UFSM

Trabalho de Graduação

Serviço de Comunicação Consciente do Estado da Rede em um Ambiente Pervasivo

Rafael Pereira Pires

Curso de Ciência da Computação

Santa Maria, RS, Brasil

2005

Serviço de Comunicação Consciente do Estado da Rede em um Ambiente Pervasivo

por

Rafael Pereira Pires

Trabalho de Graduação apresentado ao Curso de Ciência da Computação – Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação.

Curso de Ciência da Computação

Trabalho de Graduação nº 210

Santa Maria, RS, Brasil

2005

Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Graduação

Serviço de Comunicação Consciente do Estado da Rede em um Ambiente Pervasivo

elaborado por

Rafael Pereira Pires

como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Comissão Examinadora:

Prof^a. Dr^a. Iara Augustin
(Orientador)

Prof. João Carlos Damasceno Lima
(Co-orientador)

Prof^a. Dr^a. Márcia Pasin

Prof. Antônio Marcos de Oliveira Cândia

Santa Maria, 22 de dezembro de 2005.

Agradecimentos

São tantas as pessoas que contribuíram de alguma forma para a realização deste trabalho e para que eu chegasse aonde estou. Agradeço a todas elas: aos professores, colegas, familiares, amigos e conhecidos.

Em especial, gostaria de agradecer à minha orientadora, Iara Augustin, pelo apoio e disposição em ajudar durante praticamente toda graduação. Ao Caio, pelo humor, objetividade nas sugestões de implementação e pelas reuniões do GMob nas sextas à tardinha no posto.

Agradeço também aos fraternos colegas Rubens, Cristóvão, Aline e Reck, amigos desde o início do curso, pela parceria de sempre e pelas longas tardes ociosas regadas a café na sala do PET. Ao Cristóvão, pela amizade e companheirismo nos trabalhos da faculdade, festas, praia e carnaval. A saudade é grande.

Aos demais colegas, sou grato pela troca de idéias, dicas e opiniões. Acredito que o maior aprendizado ocorre fora da sala de aula, com a convivência e a troca de experiências. Em especial, agradeço ao Redin, colega de GMob, e à gurizada do LSC: Veiga, Edmar e Elton. Espero que possamos manter contato nessa próxima etapa em que estamos ingressando.

Sou grato também ao pessoal confirmado no DCE/Aldeia, ao sempre amigo, Iuri, aos colegas do ensino médio que até hoje me comunico e aos demais amigos conquistados antes e durante a graduação.

Agradeço imensamente aos meus pais, Jorge e Rejane, pelo carinho e apoio incondicional às minhas decisões e aos meus pedidos. Aos meus irmãos, Gaspar e Bárbara, agradeço pelo companheirismo, amizade e até pelas briguinhas e discussões

- faz parte. Aos meus avós, tias, tios e primos, agradeço por todas as palavras e ações positivas que me atingiram de alguma forma.

Finalmente, agradeço ao cara lá de cima, pelas oportunidades que me são oferecidas e pelas pessoas que me rodeiam.

Sumário

Lista de Figuras

RESUMO

Trabalho de Graduação
Ciência da Computação
Universidade Federal de Santa Maria

Serviço de Comunicação Consciente do Estado da Rede em um Ambiente Pervasivo

Autor: Rafael Pereira Pires

Orientador: Prof^a. Dr^a. Iara Augustin

Data e Local da Defesa: Santa Maria, 22 de dezembro de 2005.

Este trabalho descreve a elaboração de um serviço de comunicação consciente do estado da rede para dar suporte às aplicações do projeto pBuy. O projeto pBuy busca introduzir características de um ambiente pervasivo a um sistema legado, chamado Portal de Compras, instalado na UFSM. Considerando a instabilidade na comunicação inerente à comunicação sem fio e as desconexões voluntárias causadas pela economia dos restritos recursos dos dispositivos portáteis, situações comuns no ambiente pervasivo, este serviço se faz necessário para permitir o acesso às aplicações do Portal de Compras via dispositivos móveis (PDAs e telefones celulares) e mantê-las funcionais em face a desconexões.

Capítulo 1

Introdução

A Computação Ubíqua [WEI 91] propõe que a computação seja integrada e invisível à vida do usuário. Esta visão enfatiza ainda a centralização nas atividades do usuário, onde o computador se ajusta às suas necessidades, diferente da situação atual, onde o usuário deve aprender a lidar com o computador.

