo está ativo***
 - *Objetivo:* Indicar se um dado grupo está ativo no momento.
 - *Parâmetros:*
 - Grupo a ser testado.
 - *Retorno:* Uma indicação se o grupo está ativo ou não no momento.

- **Ativar grupo**
 - *Objetivo:* Permitir a ativação de um grupo mesmo que nem todos os participantes estejam presentes. Nesse caso, o grupo será criado envolvendo os usuários que estiverem próximos de quem ativar esta função.
 - *Parâmetros:*
 - Grupo a ser ativado.
 - *Retorno:* O grupo é ativado, e os serviços do grupo se tornam disponíveis para os participantes próximos de quem ativou a função.

- **Desativar grupo**
 - *Objetivo:* Permitir a desativação de um grupo mesmo que todos os participantes estejam presentes.
 - *Parâmetros:*
 - Grupo a ser desativado.
 - *Retorno:* O grupo é desativado, e os serviços do grupo se tornam indisponíveis para seus participantes.

Grupos dinâmicos

Dentro deste subgrupo temos a possibilidade de criação de grupos dinâmicos (que foram brevemente discutidos na seção anterior). Grupos dinâmicos são grupos criados em tempo real quando pessoas que compartilham interesses semelhantes se encontram reunidas num mesmo espaço físico. Os interesses pessoais poderão ser configurados utilizando os serviços de perfil dos usuários, que estão descritos mais à frente no documento.

- **Ativar grupos dinâmicos**
 - *Objetivo:* Ativar a função de grupos dinâmicos. Com essa função ativada, o sistema passará a monitorar suas regiões e usuários. Quando usuários com interesses semelhantes estiverem próximos, o sistema ativará um novo grupo formado por esses usuários. Grupos dinâmicos não precisam ser pré-cadastrados, basta que esta função esteja ativa, que sempre que usuários com interesse semelhantes se aproximem um grupo será formado (mediante

aceitação por parte dos usuários) para permitir interação entre seus participantes.

- *Parâmetros:*
 - Quais serviços serão disponibilizados para grupos criados dinamicamente:
 - Compartilhamento de arquivos.
 - Troca de perfis pessoais (informações dos participantes do grupo).
 - Quantas pessoas com o mesmo interesse devem estar reunidas para que um novo grupo seja criado.
 - De que maneira o sistema avisa aos usuários que existem pessoas próximas a ele que compartilham interesses?
 - Através de uma mensagem.
 - De um alerta.
 - Através de um convite para se juntar ao grupo.
 - Informação que indique se grupos dinâmicos podem se ativar em qualquer região, ou somente numa localização específica.
 - Informação que indique o raio de atuação de grupos dinâmicos: quão próximos usuários devem estar para que o grupo dinâmico seja ativado. Devem estar todos na mesma região? A alguns metros de distância?
- *Retorno:* Um grupo dinâmico é criado e seus participantes podem usufruir dos seus serviços. Após o distanciamento dos usuários, ou desativação do grupo por um deles, o grupo deixa de existir.

- ***Enviar convite para grupo dinâmico***

- *Objetivo:* Permitir a adição de um usuário num grupo dinâmico mesmo que ele não compartilhe o mesmo interesse que os outros participantes. Essa funcionalidade tem como objetivo se aproveitar das características socializantes dos grupos dinâmicos, para permitir a adição de pessoas que estão presentes fisicamente, mas ficaram de fora do grupo formado.
- *Parâmetros:*
 - Usuário a ser adicionado no grupo.

- Grupo onde o usuário será adicionado.
- *Retorno*: O usuário é adicionado no grupo dinâmico.

Comunicação dentro do grupo

Dentro deste subgrupo serão especificados serviços que permitem a comunicação e interação entre membros de um grupo de usuários.

