Modelo conceitual e requisitos de projeto de um serviço de privacidade para aplicações baseadas em localização

Vagner Sacramento, Markus Endler*e Clarisse Souza

[1]Departamento de Informática, PUC-Rio
R. Marquês de São Vicente 225
22453-900, Rio de Janeiro, Brazil

{vagner,endler,clarisse}@inf.puc-rio.br

Abstract. *Privacy is a very complex, subjective and far reaching topic to deal with. There are many theoretical works about the design of privacy services for location-based applications, but just few describe practical experience and methods that aid developers of such applications in this task. With this purpose, this article discusses a conceptual model for a privacy service to be used by a community of users and discusses some privacy control requirements that should be considered for the design of this service. Most of these requirements have guided the design and implementation of the "***Co***ntext* ***P****rivacy* ***S****ervice" CoPS, which in turn has been integrated to and deployed with the context provisioning services of the MoCA (****M****obile* ***C****ollaboration* ***A****rchitecture) middleware.*

Resumo. *Privacidade é uma questão muito subjetiva, abrangente e complexa a ser tratada, modelada e implementada. Existem muitos trabalhos teóricos sobre o projeto de serviços de privacidade para aplicações baseadas em localização, mas há poucos que descrevem métodos práticos para auxiliar os desenvolvedores de tais aplicações nessa tarefa. Com esse propósito, este artigo discute um modelo conceitual para um serviço de privacidade a ser utilizado por uma comunidade de usuários e discute alguns requisitos de controle de privacidade que devem ser considerados no projeto e implementação desse serviço. Tais requisitos foram levados em conta na implementação do "***Co***ntext* ***P****rivacy* ***S****ervice" CoPS, que por sua vez foi integrado e implantado com os serviços de provisão de contexto do middleware MoCA (****M****obile* ***C****ollaboration* ***A****rchitecture).*

1. Introdução

A partir da evolução das tecnologias de rede sem fio e localização (e.g., RFID, GPS), vários tipos de serviços e aplicações sensíveis a localização (também conhecidas como aplicações LBS - *Location-Based Services*) têm sido propostos por diferentes grupos de pesquisa [MoCATeam 2005a, Gonçalves et al. 2004, Hightower and Borriello 2001]. Entretanto, enquanto essas inovações tecnológicas oferecem uma série de benefícios interessantes para os usuários tais como usfruir de um guia turístico ou localizador de rotas, etc, elas também introduzem novos riscos e ameaças a privacidade dos mesmos.

Na computação móvel e sensível à localização, existem vários níveis de ameaças à privacidade da informação de localização do usuário. Essa pode ser comprometida tanto

*Parcialmente financiado pelos projetos de pesquisa CNPq proc. nr. 552.068/02-0 (Projeto ESSMA) e nr. 479824/04-5.

pelos protocolos de enlace/rede quanto pelos serviços de contexto e aplicações LBS (que inferem e fazem uso da informação de localização, respectivamente). Tais possibilidades tornam-se cada vez mais factíveis dada as inúmeras ferramentas e agentes existentes que implementam ataques automatizados através da comunicação, observação e inferência de comportamento dos usuários [Gorlach et al. 2004].

No nível da comunicação, um agente malicioso poderia explorar vulnerabilidades ou facilidades dos protocolos e aplicações com o propósito de obter e revelar a localização do dispositivo do usuário na rede. Isso pode ser feito de diversas formas, como, por exemplo, explorando o comportamento dos sensores Bats e *Actives Badges* [Ward et al. 1997] que propagam a sua localização abertamente, ou das interfaces 802.11, que transmitem os seus endereços MAC por meio de *broadcast*. A partir desses identificadores e da identificação lógica do usuário (e.g., e-mail, nome do usuário) que está usando o equipamento é possível inferir a localização aproximada do mesmo. Tais riscos tornam-se ainda mais evidentes nos serviços de contexto que inferem a localização dos usuários ou nas aplicações LBS que fazem uso da mesma, pois esses lidam diretamente com a informação de localização e, em princípio, podem divulgá-la ou utilizarem-na de forma maliciosa e indevida. Como podemos ver, as ameaças à privacidade da informação de localização dos usuários podem estar presentes em todos os níveis da comunicação. No entanto, neste trabalho, nós restringimos o nosso foco de pesquisa na proposta de um modelo de privacidade que auxilia os usuários a controlarem a privacidade no nível da aplicação, ou seja, auxiliá-los a determinar como, quando e para quem o serviço de contexto poderá divulgar a localização.

Com o objetivo de oferecer um método sistemático para auxiliar os desenvolvedores a identificarem, compreenderem e priorizarem questões importantes ligadas ao projeto e implementação de um serviço de privacidade, nós descrevemos neste trabalho um modelo conceitual e uma lista de requisitos de projeto que apresentam em linhas gerais a integração e uso de um serviço de privacidade com um terceiro (i.e., provedor de contexto) responsável por processar e divulgar a localização dos usuários para aplicações LBS. Tais requisitos delinearam o projeto e implementação do **Co**ntext **P**rivacy **S**ervice (CoPS) [Sacramento et al. 2005b], que por sua vez, foi integrado aos serviços de contexto da arquitetura MoCA (**Mo**bile **C**ollaboration **A**rchitecture) [Sacramento et al. 2004, MoCATeam 2005a].

Algumas questões e discussões do modelo conceitual e dos requisitos de privacidade foram adotados de trabalhos anteriores, e derivados da análise dos resultados de uma pesquisa sobre as preferências de privacidade que realizamos com aproximadamente 120 usuários. Alguns resultados parciais da pesquisa estão disponíveis em [Sacramento et al. 2005a, MoCATeam 2005b]. No entanto, por seguir uma abordagem centralizada e por definir um conjunto específico de requisitos, nós estamos cientes de que o modelo proposto não se aplica a cenários *ad hoc* no qual a inferência e a privacidade são tratadas de forma integrada e nem atende a todos possíveis requisitos de uma aplicação LBS.