A Computação Pervasiva é um cenário computacional que surgiu pelas possibilidades introduzidas pelos equipamentos portáteis e redes sem-fio. Com uma visão mais realista do que a Computação Ubíqua para a tecnologia disponível atualmente, a computação pervasiva está se consolidando como o ambiente computacional do futuro [SAH 2003]. Nesta visão, o poder computacional está sempre disponível, se encontra em qualquer lugar, a todo momento e é acessível com qualquer dispositivo [IBM 2005]. A computação pervasiva propõe o deslocamento da computação para a centralização no usuário e suas atividades. Nesta visão, quem deve ser reconhecido pelo sistema é o usuário e não os equipamentos que ele porta ou usa (como estão definidos os sistemas computacionais atuais).

A computação móvel, na década de 90, introduziu o uso de dispositivos portáteis (PDAs) e, mais recentemente, o acesso a recursos computacionais via telefones celulares. A natureza de suas propriedades - portabilidade, mobilidade e conectividade - introduz restrições aos sistemas e aplicações [AUG 2004]. Apesar da evolução natural da tecnologia, acredita-se que as limitações dos dispositivos portáteis, como bateria, tamanho do display, capacidade de processamento, capacidade de armazenamento e largura de banda da rede móvel, permanecerão limitados, principalmente

se comparados ao ambiente de rede fixa [SAT 2001]. Desta forma, a construção de aplicações para o ambiente pervasivo deve levar em conta essas restrições naturais a fim de superá-las, ou, ao menos, amenizá-las.

As aplicações pervasivas requerem um novo modelo de projeto e execução para expressar a semântica 'siga-me': as aplicações seguem o usuário conforme este se desloca no espaço [AUG 2004]. Neste modelo, as aplicações são distribuídas, móveis e conscientes do contexto (adaptam-se aos recursos disponíveis no momento e local onde o usuário se encontra).

A origem da motivação deste trabalho está no projeto sendo desenvolvido pelo grupo GMob (Grupo de Pesquisa em Sistemas de Computação Móvel - UFSM), chamado pBuy , que objetiva a introdução de características de pervasividade em um sistema legado, chamado Portal de Compras, instalado na UFSM, de forma que o usuário do sistema obtenha acesso a este independente do lugar e do dispositivo que está usando. Dentre os estudos realizados para a modelagem do pBuy, identificou-se que as aplicações necessitarão de um serviço de comunicação que trate as características peculiares dos dispositivos portáteis.

Este trabalho descreve, então, a modelagem e implementação de um serviço que possibilita a comunicação do ponto de vista dos dispositivos portáteis (PDAs e telefones celulares) considerando as implicações que o ambiente pervasivo impõe. O serviço de comunicação está permanentemente disponível no dispositivo móvel (PDA, celular) e é a ponte de comunicação entre as aplicações do pBuy que executam no dispositivo com o mundo externo. O serviço também é responsável por identificar o estado da rede e adaptar-se a essa situação. Estratégias de *caching* são utilizadas nos momentos de desconexões.

O resto do texto está organizado da seguinte forma: o capítulo 2 apresenta a revisão da literatura; o capítulo 3 mostra a arquitetura dos serviços do projeto pBuy; o capítulo 4 detalha a modelagem do serviço de comunicação bem como as tecnologias utilizadas na solução. Considerações finais e conclusões são registradas no capítulo 5.

Capítulo 2

Revisão da Literatura

Sistemas pervasivos somente agora começam a ser modelados. A comunicação é um dos aspectos que deve ser tratado, e esta tem sido abordada, principalmente, no nível de rede, com os protocolos de rede sem fio, e na área de acesso e gerenciamento de dados. Neste capítulo, faz-se uma revisão dos principais protocolos de redes sem fio e dos principais resultados e trabalhos que influenciaram as soluções de modelagem da comunicação entre componentes residentes em dispositivos móveis e estáticos. Como o tema é recente, não existem trabalhos relacionados diretamente à modelagem de comunicação entre componentes do sistema pervasivo. Desta forma, buscou-se situar o problema e relacionar os trabalhos mais influentes em modelos de comunicação móvel, como o Sistema de Arquivos Coda e o Projeto Lime.

2.1 Tecnologias de Redes sem Fio

Atualmente, pode-se encontrar WLAN (*Wireless* LAN) em casas, escritórios, fábricas, hotéis e centros de convenção, além do constante aumento de seu uso em aeroportos e lojas. Os *access points* (pontos de conexão para as redes sem fio) têm sido utilizados na conexão de todos os tipos de equipamentos móveis, tais como: notebooks, computadores de mão e telefones.

As redes sem fio estão sendo utilizadas em diversas aéreas e aplicações. Através dos esforços de padronização do IEEE (*Institute of Electrical and Electronis Engineers*), dos esforços de certificação da WECA (*Wireless Ethernet Compatibility*

Alliance) e das facilidades que oferecem, as redes sem fio estão deixando de ser uma alternativa para se tornarem a principal opção de conexão para todos os usuários de rede onde o cabeamento estruturado se torna inviável.