- ***Compartilhar dados de perfil***
 - *Objetivo*: Permitir que usuários enviem seus dados pessoais para outros participantes de um grupo. Os dados pessoais do usuário podem ser: seu nome, seu e-mail e interesses pessoais, por exemplo. Essa função é particularmente útil junto ao conceito de grupos dinâmicos, para permitir interações entre pessoas desconhecidas.
 - *Parâmetros*:
 - Usuários que irão receber os dados.
 - Dados pessoais que serão enviados.
 - *Retorno*: Os dados pessoais são enviados para outros participantes do grupo.

- ***Enviar mensagem***
 - *Objetivo*: Permitir que usuários enviem mensagens para outros participantes do grupo. Uma mensagem poderá ser enviada para todo o grupo de usuários.
 - *Parâmetros*:
 - Mensagem a ser enviada.
 - Usuários que irão receber a mensagem.
 - *Retorno*: A mensagem é enviada para seus destinatários.

Compartilhamento de arquivos

Dentro deste subgrupo serão especificados serviços que permitem o compartilhamento de arquivos entre usuários de um grupo.

- ***Compartilhar arquivo***

- *Objetivo:* Permitir que usuários compartilhem arquivos entre os membros de um grupo.
 - *Parâmetros:*
 - Arquivo a ser compartilhado.
 - Usuários dentro do grupo que terão acesso ao arquivo (poderá ser todo o grupo, ou um subconjunto do grupo de acordo com o perfil do usuário, por exemplo).
 - Informação de configuração que indique se o arquivo estará sempre disponível aos participantes, ou somente quando o grupo de usuários estiver ativo.
 - *Retorno:* O arquivo se tornará compartilhado entre membros do grupo.
- ***Receber arquivos compartilhados***
 - *Objetivo:* Permitir que usuários recebam arquivos que estão sendo compartilhados por algum membro do grupo.
 - *Parâmetros:*
 - Arquivo a ser baixado.
 - *Retorno:* O arquivo será baixado.

Serviços de Perfil dos Usuários

Para possibilitar os serviços descritos acima, precisamos de uma classe de funcionalidades que permita criar usuários, com um perfil associado a esses usuários. Essas funcionalidades são necessárias para servir como uma infraestrutura de base para os serviços do Gerenciados de Grupos de Usuários.

É necessário que exista um mecanismo de criação de usuários, e que os usuários possuam informações como dados pessoais e informações de perfil. A idéia é que o serviço provido seja genérico o suficiente para permitir a implementação de aplicações diversas. De acordo com a aplicação que estiver sendo desenvolvida, o analista do sistema poderá escolher que características deve possuir o usuário, para possibilitar a aplicação que irá utilizar o Gerenciador de Grupos de Usuários.

5.4 Comunicação com o MoCA

Para possibilitar o funcionamento do Gerenciador de Grupos de Usuários, vários componentes da estrutura existente do MoCA deverão ser utilizados.

Para permitir a implementação do serviço que ativa um grupo (quando os participantes do grupo estão reunidos) devem ser utilizados métodos do LIS, que possibilitam a localização de usuários na rede. Por exemplo, após a criação de um grupo, o sistema deve passar a mapear a localização de todos os usuários da rede para conseguir interpretar quando um determinado grupo estiver reunido. Para isso devem ser utilizadas as funções que resgatam as localizações de todos os aparelhos que se encontram na rede.

Para realizar a implementação dos serviços de comunicação interna do grupo, podem ser utilizados métodos das API's cliente e servidor do MoCA. Utilizando essas API's é possível o envio e recebimento de objetos. Esses objetos podem ser perfis de usuários (no caso da funcionalidade *compartilhar perfil*) ou mensagens de texto (no caso da funcionalidade *enviar mensagem*).