O restante do artigo está estruturado como segue: na Seção 2 é descrito o modelo conceitual e as hipóteses adotadas sobre o uso do serviço de privacidade em um ambiente organizacional. Na Seção 3 são discutidos os requisitos de propósito geral que devem ser considerados no projeto de um serviço de privacidade. Em seguida, são descritos nas

Seções 4 e 5, uma visão geral da arquitetura e implementação do CoPS. Por fim, nas Seções 6 e 7 são apresentados os trabalhos correlatos e as considerações finais.

2. Modelo Conceitual

Neste artigo, apresentamos um modelo conceitual de um serviço de privacidade que trata principalmente das questões de privacidade relacionadas ao acesso à informação de localização, apesar de poder ser utilizado para gerenciar o acesso a outras informações de contexto (e.g., contexto computacional e pessoal do usuário). No modelo, cada usuário possui uma localização, representada por regiões simbólicas (organizadas hierarquicamente), que pode variar no tempo e no espaço. A informação de localização é processada e inferida pelo serviço de contexto e repassada para as aplicações LBS. Essas aplicações usam a localização do usuário para prover serviços ao próprio usuário ou a terceiros.

O modelo do serviço de privacidade proposto é formado por várias entidades cujos papéis são descritos como segue:

- *Requester* (um usuário intermediado por uma aplicação LBS) é a entidade que, após devidamente autenticada, solicita o acesso à informação de localização de um Subject divulgada por um serviço de contexto;
- *Subject* é o usuário que tem a sua localização inferida pelo serviço de contexto;
- *PolicyMaker* é o usuário responsável por definir (ou redefinir, caso já exista) a política de privacidade. Esse pode ou não ser o próprio Subject, pois o modelo proposto permite que tanto o usuário Subject quanto o administrador do sistema definam as políticas de privacidade;
- *Serviço de contexto* é a entidade responsável por processar as requisições dos Requesters, inferir e compartilhar a informação de localização do Subject mediante a autorização avaliada por um serviço de privacidade;
- *Aplicação LBS* é o meio de interação/comunicação através do qual o Requester requisita o acesso à informação de localização do Subject;
- *Serviço de privacidade* é a ferramenta através da qual o Subject controla o acesso e o compartilhamento da sua informação de localização no escopo das concessões e restrições da política de privacidade definidas pelo PolicyMaker.

2.1. Padrão de Interação

A Figura 1 ilustra o padrão de interação entre as entidades do modelo do serviço de privacidade.

Inicialmente, o PolicyMaker (e.g., o Subject) define a política de privacidade através da interface de gerenciamento de políticas/regras (1). Em paralelo, o serviço de contexto recebe periodicamente dados de sensores para inferir a localização do Subject (2). Entretanto, a localização do usuário somente será divulgada mediante as restrições impostas pela sua política de privacidade. A autenticidade da identidade do Requester deve ser garantida por algum serviço de autenticação da rede. Quando a requisição de acesso à informação de localização é recebida pelo serviço de contexto (3), ela é processada e repassada para o serviço de privacidade. Se a requisição do Requester é permitida, o serviço de privacidade responde com uma mensagem “Grant”, caso contrário responde com um resultado “Deny” ou “Not Available” (4).

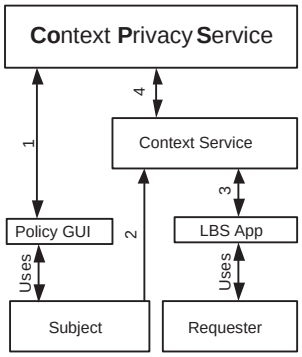

Figura 1. Padrão de Interação

2.2. Hipóteses do Modelo

Todo modelo é uma abstração da realidade que, para ser consistente e definir bem o seu escopo, precisa estar fundamentado em algumas hipóteses sobre as entidades do sistema. A seguir, enunciamos e discutimos as principais hipóteses adotadas no modelo.

- O Requester, Subject e o PolicyMaker devem ter uma única identidade não forjável;
- O serviço de privacidade não trata das possíveis implicações do uso da informação de localização fora do seu propósito e contexto previsto;
- O serviço de privacidade deve ser utilizado em uma comunidade de usuários na qual as pessoas têm uma certa relação de confiança entre si, por exemplo, por trabalharem juntas ou estarem associadas diretamente à mesma organização ou departamento, ou por simplesmente se conhecerem pessoalmente.

Em nosso modelo, estamos considerando o uso do serviço de privacidade dentro de uma comunidade de usuários para tirar proveito da relação de confiança intrínseca ao ambiente social. Essa abordagem facilita a identificação dos possíveis tipos de requisitantes que devem ser considerados na definição da política de privacidade. A partir dos estudos reportados em [Lederer et al. 2003], podemos concluir que, o uso do serviço de privacidade dentro de uma comunidade de usuários em que as pessoas se conhecem pessoalmente e as suas identidades não podem ser forjadas, simplifica significativamente a definição e o gerenciamento da política de privacidade. Pois os usuários (e.g., Subjects) demonstraram uma maior predisposição para ajustar a precisão da sua informação a ser revelada em função da identidade do Requester do que das circunstâncias em que a requisição foi recebida.

No entanto, a visão de comunidade ou do que ela representa é algo estritamente pessoal. Por isso, mesmo dentro da sua percepção de comunidade, os usuários podem ter a necessidade de manter a sua privacidade de uma maneira muito peculiar. Por exemplo, o Subject pode querer omitir ou negar acesso à sua localização em função da identidade do Requester, das circunstâncias, da granularidade da informação requisitada, dentre outros. Neste trabalho, tais necessidades são contempladas pelo modelo conceitual proposto.