As redes sem fio podem ser divididas em infra-estruturadas, que usam o protocolo IEEE 802.11 para se conectar à rede fixa existente (Internet), ou *ad-hoc*, que usam o protocolo Bluetooth e formam uma rede espontânea, criada à medida que os dispositivos entram no raio de acesso de outros equipamentos da vizinhança.

2.1.1 IEEE 802.11

Atualmente, o protocolo IEEE 802.11 (http://grouper.ieee.org/groups/802/11/) é o mais utilizado em redes locais sem fio e opera na banda de freqüência de 2,4 GHz. Tornou-se padrão nos últimos anos seguindo as especificações 802.11a, 802.11b e 802.11g.

Uma rede local sem fio IEEE 802.11 é baseada em uma arquitetura celular onde o sistema é subdividido em células e cada célula é controlada por uma estação-base (chamada ponto de acesso). Mesmo que as LANs sem fio possam ser formadas por uma única célula, com apenas um ponto de acesso, a maioria das instalações são formadas por várias células, de forma que os pontos de acesso estejam conectados através de uma espécie de *backbone*, tipicamente *Ethernet*, e em alguns casos, com conexão sem fio.

2.1.2 Bluetooth

Formado em fevereiro de 1998 pelas empresas Ericsson, IBM, Intel, Nokia e Toshiba, o *Bluetooth Special Interest Group* (www.bluetooth.org) provê um mecanismo que permite que dispositivos móveis sejam interligados através de enlaces de rádio de baixa freqüência. Desta forma essa tecnologia tenta responder às necessidades de conectividade das redes sem fio em três áreas: acesso a dados e voz, substituição de cabos e redes *ad hoc*. A primeira especificação do protocolo foi publicada em 1999. Aparelhos com esta tecnologia têm sido fabricados por várias empresas.

Pela especificação, os enlaces de comunicação podem ter alcance de 10cm à 10m, sendo que este limite pode ser estendido até 100m aumentando-se a potência de transmissão. A arquitetura opera na faixa de freqüência de 2.4GHz e permite taxas de transferência de até 1Mbps. Para funcionar globalmente, necessita de uma faixa de rádio livre de tributação e aberta a qualquer rádio. Aparelhos como telefones sem fio e fornos de microondas que operam na mesma faixa de freqüência podem causar interferências.

Existem ainda muitos desafios a serem superados por esta tecnologia como o tratamento das interferências, proteção e segurança, o consumo de energia e a disponibilização de aplicações. Além disso, existe a dificuldade do processo de integração com pilhas de protocolos, como TCP/IP.

2.2 A Comunicação na Computação Pervasiva

Com a popularização dos dispositivos eletrônicos e do acesso à informação em escala global, tem-se notado um grande aumento na variedade e no número desses dispositivos. Esse aumento está causando uma descentralização na estrutura das redes. A tendência é que os pontos finais de comunicação não sejam somente *desktops*, mas uma gama de dispositivos com tamanho e funcionalidades variadas que se comunicam através de redes cabeadas e sem fio.

A dispersão dos elementos computacionais e a possibilidade de acesso de qualquer lugar ou dispositivo faz com que a comunicação seja centrada no usuário. O ser humano passa a ser o ponto de partida no projeto dos serviços. Assim como a comunicação humana é caracterizada por interações com um conjunto de objetos em seu ambiente, os sistemas de comunicação não devem ser construídos baseados em tecnologias específicas, mas na análise do espaço de comunicação individual. O resultado é um sistema de comunicação que se adapta às demandas específicas de cada indivíduo (*I-centric*) [ZEL 2004].

A comunicação *I-centric* deve agir em função das necessidades do usuário e, para isso, requer um alto grau de pró-atividade e auto adaptação dos sistemas de

suporte a ela. Além disso, deve também oferecer características como consciência do ambiente, personalização e adaptabilidade à situação.

No momento atual da tecnologia está se iniciando a construção de um ambiente computacional pervasivo, onde a computação está inserida em todo o lugar e é, preferencialmente, invisível ao usuário-final. Os elementos computacionais embutidos no ambiente terão um alto grau de interação para atender as preferências e atividades de cada usuário. Isto requer repensar o modelo de comunicação entre os componentes das aplicações.

2.2.1 As limitações impostas pelo ambiente móvel atual

O ambiente móvel atual tem limitações inerentes ao meio em que opera. A largura de banda efetiva das Redes Sem Fio é somente uma fração da disponível nas redes cabeadas, que variam de 10 a 100Mbps - rede Ethernet, e de 622Mbps a gigabits (previsões) - rede ATM. Comparada a elas, a rede sem fio é limitada. Embora a velocidade de acesso máxima hoje seja 108Mbps para as redes sem fio, o padrão mais comum é o acesso a 11Mbps. Portanto, a utilização eficiente da banda é de vital importância para a aplicação, pois existirão redes com diferentes capacidades e as aplicações devem executar em todos esses ambientes de forma a atingir a expectativa do usuário-final.