Para permitir a funcionalidade de grupos dinâmicos, devem ser utilizados métodos do ECI. Este módulo permite que clientes registrem interesse em eventos específicos e sejam notificados quando um destes eventos ocorrer. Utilizando este serviço, o Gerenciador de Grupos de Usuários permite aplicações onde uma pessoa se cadastre como interessada em participar de grupos dinâmicos. Sempre que um evento de formação de um grupo dinâmico envolvendo aquela pessoa ocorrer, ela será notificada. Para permitir a formação de grupos dinâmicos, também são necessárias a utilização do LIS, para conseguir identificar a localização dos usuários, e informações dos interesses específicos de cada usuário. Esses interesses estarão disponíveis através dos serviços de Perfil dos Usuários, descritos na seção anterior.

Para permitir o compartilhamento de arquivos, deverá ser utilizado mais uma vez o ECI. Através dele o usuário pode cadastrar seu interesse em compartilhar e ter seus arquivos compartilhados. Quando o grupo estiver ativo, o ECI irá verificar o desejo do usuário de compartilhar arquivos e irá disponibilizar esse serviço para o grupo. O compartilhamento em si (envio e recebimento de arquivos) será

possibilitado pelas API's cliente e servidor do MoCA, que como já foi explicado, permitem o envio e recebimento de objetos.

5.5 Aplicações

Para finalizar a definição do Gerenciador de Grupos de Usuários serão apresentadas algumas aplicações que podem surgir a partir desse novo serviço.

A aplicação MyOfficeSpace pode utilizar o Gerenciador de Grupos de Usuários para expandir suas funcionalidades. Além das funcionalidades já descritas, o MyOfficeSpace poderia ter uma função que permitisse a criação de grupos. Seria possível então criar grupos com as equipes de trabalho dos funcionários de uma empresa, ou grupos quaisquer que envolvam os membros da empresa. Dentro das propriedades do grupo, seriam configuradas suas características, como por exemplo, em que região o grupo seria ativado. Dessa maneira, seria possível que um determinado grupo se tornasse ativo somente quando estivesse na sala de reunião (grupo da equipe de um projeto), enquanto outro grupo se ativasse em outro local, como a sala de convivência (grupo de amigos). Após a criação do grupo e do cadastro de seus membros, todos poderiam desfrutar dos serviços disponibilizados por ele. Vamos, em seguida, descrever um possível cenário da utilização do Gerenciador de Grupos de Usuários no MyOfficeSpace.

Vamos considerar a criação de um grupo que envolvesse os membros da uma equipe de projeto. A aplicação MyOfficeSpace avisaria a todos sobre uma reunião (baseado na informação da agenda de tarefas do grupo), como já foi explicado, essa funcionalidade já existe no MyOfficeSpace. Após a chegada de todos na sala de reunião, o grupo de usuários da equipe de projeto se tornaria ativo. Seriam disponibilizadas funções como cadastrar um novo usuário no grupo, por exemplo, que seria útil no caso de um consultor que esteja participando pela primeira vez de uma reunião. Após o cadastro desse consultor, ele poderia receber informações de perfil (contatos pessoais) de todos os participantes do grupo, de maneira fácil e rápida durante a reunião. Durante uma reunião é comum que sejam produzidos artefatos como atas de reunião, imagens de um quadro, notas pessoais, etc. Todos esses artefatos poderiam ser compartilhados entre os membros daquele grupo, durante a própria reunião.

Outras novas aplicações podem surgir a partir do Gerenciador de Grupos. Poderíamos ter por exemplo uma aplicação que rodasse nos dispositivos móveis de participantes de uma conferência. Através da função de grupos dinâmicos, um usuário poderia cadastrar seus interesses (áreas de estudo, por exemplo) e ser notificado sempre que uma ou mais pessoas, com o mesmo interesse, estivessem próximas dele. A partir daí, os usuários poderiam trocar contatos pessoais ou até mesmo compartilhar arquivos. Uma pessoa que está próxima ao grupo, mas ficou de fora do grupo dinâmico por não compartilhar os mesmos interesses, pode ser convidada a entrar no grupo e compartilhar informações. A funcionalidade de grupos dinâmicos permite uma série de aplicações que estimulam a interação física entre pessoas, se aproveitando do conceito de usar aplicações móveis como uma maneira de facilitar a interação “real”, “olho a olho”, entre pessoas.