2.3. Cenários de Aplicações

Para ilustrar os tipos de aplicações atendidas pelo modelo conceitual do serviço de privacidade, são descritos a seguir dois cenários de uso de aplicações LBS e algumas questões de privacidade associadas ao uso das mesmas.

O primeiro cenário considera o uso de uma aplicação, conhecida como *People Finder*, através da qual o Requester solicita explicitamente a localização de um dado usuário para algum propósito específico, como, por exemplo, saber o local da reunião de um grupo de estudo do qual ambos fazem parte. No entanto, os usuários deste tipo de aplicação (i.e., os Subjects) podem ter certas preocupações com relação a sua privacidade. Por exemplo, a dificuldade de manter-se oculto às percepções de terceiros, pois, os Requesters, além de obterem a localização, poderiam também deduzir certas informações sobre o padrão de comportamento dos Subjects. Por exemplo, a razão do seu atraso para uma reunião, o porquê do não comparecimento à aula, onde geralmente almoça, etc.

O segundo cenário ilustra o uso de aplicações LBS do tipo *Mural Virtual*[1] [Gonçalves et al. 2004] baseado em localização. Através dessa aplicação, usuários, por exemplo, professores e alunos, podem postar mensagens para determinados lugares geográficos representados como regiões simbólicas (e.g., Sala de aula 511, Corredor Sul do prédio RDC, Auditório 5, etc) cadastradas no serviço de localização. Por exemplo, um professor poderia postar uma mensagem para seus alunos em uma dada sala de aula informando que chegará 15 minutos atrasado. Para entregar essa mensagem aos usuários presentes (ou que entrarem) na sala de aula, o servidor dessa aplicação solicita periodicamente a localização de todos os usuários registrados. Ao contrário do cenário de uso da aplicação *People Finder*, esta aplicação requisita ao serviço de contexto, sem a intervenção de nenhum usuário, a informação de localização para desempenhar suas funções.

No entanto, tal comportamento introduz certas preocupações de privacidade no que diz respeito ao uso da informação de localização obtida periodicamente. Após ter a sua localização divulgada sob as imposições e restrições da sua política de privacidade, o Subject perde completamente o controle sobre a mesma. Ou seja, ele não sabe como e por quanto tempo a sua informação de localização persistirá no servidor da aplicação, se esse mantém o histórico da mesma ou, até mesmo, se esse servidor divulgará tal informação pela rede. Além disso, o usuário da aplicação *Mural Virtual* pode receber mensagens de Spam que caracterizam uma invasão de privacidade. Contudo, ele não poderia bloqueá-las através da sua política de privacidade. Nessa situação, cabe ao administrador da aplicação moderar o conteúdo das mensagens postadas através dessa aplicação.

3. Requisitos do Serviço de Privacidade

A partir dos cenários discutidos na seção anterior, fica evidente a necessidade do serviço de privacidade contemplar funcionalidades que possibilitem aos usuários permitir, omitir ou negar acesso a sua localização em função da identidade do requisitante, do momento, do contexto corrente, dentre outros. Nesta seção, são discutidos alguns requisitos de privacidade que atendem a tais questões e que auxiliam os usuários a definirem e refinarem as suas políticas gradativamente, de acordo com as suas necessidades. Esses requisitos delinearam as decisões de projeto e implementação do CoPS [Sacramento et al. 2005b].

3.1. Configuração da Política de Privacidade

Para atender às necessidades de privacidade de ambientes organizacionais e/ou de indivíduos específicos, o serviço de privacidade deve organizar as políticas de privacidade

[1] De fato, uma aplicação desse tipo foi implementada em nosso grupo.

em uma hierarquia de três níveis: política da organização, do usuário e política padrão. A política em nível da organização é definida pelo administrador do sistema e tem maior precedência sobre as demais. Essa política é utilizada pela corporação para impor-se perante as políticas dos usuários por alguma necessidade específica, por exemplo, exigir que a localização dos mesmos esteja sempre disponível para uma aplicação de controle de situações emergenciais (e.g., incêndios), independentemente se a política do usuário está negando acesso à mesma. Por uma questão de flexibilidade, a implementação dessa política é facultativa, pois nem sempre essa imposição é necessária ou aceitável por parte dos usuários. A política em nível de usuário é definida pelo próprio Subject e tem uma maior precedência em relação à política padrão definida pelo administrador do sistema.

Através da política padrão, um *template* de regras que é associado a cada novo usuário do sistema, a organização pode determinar, inicialmente, uma política que contempla as principais necessidades de privacidade dos usuários com o objetivo de minimizar os esforços na configuração inicial da política de privacidade. Em [Palen 1999, Patil and Lai 2005] são apresentados resultados que demonstram que a grande maioria dos usuários não muda a configuração padrão do sistema e das aplicações. Sendo assim, a política padrão pode ser muito útil, principalmente, na fase inicial de uso de uma aplicação LBS, pois os usuários não precisarão, de antemão, configurar a sua própria política para ter o mínimo de privacidade desejável.

O serviço de privacidade também deve oferecer três políticas de controle de acesso: Reservado, Liberal e Sob-Demanda. No modo Reservado, por definição, todas as requisições são negadas, exceto aquelas que casam com alguma regra que libera o acesso. No controle de acesso Liberal, por definição, todas as requisições são aceitas, exceto aquelas que casam com alguma regra que explicitamente nega o acesso. No controle de acesso Sob-Demanda, o resultado a ser aplicado às requisições é obtido interativamente, a partir de uma consulta ao Subject.

Para cada tipo de política de controle de acesso, o PolicyMaker pode definir um conjunto de regras que determinará sob quais condições as informações do Subject serão reveladas. O Subject poderá mudar a sua política de controle de acesso corrente através da interface gráfica de configuração de regras. Tais políticas oferecem certas flexibilidades que amenizam o ônus da configuração da política de privacidade, pois após escolher a política de acesso Reservado ou Liberal, o PolicyMaker terá somente que especificar regras com um dos seguintes resultados: "Grant" ou "Deny" (mas não ambos), "Not Available" ou "Ask me". Além disso, o PolicyMaker (i.e., o Subject) tem a opção de não definir nenhuma regra inicialmente e, gradativamente, através da política Sob-Demanda, configurar a sua política de privacidade interativamente.