Outro fator a considerar é a velocidade do processamento dos dispositivos móveis. Quanto maior a latência que pode ser tolerada no processamento, menor será o consumo de energia. Assim, velocidade de processamento, custo de armazenamento (em termos de energia), quantidade de dados transmitidos e recebidos e a latência tolerada são os fatores a serem considerados nos vários aspectos do acesso e organização dos dados para comunicação.

Logo, as soluções eficientes no consumo de energia são importantes neste ambiente por que:

- tornam possível o uso de baterias pequenas e com menor capacidade para executar o mesmo conjunto de aplicações. Baterias menores são importantes

do ponto de vista da portabilidade, tornando a unidade móvel mais compacta e leve;

- o cliente pode trabalhar por um tempo maior sem o problema da troca de bateria, o que conduz a uma economia financeira e à diminuição no impacto ambiental.

Assim, novos modelos de organização e acesso aos dados e de aplicações devem ser projetados a partir das considerações acima. Para tal, um dos principais problemas a ser tratado vem da variabilidade da conectividade à rede, que é inerente ao movimento do usuário.

2.2.2 Desconexões

As desconexões no ambiente móvel têm várias causas: voluntária, falta de energia, mudança de área de cobertura (*handoff*). Os terminais móveis são freqüentemente desconectados da rede, como uma forma de economizar energia. As unidades móveis podem se desconectar de uma determinada rede caso não concordem com o nível de serviço disponível ou queiram economizar energia. A preocupação com a desconexão faz parte do suporte à computação móvel, por isso mecanismos para tratar *handoffs*, recuperar informações e consistência de *caching* são necessários.

Redes sem fio também estão sujeitas a interferências de outras ondas eletromagnéticas e barreiras físicas como paredes ou prédios. Essas alterações variam no tempo e espaço, sendo na maioria das vezes imprevisíveis. Esta característica gera a necessidade de utilização de estratégias preventivas à desconexão, como *prefetching*, e de tolerância a falhas.

A principal distinção entre desconexão e falha é sua natureza eletiva, pois estas podem ser tratadas como falhas planejadas - as quais podem ser antecipadas e preparadas. Durante o tempo de execução de uma aplicação, ela pode experimentar vários graus de conectividade, desde a desconexão total até a conexão forte. Previsões futuras indicam que o acesso a sistemas de alto desempenho, redes confiáveis e forte conectividade serão limitadas a poucos lugares, tais como trabalho ou casa, e

o restante serão definidos por uma fraca conectividade.

Como visto, existem vários graus de conectividade [MUM 95] :

- desconexão - o cliente não tem conexão com a rede;
- conexão fraca - conexão sobre um canal sem fio com largura de banda restrita; e
- conexão forte - conexão sobre uma rede fixa, rápida e confiável.

Assume-se que o cliente móvel experimenta cada um desses tipos de conexão ao longo de um período de tempo. Entretanto, considera-se que somente um tipo de conexão será experimentado durante uma mesma sessão. Essas conexões variantes determinam os modos de operação dos sistemas, que devem dar suporte para as aplicações continuarem operando, apesar da variabilidade da conectividade à rede e sua conseqüência na disponibilidade e acesso aos dados.

Nas próximas seções são descritos trabalhos pioneiros que envolvem a operação desconectada.

2.3 Operação Desconectada no Sistema de Arquivos Coda

Segundo [MUM 95], operação desconectada é um modo de operação onde o cliente continua a usar os dados da sua cache durante falhas temporárias da rede ou do servidor. A habilidade do funcionamento desconectado também é útil mesmo quando a conectividade é disponível, para, por exemplo, economizar bateria dos dispositivos móveis.

Porém, quando desconectado, o cliente sofre várias limitações:

- Falhas de cache podem impedir o progresso de aplicações;
- Atualizações em dados compartilhados não são visíveis a outros clientes;
- Maior propensão a conflitos de atualização;

- Exaustão do espaço da cache é uma possibilidade.

No Coda [KIS 92, MUM 95] - sistema de arquivos móveis que propôs e aprofundou o estudo da desconexão planejada - um gerente de cache opera em três estados: (i) *hoarding*[1], modo conectado, onde o gerente guarda dados úteis em antecipação à desconexão; (ii) emulação, estado em que o cliente está desconectado fisicamente; e (iii) reintegração, quando reconectado, sincroniza sua cache. A figura 2.1 ilustra os estados e as transições entre eles.