6 Conclusões

Neste trabalho, apresentamos inicialmente uma pesquisa sobre aplicações móveis cientes de contexto, onde foram analisados os conceitos definidos nessa área, tipos de aplicações cientes de contexto, questões envolvendo privacidade e por fim alguns exemplos de aplicações existentes. Seguindo esta apresentação, foi realizada uma pesquisa sobre o MoCA, um middleware que permite a implementação de aplicações móveis cientes de contexto. Foram apresentados os componentes do middleware, sua arquitetura e seu funcionamento, além de alguns exemplos de aplicações que já foram desenvolvidas utilizando o middleware.

Neste trabalho, também desenvolvemos uma aplicação móvel ciente de contexto utilizando o MoCA, o MyOfficeSpace. Essa aplicação serviu como uma prova do conceito estudado, e permitiu pôr em prática o que foi aprendido sobre ciência de contexto e MoCA. Em seguida, foi especificado um novo serviço para o MoCA, o serviço de Gerenciador de Grupos de Usuários. Este serviço foi proposto com a intenção de permitir o surgimento de novas aplicações usando o MoCA, provendo funcionalidades que não existem atualmente no middleware. A estrutura que foi proposta para o novo serviço ainda pode ser mais especificada, no entanto, ela serve como um passo inicial para a implementação do serviço.

Após o estudo geral do tema, pudemos identificar que, apesar de existirem inúmeras implementações de aplicações utilizando conceitos de ciência de contexto, nenhuma delas se configura como uma *killer application*. As aplicações estudadas ou não possuem um forte apelo ao usuário, ou possuem uma usabilidade muito fraca, que inviabiliza seu uso. Está faltando encontrar um nicho de aplicações, ou uma aplicação, que realmente consiga unir essas duas características, trazendo benefícios reais para o usuário, e utilizando a ciência de contexto como um meio para permitir uma grande aplicação. Atualmente a ciência de contexto tem sido vista como um conceito tecnológico em busca de grandes aplicações. Apesar dessa carência de *killer applications*, as ferramentas e dispositivos necessários para o funcionamento de aplicações móveis cientes de contexto já estão disponíveis, como é o caso do MoCA por exemplo. Estudos recentes envolvendo novas tecnologias, como Mobile TV e a JSR272, que irá permitir TV digital móvel interativa, e a aplicação de ciência de contexto em áreas como a medicina, podem abarcar futuras *killer applications*.

6.1 Trabalhos Futuros

Ao concluir este trabalho, podemos identificar alguns encaminhamentos futuros no sentido de aperfeiçoar a especificação do Gerenciador de Grupos de Usuários proposto nesse trabalho. É necessário realizar uma especificação mais completa e de mais baixo nível do serviço proposto. Neste trabalho, a especificação do gerenciador de grupos se encontra no nível abstrato de funcionalidades, com indicações de como essas funcionalidades irão utilizar a estrutura existente do MoCA. É necessário descer o nível e realizar a especificação de como cada funcionalidade será implementada na prática.

Após a especificação mais detalhada do Gerenciador de Grupos de Usuários, este serviço poderá ser implementado e integrado ao MoCA. Após a implementação do serviço, sugerimos o desenvolvimento de aplicações de prova de conceito que utilizem o serviço implementado juntamente com outras funcionalidades oferecidas pelo MoCA. Os estudos de caso descritos no final do capítulo 5 podem ser usados como base nesse desenvolvimento.

6.2 Considerações Finais

Com base no que foi estudado nesse trabalho, podemos concluir que aplicações móveis cientes de contexto é um tema riquíssimo com grandes chances de futuramente abarcar várias *killer applications*. Para o surgimento dessas aplicações é necessário unir um sentimento de necessidade no usuário, uma ótima usabilidade e uma infra-estrutura disponível, de baixo custo. Com a proposta do novo serviço neste trabalho, esperamos agregar valor ao MoCA e ao campo da ciência de contexto, possibilitando o surgimento de aplicações móveis cientes de contexto inovadoras.