O resultado "Not Available" satisfaz o requisito de *plausible deniability* [Hindus et al. 2001, Hong and Landay 2004]. O objetivo desse tipo de resposta é negar o acesso à informação sem que o requisitante saiba que o acesso foi negado. Ao receber como resposta uma mensagem "Not Available", o requisitante não saberá se a resposta esperada não foi obtida por causa de uma falha técnica do sistema, de um problema de comunicação, ou se a localização não pode ser inferida ou se o Subject negou acesso à mesma. O resultado "Ask Me" de uma regra de privacidade faz com que o serviço envie uma consulta de autorização de acesso ao Subject. Ao contrário da política Sob-Demanda, o resultado "Ask Me" pode ser utilizado em uma regra específica para configurar um con-

trole de acesso interativo, no qual, o sistema deve interagir com o Subject para deferir o resultado a ser aplicado à requisição recebida.

Vale ressaltar que a configuração de uma nova regra de privacidade oferece ao Subject o bônus da privacidade da informação controlada e, por outro lado, o ônus de ter que se lembrar de desfazê-la ou removê-la quando a regra em questão não se aplica mais as suas necessidades correntes. Com objetivo de amenizar o ônus da re-configuração da política de privacidade, o serviço de privacidade deve permitir que o PolicyMaker defina regras temporárias que são automaticamente desabilitadas e removidas após um determinado período.

Segundo a pesquisa publicada em [Patil and Lai 2005], grupos de usuário oferecem uma maior flexibilidade para controlar o acesso às informações pessoais, facilitam a identificação dos requisitantes e diminuem consideravelmente o esforço envolvido na configuração das regras. Sendo assim, para usufruir de tais benefícios, o serviço de privacidade deve permitir o uso de grupos na configuração das regras de privacidade.

3.2. Refinamento e Manutenção da Política de Privacidade

As questões relacionadas à definição e manutenção da política de privacidade representam os maiores desafios a serem atendidos pelos serviços de privacidade. Sendo assim, as seguintes funcionalidades devem ser oferecidas aos usuários para auxiliá-los nessas tarefas: notificações de acesso ao contexto (i.e., acesso à localização), relatório de estatística de acesso, controle de acesso interativo e regras temporárias.

Durante a configuração de uma regra, o PolicyMaker deve poder especificar o tipo de notificação (e.g., e-mail, mensagem SMS) a ser enviada ao Subject quando for recebida uma requisição. Nós acreditamos que tal funcionalidade aumenta o nível de transparência sobre quem está tentando acessar o que e com que periodicidade e, conseqüentemente, pode ser utilizada como um parâmetro de decisão para a atualização das regras de privacidade. Além disso, essa funcionalidade cria, implicitamente, um protocolo social entre os usuários da comunidade que pode evitar determinadas ações maliciosas. O fato dos Requesters estarem cientes de que o Subject pode estar sendo notificado a cada tentativa de acesso, pode inibir certas atitudes que caracterizam uma invasão de privacidade. Por exemplo, usando a aplicação *People Finder*, um Requester (e.g., o orientador de João) poderia se sentir intimidado em fazer repetidas consultas à localização de João para não criar uma situação embaraçosa/constrangedora entre eles.

Além das notificações, o serviço de privacidade deve oferecer também relatórios de acesso à informação de localização representados hierarquicamente pela granularidade dos acessos no ano, meses, semanas e dias. Através destes, é possível obter informações estatísticas dos acessos bem ou mal sucedidos de um Requester específico (ou de grupos de Requesters). Além disso, para facilitar a manutenção da política de privacidade, o usuário poderá visualizar nos relatórios no nível de dias a regra que permitiu ou bloqueou uma dada tentativa de acesso. O usuário também pode configurar em suas preferências a periodicidade em que ele gostaria de receber via e-mail um relatório com as estatísticas de acesso de um período específico. Nós acreditamos que tais informações aumentam o nível de controle e confiança do usuário para com o sistema e aumenta o nível de transparência sobre quem conseguiu ou não conseguiu acessar o quê, com que periodicidade tais tentativas ocorreram, transparece a disparidade (aumento ou decréscimo) do número

de tentativas de acesso entre dias, semanas, ou meses diferentes, dentre outros. A partir dessas informações, se necessário, o usuário pode configurar alguma contramedida a um determinado evento que está sendo permitido ou bloqueado.

Além das funcionalidades supracitadas, nós acreditamos que a política de acesso Sob-Demanda e a tag "Ask Me" também auxiliam o processo de manutenção da política do usuário. Essas permitem ao Subject implementar um controle de acesso interativo através do qual ele pode determinar, gradativamente, o que deve ser permitido ou negado a determinados grupos ou indivíduos. Além disso, conforme já discutido, as regras temporárias também diminuem os esforços de gerenciamento da política de privacidade.

3.3. Controle de acesso

Com intuito de oferecer um controle de acesso flexível, o serviço de privacidade deve permitir ao Subject ajustar a granularidade temporal, espacial e a precisão da informação a ser revelada. Por exemplo, considere um cenário em que o usuário João pretende compartilhar sua localização com seus colegas de sala de aula para que eles possam se coordenar através da aplicação *People Finder*. No entanto, João pode não se sentir confortável em compartilhar a sua localização exata. Neste caso, ele poderia ajustar a granularidade espacial da sua informação de localização revelando que está no prédio "RDC" da PUC-Rio ao invés da "Sala 512". João também poderia implementar uma restrição temporal limitando o acesso a sua localização a um grupo específico de requisitantes somente em um determinado horário (e.g., de segunda à sexta, entre 9:00 e 12:00 am). E, se necessário, ele também pode especificar a precisão (*freshness*) da informação a ser revelada, determinando que ao invés da sua localização corrente, somente a localização conhecida de 30 minutos atrás deve ser divulgada. Para atender essa última funcionalidade, o serviço de contexto deve manter um histórico da informação a ser divulgada.