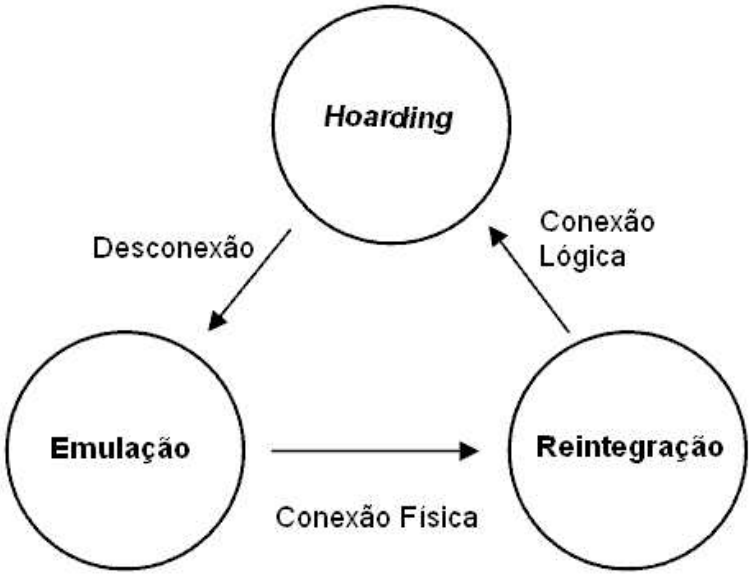

Figura 2.1: Estados do gerente de cache no Coda [KIS 92]

Para contornar o problema da imprevisibilidade das desconexões, o Coda realiza, de tempos em tempos, uma atualização na sua *cache* (*hoard walking*) verificando se os arquivos que interessam às aplicações (*hoard profiles*, providos pelos usuários) devem ser retidos na *cache* ou não, a fim de prevenir uma possível desconexão.

2.4 Espaço de objetos - resultados do projeto LIME

A comunicação entre componentes da aplicação que estejam espalhados por várias máquinas, as quais podem estar desconectadas momentaneamente, requer

[1]Uma tradução livre seria "Estado de acumulação". Outros autores também não traduzem para o português.

um novo modelo baseado num conceito de desacoplamento espacial e temporal entre emissor e receptor e com comunicação oportunista. Desacoplado no sentido que a comunicação acontece em face a desconexões, e oportunista porque explora a conectividade se esta torna-se disponível. Um dos modelos com esta característica de assincronismo é o espaço de tuplas. O espaço de tuplas é uma memória associativa disponibilizada de forma global, utilizada pelos processos para se comunicarem [GEL 85].

No modelo LIME (*Linda in a Mobile Environment*) [MUR 2001], agentes móveis são programas que podem viajar entre hosts móveis e toda comunicação é via um Espaço de Tuplas Transiente, que incorpora o conceito de mobilidade (física e lógica). O Espaço de Tuplas é permanentemente associado aos agentes móveis nos hosts móveis. O compartilhamento transiente permite a reconfiguração dinâmica de seu conteúdo de acordo com a migração do agente ou variações na conectividade. Cada agente móvel tem acesso à Interface do Espaço de Tuplas (IET) que é permanentemente associada com o agente e transferida conforme este se movimenta. Cada IET contém informações que deseja compartilhar com outros agentes, e é acessada com operações tradicionais: leitura com retirada, leitura sem retirada e escrita.

2.5 Detecção do Estado da Conexão

Em sistemas distribuídos assíncronos, como o da arquitetura do sistema pBuy, não existe limite para o tempo de resposta de uma mensagem ou um tempo determinado para se executar algum passo. Devido a isto, não se pode saber *a priori* se a mensagem está somente levando muito tempo para alcançar seu destino ou se ocorreu uma falha [FIS 85]. Para circunscrever essa impossibilidade, alguns detectores de defeitos baseados em consenso foram propostos, os chamados *detectores de defeito não confiáveis* [CHA 96].

Além de poderem não estar falhos nem lentos, processos em dispositivos móveis podem não se encontrar conectados por terem se movido para fora da área de cobertura. Detectores de conexão são baseados em gerentes de conexão. A idéia é

monitorar os recursos da rede para **prever** a desconexão e notificar as aplicações e os outros dispositivos do sistema distribuído. Esta funcionalidade é complexa por natureza. Para torná-la viável, espera-se que os fabricantes de hardware e software disponibilizem uma chamada de sistema que forneça a largura de banda real ou a taxa de ruído no sinal [CON 2002].