7 Referências Bibliográficas

[Abowd et al., 1997]

Abowd, G. D.; Atkeson, C. G.; Hong, J.; Long, S.; Kooper, R.; Pinkerton, M. *Cyberguide: A mobile context-aware tour guide*. *Wireless Networks*, 3(5):421-433, October 1997.

[Asthana et al., 1994]

Asthana, A.; Cravatts, M.; Krzyzanowski, P. *An indoor wireless system for personalized shopping assistance*. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 69-74, Santa Cruz, California, USA, December 1994.

[Bardram, 2004]

Bardram, J. E. *Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles*. In *Proceedings of ACM Symposium on Applied Computing (ACM SAC)*, pages 1574-1579, 2004.

[Benett et al., 1994]

Bennett, F.; Richardson, T.; Harter, A. *Teleporting - making applications mobile*. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 82-84, Santa Cruz, California, USA, December 1994.

[Brown et al., 2000]

Brown, P.; Burleston, W.; Lamming, M.; Rahlff, O.; Romano, G.; Scholtz, J.; Snowden, D. *Context-awareness: some compelling applications*. 2000. Disponível em: <http://www.dcs.ex.ac.uk/~pjbrown/papers/huc2k.pdf>

[Chen & Kotz, 2000]

Chen, G.; Kotz, D. *A Survey of Context-Aware Mobile Computing Research*. Dartmouth Computer Science Technical Report TR2000-381, Department of Computer Science, Dartmouth College, New Hampshire, USA, 2000. Disponível em: <http://elans.cse.msu.edu/ni/restrict/ChenKotz2000.pdf>

[Dey & Abowd, 1999]

Dey, A.K.; Abowd, G.D. *Towards a Better Understanding of context and context-awareness*. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, June 1999. Disponível em: <http://www.it.usyd.edu.au/~bob/IE/99-22.pdf>

[Dey et al., 1999]

Dey, A. K.; Futakawa, M.; Salber, D.; Abowd, G. D. *The Conference Assistant: Combining Context-Awareness with Wearable Computing*. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99)*, pages 21-28, San Francisco, California, USA, October 1999.

[Helal et al., 2003]

Helal, S.; Winkler, B.; Choonhwa, Y.; Ran, L.; Giraldo, C.; Kuchibhotla, S.; Mann, W. *Enabling Location-Aware Pervasive Computing Applications for the Elderly*. 1st IEEE Conference on Pervasive Computing and Communications, Fort Worth, March, 2003.

[Klante et al., 2004]

Klante, P.; KrÄosche, J.; Boll, S. *Accessights – a multimodal tourist information system*. In Proceedings of the ICCHP, 2004.

[Lamming et al., 2000]

Lamming, M.G.; Eldridge, M.; Flynn, M.; Jones, C.; Pendlebury, D. *Satchel: providing access to any document, any time, anywhere*. Published in ToCHI, 2000.

[Long et al., 1996]

Long, S.; Kooper, R.; Abowd, G. D.; Atkeson, C. G. *Rapid prototyping of mobile context-aware applications: the Cyberguide case study*. In Proceedings of the Second Annual International Conference on Mobile Computing and Networking, pages 97-107, White Plains, New York, USA, November 1996.

[Mayo, 2000]

Mayo, B. *Factoid*. Compaq Western Research Lab., 2000. Disponível em: <http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-60.pdf>

[Marmasse and Schmandt, 2000]

Marmasse, N.; Schmandt, C. *Location-aware information delivery with ComMotion*. In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, pages 157-171, Bristol, UK, September 2000.

[MoCA website, 2006]

MoCA Website. Disponível em: <http://www.lac.inf.puc-rio.br/moca/>. Último acesso: 27/09/2006.