Com base nos trabalhos de Goffman em [Goffman 1956], nós incorporamos diferentes papéis no convívio em sociedade que revela a nossa face ou aspectos diferentes sobre nós mesmos. Por exemplo, muitas pessoas assumem e mantêm diferentes posturas e estereótipos no relacionamento com os seus subordinados ou superiores dentro do ambiente de trabalho, diferentemente do estado descontraído e brincalhão com os amigos mais íntimos ou familiares. Isso nos leva a acreditar que, para o controle da privacidade, alguns usuários desejam definir a sua política de privacidade em função do papel ou mais especificamente do estado corrente. Alguns desses estados, por exemplo, "Descansando", "Em Reunião", "Ocupado", dentre outros, pode representar o contexto corrente do usuário que somente ele próprio pode determinar e, para cada papel/contexto, provavelmente, o usuário deseja definir uma política de privacidade mais adequada à situação.

Para satisfazer esse requisito, o serviço de privacidade deve permitir o PolicyMaker criar *Perfis de privacidade*, para os quais, ele possa definir políticas de acesso em função do contexto em que o Subject pode estar engajado. Ou seja, ao selecionar manualmente o perfil "Em Reunião", somente as regras que regem a política de privacidade do usuário nessa situação (ou atividade) serão habilitadas. De uma certa forma, essa funcionalidade também auxilia os usuários a terem uma visão/percepção mais clara do que está sendo permitido ou negado pela sua política de privacidade corrente. Pois, após selecionar um determinado perfil, por exemplo, "Descansando", João estaria ciente de que a única pessoa que poderia localizá-lo, através da aplicação *People Finder*, seria a sua esposa.

Além dos perfis de privacidade, o serviço de contexto deve permitir que o usuário fique "invisível" a requisições de terceiros de maneira fácil e simples. Para tanto, o serviço deve oferecer uma funcionalidade, chamada *"Modo Invisível"*, a partir da qual, o sistema negará acesso à localização do usuário retornando uma mensagem "Not Available" aos requisitantes para tirar proveito da *plausible deniability*.

Baseando-se nos resultados reportados em [Adams 2000], nós definimos no modelo proposto que o serviço de privacidade deve permitir que a aplicação envie ao Subject uma descrição do "Contrato de uso do contexto". Este tem o intuito de amenizar as questões de privacidade relacionadas ao uso de aplicações do gênero da aplicação *Mural Virtual*. Tal contrato pode apresentar ao usuário uma descrição clara dos custos/benefícios em divulgar a informação de localização em diferentes granularidades, explicitando por quanto tempo e para quê tal informação será utilizada, se será divulgada para terceiros, etc. Naturalmente, nesse cenário, configura-se uma relação de confiança entre o usuário e a aplicação, pois aquele não saberá se essa honrará o contrato pré-estabelecido entre eles. Dentro da comunidade, cabe ao administrador do sistema identificar e adotar medidas preventivas contra aplicações maliciosas que tendem a desrespeitar tais contratos.

Para oferecer uma maior flexibilidade, o serviço de privacidade deve implementar um controle de acesso de granularidade fina através de um algoritmo de especificidade que escolhe, de forma determinística, a regra mais específica da política de privacidade do usuário que deverá avaliar uma dada requisição. Esse nível de granularidade permite o usuário especificar em cada regra uma restrição ou uma determinada ação a ser executada, caso ela seja selecionada para avaliar a requisição de um determinado grupo ou indivíduo específico. Por exemplo, para cada regra, o usuário pode especificar um tipo de notificação diferente, ajustar a granularidade da informação a ser revelada, restringir o acesso a um grupo específico em um período/horário predeterminado, etc.

4. Arquitetura do CoPS

Nesta seção, é descrita brevemente a arquitetura do serviço de privacidade de contexto (CoPS - **C**ontext **P**rivacy **S**ervice) que contempla os requisitos de privacidade discutidos na Seção 3. A arquitetura do CoPS, ilustrada na Figura 2, oferece um controle de acesso de granularidade fina e flexível, que permite ao usuário definir e gerenciar a sua política de privacidade, gradativamente, no decorrer do uso da aplicação LBS.

4.1. Interação entre as Entidades do Serviço

Em linhas gerais, o CoPS é formado por um servidor e duas APIs clientes. O servidor gerencia o acesso às informações de contexto e as APIs ocultam do desenvolvedor alguns detalhes envolvidos na comunicação com o CoPS. A API *Context Access Authorization (CAA)* é utilizada pelo serviço de contexto para enviar ao servidor CoPS requisições de autorização de acesso. A API *User and Policy Management (UPM)* é utilizada pelas aplicações clientes do Subject e Requester para acessar e analisar logs, verificar a consistência das regras de privacidade, dentre outros.

A interação entre esses elementos, ilustrada na Figura 3, é similar ao descrito no padrão de interação do serviço de privacidade na Seção 3. As principais diferenças consistem do uso das APIs *CAA* e *UPM* e a implementação da autenticação do usuário no próprio serviço de privacidade.