Capítulo 3

O Projeto pBuy

O projeto pBuy, financiado pela FINEP, com período de execução de feveveiro/2005 a dezembro/2006, está sendo desenvolvido pelo grupo de pesquisa GMob (Grupo de Pesquisa em Sistemas de Computação Móvel - UFSM). O desafio é inserir tecnologia móvel com acesso pervasivo em um sistema legado, chamado Portal de Compras, desenvolvido pela empresa SIG Soluções em Informática e Gestão (www.sigbrasil.com.br). Grande parte das aplicações existentes na área da computação pervasiva é experimental e interna aos grupos de pesquisa. O projeto pBuy é uma oportunidade de avaliar o impacto que essa nova tendência tecnológica traz ao mercado de produção de software.

O software Portal de Compras tem como base um sistema de leilão virtual para a realização de compras. O leilão é dividido em fases, delimitadas por datas ou ações da parte que realiza o leilão, e cada fase gera tanto mensagens do usuário para o sistema quanto mensagens do sistema para o usuário, sendo que a última deve ser entregue onde o usuário estiver e no dispositivo em uso no momento por ele [PIR 2005]. Essas características geram a necessidade de um serviço de envio de mensagens (Serviço de Disseminação) para o usuário autenticado pelo sistema (Serviço de Autenticação) de forma uniforme e independente do dispositivo que será utilizado. O sistema deve estar ciente do dispositivo do usuário no momento do envio para fazer a adaptação do conteúdo (Serviço de Apresentação) de forma que o dispositivo aceite e disponibilize para o usuário, levando em consideração o estado da rede e o tipo de conexão através do **Serviço de Comunicação**. A figura 3.1

mostra a integração dos serviços para o sistema proposto [PIR 2005] e a divisão entre os serviços do lado servidor e cliente.

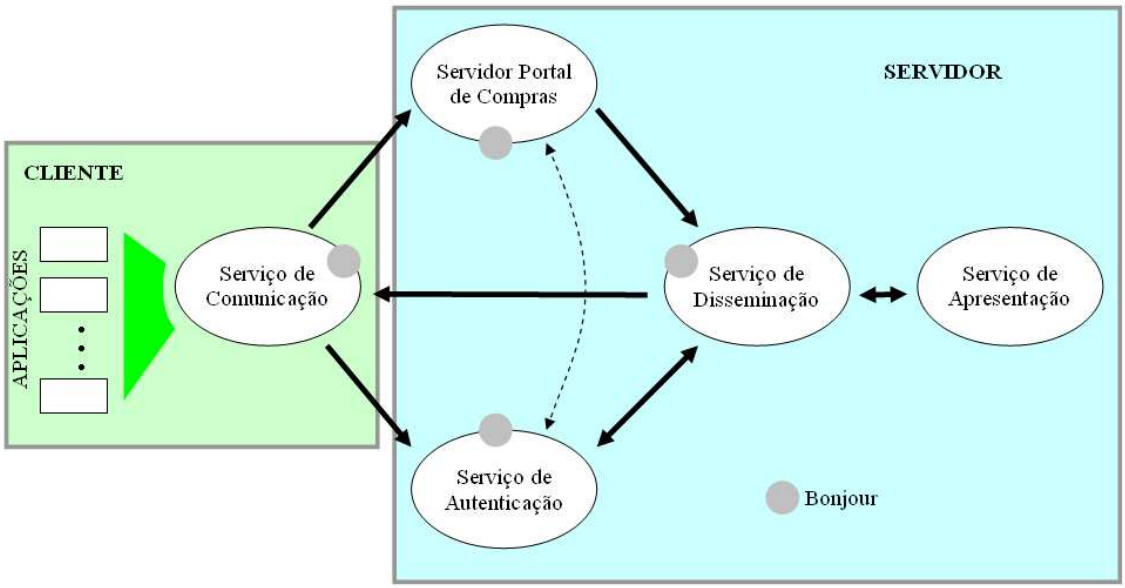

Figura 3.1: Relação entre os serviços da arquitetura proposta no projeto pBuy

A figura 3.2 ilustra, através do diagrama de seqüência, as trocas de mensagens entre os serviços e o fluxo de execução do atendimento à requisição da página inicial do portal, após a autenticação. Ressalta-se que no diagrama é considerado que o dispositivo está acessível durante todo o processo. Inicialmente, quando o Serviço de Comunicação for ativado no cliente, ele descobrirá, através do Bonjour (ver seção 4.1.1), os Serviços de Autenticação, Disseminação e o Servidor do Portal de Compras. O usuário deverá, então, autenticar-se no sistema e, com isso, receber o *ticket* que confirma o sucesso na autenticação (1). Em seguida, é solicitada ao Servidor do pBuy, a página inicial do sistema (2). O servidor do pBuy, por sua vez, após consultar o Serviço de Autenticação para obter as informações do usuário que correspondem àquele ticket (3), envia o documento solicitado (em formato XML - *eXtensible Markup Language*), juntamente com a identificação do usuário destino ao Serviço de Disseminação (4). O Serviço de Disseminação deverá verificar, consultando o Serviço de Autenticação (5), se o usuário está conectado e autenticado. Em caso positivo, o documento XML é enviado ao Serviço de Apresentação (6),

para que seja transformado em algo visualizável pelo dispositivo do usuário. Finalmente, as informações solicitadas pelo dispositivo serão entregues pelo Serviço de Disseminação (7).