[Moran & Dourish, 2001]

Moran, T.; Dourish, P. *Introduction on Context Aware Computing*. Published on to the special issue of Human-Computer Interaction, Vol. 16, 2001. Disponível em: <http://hci-journal.com/editorial/si-context-aware-intro.pdf>

[Pascoe, 1998]

Pascoe, J. *Adding generic contextual capabilities to wearable computers*. In Proceedings of the Second International Symposium on Wearable Computers, Pittsburgh, Pennsylvania, USA, October 1998.

[Pascoe et al., 1998]

Pascoe, J.; Morse, D.; Ryan, N. *Developing personal technology for the Field*. Personal Technologies, 2(1), March 1998.

[Pascoe et al., 2000]

Pascoe, J.; Morse, D.R.; Ryan, N.S. *HCI issues in fieldwork environments*. ACM TOCHI, pages 417-437, 2000.

[Rodrigues et al., 2004]

Rodrigues, V.J.S.; Endler, M.; Rubinsztejn, H.K.; Lima, L.; Gonçalves, K.M.; Nascimento, F.N.; Bueno, G.A. *MoCA: A Middleware for Developing Collaborative Applications for Mobile Users*. PUC-RioInf.MCC44/04 December, 2004. Disponível em: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1355907

[Schilit et al., 1994]

Schilit, B.; Adams, N. ; Want, R. *Context-Aware Computing Applications*. Appeared in the IEEE Workshop on Mobile Computing Systems and Applications, December 1994. Disponível em: <http://seattleweb.intel-research.net/people/schilit/wmc-94-schilit.pdf>

[Schmidt et al., 1999]

Schmidt, A.; Aidoo, K.A.; Takaluoma, A.; Tuomela, U.; Laerhoven, K.V., Velde, W.V. *Advanced interaction in context*. In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC 1999, pages 89-101, Karlsruhe, Germany, September 1999. Disponível em: http://www.teco.uni-karlsruhe.de/~albrecht/publication/huc99/advanced_interaction_context.pdf

[Smailagic et al., 2001]

Smailagic, A.; Siewiorek, D.P.; Anhalt, J.; et al. *Towards Context Aware Computing: Experiences and Lessons Learned*, IEEE Journal on Intelligent Systems, Vol. 16, No. 3, pp 38-46, June 2001. Disponível em: <http://www.cs.cmu.edu/~asim/DistractioFreeComputing.pdf>

[Spreitzer & Theimer, 1993]

Spreitzer, M.; Theimer, M. *Providing location information in a ubiquitous computing environment*. In Proceedings of the Fourteenth ACM Symposium on Operating Systems Principles, pages 270-283, Asheville, North Carolina, USA, December 1993. Disponível em: <http://portal.acm.org/citation.cfm?id=168641&dl=ACM&coll=portal>

[Want et al., 1992]

Want, R.; Hopper, A.; Falcão, V.; Gibbons, J. *The Active Badge location system*. ACM Transactions on Information Systems, January 1992. Disponível em: <http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=128759>

[Want et al., 1995]

Want, R.; Schilit, B.; Adams, N. I.; Gold, R.; Petersen, K.; Goldberg, D.; Ellis, J. R.; Weiser, M. *An overview of the PARCTAB ubiquitous computing experiment*. IEEE Personal Communications, pages 28-43, December 1995.

[Want et al., 1996]

Want, R.; Schilit, B.; Adams, N. I.; Gold, R.; Petersen, K.; Goldberg, D.; Ellis, J. R.; Weiser, M. *The ParcTab Ubiquitous Computing Experiment*. In Mobile Computing, chapter 2, Kluwer Academic Publishers, 1996.

[Weiser, 1993]

Weiser, M. *Some computer science issues in ubiquitous computing*. Communications of the ACM, pages 75-84, July 1993.

Assinaturas

Recife, 05 de outubro de 2006.

Carlos André Guimarães Ferraz (Orientador)

André Galamba Rodrigues dos Anjos (Aluno)