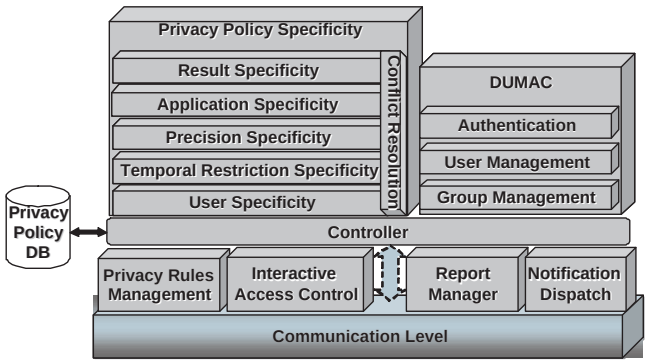

Figura 2. Arquitetura geral do CoPS

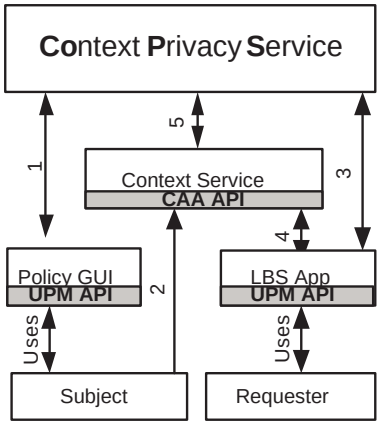

Figura 3. Interação entre cliente e servidor

4.2. Visão Geral dos Componentes da Arquitetura

Uma visão geral dos componentes da arquitetura é ilustrada na Figura 2. O componente *Communication Level* provê interfaces de comunicação síncrona e assíncrona para transmissão/recepção de dados não encriptados e encriptados via SSL (*Secure Socket Layer*). Esse componente implementa os servidores de comunicação que interagem com as APIs usadas pelas aplicações para realizar determinadas tarefas, tais como autenticação, processamento da autorização de acesso à localização, etc. As requisições recebidas pelo *Communication Level* são repassadas para o componente *Controller*, que por sua vez, interpreta o tipo de requisição recebida e interage com os demais componentes da arquitetura para desempenhar a ação solicitada. Para otimizar o tempo de resposta, o *Controller* gerencia um *pool* de *threads* para processar as requisições recebidas.

Os principais componentes da arquitetura são o *Privacy Rules Management* e o *Privacy Policy Specificity*. Estes são utilizados para processar as requisições de gerenciamento das regras ou alterações na configuração da política de privacidade que atendem necessidades específicas do usuário. Dentre as suas funcionalidades, esses componentes são responsáveis por: identificar e gerenciar as regras de privacidade sensíveis ao contexto, implementar e gerenciar as políticas de privacidade hierárquicas, gerenciar a configuração do Modo Invisível, administrar regras de privacidade temporárias e gerenciar as operações básicas de administração de regras (e.g., adicionar, remover, . . .).

4.3. Estrutura das regras de privacidade

Toda regra de privacidade está associada a uma política de acesso padrão (e.g., Reservado, Liberal ou Sob-Demanda). Tal associação é feita pelo Policy Maker e determinará o algoritmo básico de avaliação para cada requisição. Os campos das regras de privacidade e seus significados são descritos como segue.

- *Policy Maker*: Usuário que definiu/criou a regra de privacidade (pode ou não ser o próprio Subject);
- *Subject*: Usuário ou entidade cuja informação de contexto (e.g., localização) é controlada pela regra de privacidade;

- *Requester*: Usuário ou componente de software que requisita acesso ao dado de contexto do Subject;
- *Context Variable*: Tipo específico de dado de contexto requisitado pelo Requester (e.g., localização do Subject);
- *Application*: Lista de nomes das aplicações que podem ser utilizadas pelo Requester para acessar a variável de contexto. O coringa "*" representa qualquer aplicação;
- *Precision*: Especifica a precisão ou granularidade do valor da variável de contexto a ser divulgada (e.g., para informação de localização, este atributo poderia ser Prédio, Andar, Sala, etc.);
- *Temporal Restriction*: Restrições de hora e data para divulgar a informação de contexto (e.g., dias da semana, das 9:00 às 14:00);
- *Freshness*: Especifica quão recente deve ser a informação de contexto a ser disponibilizada para um dado Requester (e.g., revelar somente a localização inferida há 15 minutos atrás, ou revelar a localização corrente);
- *Timestamp*: Registra o horário em que a regra de privacidade foi criada ou atualizada.
- *AccessPolicy*: Representa a política de acesso (Reservado, Liberal ou Sob-Demanda) com a qual a regra de privacidade está associada;
- *Policy Level*: Nível da hierarquia da regra de privacidade. O CoPS prover suporte às seguintes hierarquias: "Organization", "Individual" ou "Default";
- *Result*: Resultado a ser aplicado à requisição. Os possíveis valores são: "Not Available", "Ask Me", "Grant" e "Deny";
- *Notify Me*: Tipo de notificação a ser enviada para o Subject quando a regra é selecionada para avaliar uma requisição. Por exemplo, "NoNotification", "E-Mail", ou "SMS".

4.4. Avaliação da Política de Privacidade

A avaliação da política de privacidade é uma das principais questões tratada pelo CoPS para oferecer um controle de acesso flexível na avaliação das requisições. Para tanto, o *Controller* usa o componente *Privacy Policy Specificity* para avaliar as requisições de autorização de acesso à informação de localização. Esse componente implementa o algoritmo de especificidade responsável por avaliar as requisições de acordo com a política de privacidade do Subject. Primeiro ele seleciona as regras da política de controle de acesso corrente escolhida pelo PolicyMaker, e avalia a especificidade da política de privacidade do usuário com o propósito de selecionar a regra mais específica dentre aquelas que casam com a requisição recebida. Durante o processo de avaliação, mais de uma regra pode casar com a requisição por várias razões. Por exemplo, quando o Requester pertence a vários grupos mencionados no campo "Requester" de algumas regras (e.g., "Maria" pode pertencer aos grupos "Coworker" e "MyFriend" referenciados pelas regras de João). Com base no conjunto de regras selecionadas, o algoritmo de especificidade checa e resolve possíveis conflitos para processar o resultado final a ser aplicado ("Not Available", "Ask Me", "Grant" ou "Deny").