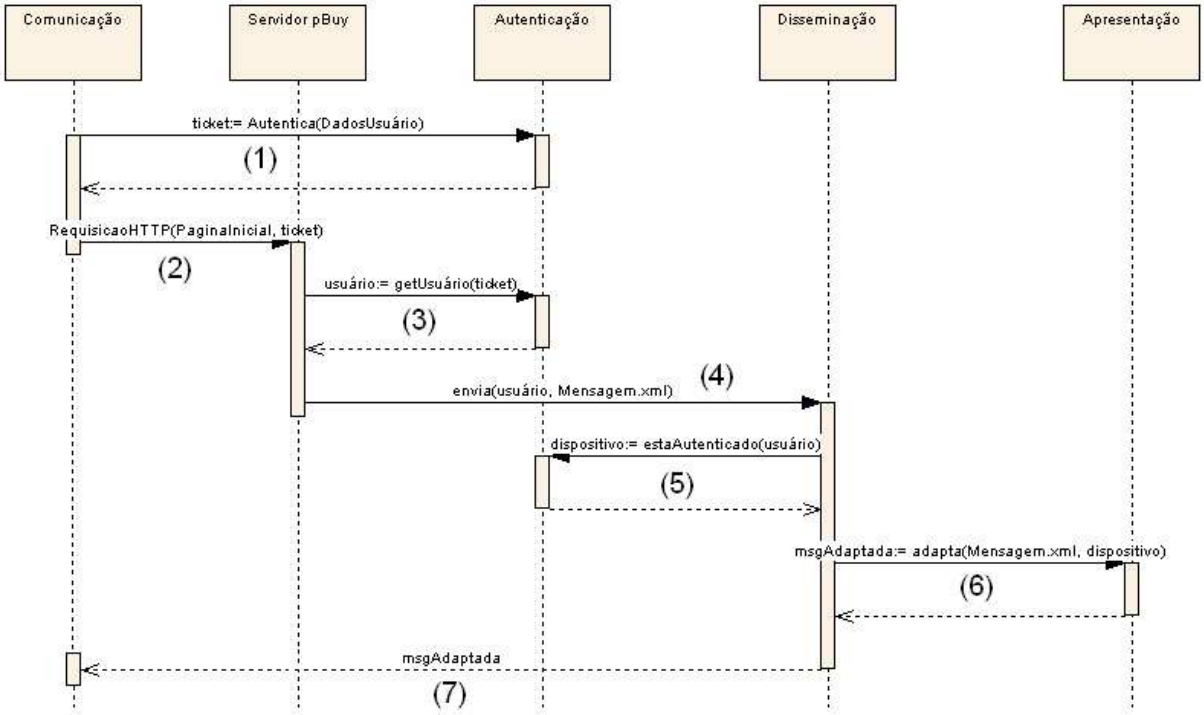

Figura 3.2: Diagrama de seqüência da troca de mensagens no sistema

A comunicação entre os serviços do lado servidor ocorre via RMI, por serem confiáveis entre si e por permitir maior abstração do que o simples envio de mensagens. Nas próximas seções, esses serviços são detalhados.

3.1 Serviço de Autenticação

Este serviço, como o nome sugere, faz a autenticação dos usuários cadastrados no sistema. Além disso, guarda informações, como o tipo de dispositivo utilizado pelo usuário na seção atual e o estado da conexão, a fim de prover essas informações aos serviços de disseminação de mensagens e apresentação de conteúdo. A informação do tipo de dispositivo é usada pelo Serviço de Apresentação que adapta os documentos a um tipo suportado por esse dispositivo. A informação sobre o estado

da conexão do usuário é usada pelo Serviço de Disseminação a fim de que ele possa optar em enviar ou guardar a mensagem para envio posterior.

As informações de autenticação, como no sistema utilizado atualmente, são: nome e tipo de usuário, e senha. O tipo de usuário determina suas permissões e pode assumir três valores: usuário comum, fornecedor e administrador.

Quando há sucesso na autenticação de um usuário, o Serviço de Autenticação gera um *ticket* que identifica aquela seção e o envia ao Serviço de Comunicação no dispositivo do usuário. Esse *ticket* é enviado ao Servidor do Portal de Compras a cada requisição, para que o usuário seja identificado pelo sistema como autenticado, por questões de segurança. Dessa forma, o Servidor do pBuy tem condições de averiguar se aquele *ticket* é válido e solicitar informações do usuário e suas permissões consultando o Serviço de Autenticação.