Em linhas gerais, o algoritmo de especificidade funciona da seguinte forma: dado um conjunto de regras selecionado previamente para avaliar uma requisição, o algoritmo identifica a regra mais específica desse conjunto comparando os campos das estruturas das regras na seguinte ordem de prioridade: *Subject*, *Requester*, *Temporal Restriction*, *Precision*, *Application* e *Result*. Ao comparar as regras com relação a um determinado campo, somente aquelas com o valor mais específico neste campo são selecionadas para

Tabela 1. Status de implementação dos Requisitos de privacidade

Requisitos	Status de implementação
Hierarquia das políticas de privacidade	Implementado
Políticas de controle de acesso	Implementado
Plausible Deniability	Implementado
Regras temporárias	Pendente
Grupos de usuários	Implementado
Notificações de acesso ao contexto	Pendente
Relatório de estatísticas de acesso	Parcialmente
Controle de acesso interativo	Pendente
Ajuste de granularidade da localização	Implementado
Restrição temporal	Implementado
Precisão (Freshness)	Implementado
Política de privacidade sensível ao contexto	Pendente
Modo invisível	Implementado
Contrato de uso de informações de contexto	Pendente
Algoritmo de especificidade	Implementado

a análise de especificidade posterior, enquanto que as demais regras não são consideradas na seleção/avaliação seguinte. Desta forma, mesmo se duas ou mais regras tem diferentes especificidades relativas (i.e., elas diferem em dois ou mais campos), o algoritmo pode identificar a regra mais específica analisando esses campos de acordo com suas prioridades. Detalhes da descrição e implementação do algoritmo de especificidade podem ser encontrados em [Sacramento et al. 2005b].

5. Implementação

O CoPS e as APIs *UPM* e *CAA* foram implementados em Java e estão disponíveis para *download* em [MoCATeam 2005a]. A Tabela 1 mostra o *status* da implementação dos requisitos discutidos na Seção 3. Dentre esses, destacam-se aqueles implementados pelos componentes *Privacy Policy Specifity* e *Privacy Rules Management* utilizados na avaliação e gerenciamento das regras de privacidade, respectivamente.

Para testar as funcionalidades projetadas e implementadas no CoPS, nós integramos a API *CAA* ao serviço de inferência de localização (LIS) da arquitetura MoCA e pretendemos avaliar as características do serviço proposto adaptando as aplicações LBS [MoCATeam 2005a] implementadas através do LIS, integrando-as com a API *UPM* para tratar as questões de privacidade.

O foco desse artigo centra-se na descrição conceitual do modelo e dos requisitos de privacidade que devem ser considerados no projeto de um serviço de privacidade, e na descrição geral da arquitetura que contempla tais questões. No entanto, em [Sacramento et al. 2005b] descrevemos mais detalhadamente a implementação do referido modelo e requisitos na arquitetura do CoPS, e discutimos a avaliação de alguns testes de desempenho que analisam o quanto o tempo gasto no processamento do CoPS influencia no tempo de aquisição do contexto.

6. Trabalhos Correlatos

Existem vários trabalhos que fazem uma discussão prescritiva e analítica sobre requisitos, dificuldades e desafios dos sistemas sensíveis a privacidade. [Palen and Dourish 2003] descrevem que privacidade não é simplesmente um problema de controle de acesso, mas ao invés disso é um processo contínuo das negociações das fronteiras da revelação, identidade e tempo. Esse artigo discute de forma sistemática algumas questões que

nos ajudaram a compreender melhor o relacionamento entre privacidade e tecnologia da informação. [Barkhuus and Dey 2003] discutem o quanto serviços baseado em localização podem ser considerados intrusivos à privacidade dos usuários e discutem se serviços de localização centralizados trazem mais riscos de privacidade do que serviços de posicionamento implementados pelo próprio dispositivo. Em [Patil and Lai 2005] Patil & Lai discutem o quanto o controle de privacidade entre pessoas que se conhecem pessoalmente afeta suas atitudes para tornarem-se mais cautelosas e conservadoras.

Essas e outras discussões foram de suma importância para fundamentar e justificar as decisões de projeto relacionadas ao modelo conceitual e aos requisitos de privacidade implementados no CoPS. Além disso, existem várias outras propostas de arquiteturas que tratam de questões de privacidade de localização. O grupo IETF Geopriv [Cuellar et al. 2002] definiu alguns requisitos de privacidade nos quais um Objeto de Localização contém a informação de localização a ser divulgada pelo servidor de localização para o requisitante e contém as regras de acesso que regem a política de privacidade dos usuários. Semelhante a alguns requisitos definidos neste artigo, o Geopriv também descreve que a localização pode ser divulgada em diferentes granularidades e sob determinadas restrições temporal. No entanto, em função da abrangência da proposta e da complexidade de lidar com questões de privacidade de uma forma mais genérica, várias questões ligadas ao gerenciamento e manutenção da privacidade são omitidas.

A Context Fabric (Confab) [Hong and Landay 2004] é uma arquitetura para provisão de informação de localização com controle de privacidade. Nesta arquitetura, a informação de localização é inferida, armazenada e gerenciada no dispositivo do usuário final para dar um maior controle ao mesmo. No entanto, essa abordagem pode comprometer a usabilidade do sistema, pois ela exige que a política de privacidade do usuário esteja restrita a um equipamento específico, que tal dispositivo tenha uma maior capacidade de processamento, armazenamento e bateria. A Confab contempla uma série de requisitos de privacidade que oferecem certas flexibilidades no uso de uma aplicação sensível a privacidade. No entanto, essa arquitetura não implementa um algoritmo de especificidade de granularidade fina que permite o usuário configurar um controle de acesso em nível de regra para diferentes usuários ou grupos de requisitantes.

7. Conclusão

Neste artigo, nós descrevemos um modelo conceitual de um serviço de privacidade em uma infra-estrutura de provisão de contexto (e.g., localização) centralizada e descrevemos algumas hipóteses que devem ser consideradas ou tratadas em um serviço do gênero. Além disso, com base em estudos de trabalhos relacionados e a partir da nossa experiência, descrevemos uma série de requisitos relativos à definição, manutenção da política de privacidade e mecanismos de controle de acesso que podem ser utilizados como base para o projeto e implementação de um serviço de privacidade. Esses fundamentaram e embasaram o projeto e implementação do CoPS, que por sua vez, foi integrado junto aos serviços de provisão de contexto da arquitetura MoCA com intuito de prover uma infra-estrutura de middleware que ofereça suporte ao desenvolvimento de aplicações sensíveis ao contexto e a privacidade.

Nós descrevemos o modelo conceitual e os requisitos de privacidade com o propósito de refinar as discussões conceituais abstratas de privacidade dentro de questões

e possíveis soluções concretas para tratar as preocupações de privacidade no uso de aplicações LBS. No entanto, estamos cientes de que parte do projeto do modelo conceitual proposto atende somente a determinados tipos de sistemas, por exemplo, a definição dos papéis e a interação entre as entidades do modelo conceitual são específicas para um serviço de privacidade centralizado. Além disso, os requisitos definidos não contemplam ou tratam todas as possíveis questões a serem consideradas no projeto de um serviço de privacidade (nem é nossa pretensão propor algo do gênero, se é que isso é possível). Entretanto, esses servem como guia para os desenvolvedores refletirem sobre o impacto social e organizacional da privacidade no uso de aplicações sensíveis a localização e, na medida do possível, projetarem e implementarem as possíveis soluções discutidas.

Um outro desafio relacionado às questões de privacidade consiste da metodologia de avaliação da usabilidade e flexibilidade do serviço de privacidade em face ao gerenciamento contínuo da privacidade dos usuários. Considerando que a necessidade e a percepção de privacidade é estritamente dependente do usuário, surgem inúmeros desafios sobre a forma de expor as funcionalidades do serviço para o usuário final de uma forma simples e não intrusiva. Pois, tacitamente, o usuário expressa o desejo de ter privacidade/controle sobre a sua localização no uso de uma aplicação LBS, mas geralmente ele não deseja ser exposto aos detalhes de configuração e administração das suas regras para desfrutar dos benefícios providos pela aplicação. Tais desafios fazem parte dos nossos trabalhos futuros, que podem ser resumidos como segue: definir e implementar uma metodologia para avaliar a usabilidade do CoPS através de cenários de uso de aplicações LBS, estender as aplicações LBS implementadas para tratar as questões de privacidade discutidas, projetar uma interface para a definição das regras de privacidade.

Referências

Adams, A. (2000). Multimedia information changes the whole privacy ballgame. In *CFP '00: Proceedings of the tenth conference on Computers, freedom and privacy*, pages 25–32, New York, NY, USA. ACM Press.

Barkhuus, L. and Dey, A. K. (2003). Location-based services for mobile telephony: a study of users' privacy concerns. In *INTERACT*.

Cuellar, J., Morris, J. B., and Mulligan, D. (2002). Ietf geopriv requirements.

Goffman, E. (1956). *The Presentation of Self in Everyday Life*. Doubleday, New York.

Gonçalves, K., Rubinsztejn, H., Endler, M., Silva, B., and Barbosa, S. (2004). Um aplicativo para comunicação baseada em localização. In *VI Workshop de Comunicação sem Fio e Computação Móvel*, pages 224–231.

Gorlach, A., Heinemann, A., and Terpstra, W. W. (2004). Survey on location privacy in pervasive computing. In *Workshop on Security and Privacy in Pervasive Computing*.

Hightower, J. and Borriello, G. (2001). Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66.

Hindus, D., Mainwaring, S. D., Leduc, N., Hagstrom, A. E., and Bayley", O. (2001). Casablanca: designing social communication devices for the home. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 325–332, New York, NY, USA. ACM Press.

Hong, J. I. and Landay, J. A. (2004). An architecture for privacy-sensitive ubiquitous computing. In *MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189. ACM Press.

Lederer, S., Mankoff, J., and Dey, A. K. (2003). Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 724–725, New York, NY, USA. ACM Press.

MoCATeam (2005a). Moca home page. http://www.lac.inf.puc-rio.br/moca (Last visited: April 2006).

MoCATeam (2005b). Results of the user survey about privacy and spontaneous collaboration. http://www-di.inf.puc-rio.br/∼endler/pub/Survey-Privacy-Results.htm (Last visited April 2006).

Palen, L. (1999). Social, individual and technological issues for groupware calendar systems. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24, New York, NY, USA. ACM Press.

Palen, L. and Dourish, P. (2003). Unpacking "privacy" for a networked world. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–136, New York, NY, USA. ACM Press.

Patil, S. and Lai, J. (2005). Who gets to know what when: configuring privacy permissions in an awareness application. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 101–110, New York, NY, USA. ACM Press.

Sacramento, V., Endler, M., and Nascimento, F. N. (2005a). Design of a context privacy service for mobile collaboration. In *SBRC '2005: Proc. do 23rd Simpósio Brasileiro de Redes de Computadores*, volume 1, pages 323–336.

Sacramento, V., Endler, M., and Nascimento, F. N. (2005b). A privacy service for context-aware mobile services. In *SecureComm '2005: Proc. of the First IEEE/CreatNet International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 182–193. IEEE Computer Society Press.

Sacramento, V., Endler, M., Rubinsztejn, H. K., Lima, L. S., Goncalves, K., Nascimento, F. N., and Bueno, G. A. (2004). Moca: A middleware for developing collaborative applications for mobile users. *IEEE Distributed Systems Online*, 5(10):2.

Ward, A., Jones, A., and Hopper, A. (1997). A new location technique for the active office. *IEEE Personnel Communications*, 4(5):42–47.