Para manter a informação do estado da conexão do dispositivo, o Serviço de Comunicação, residente no dispositivo, envia periodicamente mensagens com o objetivo de mostrar que continua conectado ao sistema, os *heartbeats* [AGU 97]. O não recebimento por parte do serviço de autenticação, determinado por um *timeout*, caracteriza o dispositivo como desconectado. Cada uma dessas mensagens enviadas pelo Serviço de Comunicação será respondida pelo Serviço de Autenticação para que o Serviço de Comunicação saiba que continua conectado ao sistema. Essa resposta pode ser de dois tipos: uma confirmação de que o dispositivo está autenticado, ou a não confirmação, para que o Serviço de Comunicação fique ciente que não está autenticado e reenvie os dados de identificação.

3.2 Servidor do Portal de Compras

O Servidor do Portal de Compras é o provedor das informações sobre os leilões virtuais, a comunicação com o sistema legado. É ele quem envia os documentos XML ao Serviço de Disseminação para que sejam adaptados e enviados ao dispositivo do usuário.

As requisições feitas pelo Serviço de Comunicação ao Servidor do Portal de

Compras são feitas através do protocolo HTTP (*HiperText Transfer Protocol* - RFC 2616). Ao atender uma requisição, o Servidor enviará a resposta ao Serviço de Disseminação, tendo como destinatário a identificação do **usuário** destino (nome e tipo de usuário).

Nem todas as informações enviadas ao usuário são requisitadas por ele. Algumas mensagens, como, por exemplo, a publicação de editais, podem ser inseridas no sistema e enviadas a uma classe específica de usuários. Para isso, o servidor irá manter uma base de informações de perfil dos usuários. Elas definirão os destinatários de determinados tipos de mensagens. Algumas das informações que compõem esse perfil são: tipo de usuário, atividade que exerce e que tipo de licitação estaria interessado em participar.

3.3 Serviço de Disseminação de Mensagens

Os usuários da computação pervasiva são móveis e podem explorar as capacidades de vários tipos de dispositivos. O serviço de disseminação de dados [RED 2005] deve ser transparente para quem o utiliza, somente o usuário destino e a mensagem propriamente dita devem ser suficientes para a sua entrega ao dispositivo correto. Assim, o serviço de disseminação deve identificar onde o usuário destino está, qual equipamento está utilizando no momento e enviar a mensagem de forma adaptada ao dispositivo [RED 2005a].

Na Arquitetura de Serviços pBuy, este serviço interage com os outros serviços de suporte às aplicações com comportamento pervasivo; em especial, com o Serviço de Apresentação, o qual realiza a adaptação do conteúdo ao dispositivo utilizado no momento pelo usuário que receberá a mensagem, e o Serviço de Autenticação, para confirmar que o destinatário esteja conectado e autenticado no sistema.

No momento em que receber uma mensagem do Servidor do pBuy, o Serviço de Disseminação consultará o Serviço de Autenticação para saber se o usuário destino encontra-se acessível e autenticado. Em caso positivo, o Serviço de Disseminação envia a mensagem ao Serviço de Apresentação para adaptar a mensagem ao dis-

positivo do usuário para que ela possa, então, ser entregue a ele. Caso o usuário não esteja acessível, a mensagem, não adaptada, é armazena em *cache* para envio posterior se não for do tipo 'imediata'.

As mensagens vindas do Servidor do pBuy terão um tempo de vida de acordo com sua relevância. Esse tempo de vida determina o tempo em que as mensagens permanecerão na *cache* do Serviço de Disseminação. Por exemplo, mensagens de confirmações de operações ou de erros de validação de dados não necessitam de um tempo de vida muito longo. O usuário não estaria interessado nessas informações depois de ter passado muito tempo da realização das operações. Já mensagens como, por exemplo, a abertura de editais, têm um tempo de vida maior, pois os usuários que a recebem, provavelmente, estarão interessados em participar da licitação. Esta característica gera a necessidade do armazenamento persistente de algumas mensagens.

3.4 Serviço de Apresentação de Conteúdo

A adaptação de conteúdo permite que informações sejam mostradas corretamente em diversos ambientes. Como a computação pervasiva tem como característica o acesso de qualquer dispositivo, existe a necessidade deste serviço. Na arquitetura de serviços pBuy, o Serviço de Apresentação [BEL 2005] é utilizado pelo Serviço de Disseminação de informações que informa o tipo do dispositivo e os dados que sofrerão o processo de adaptação automática. Os dados adaptados ao dispositivo devem retornar ao Serviço de Disseminação para envio na forma de mensagem.

O serviço de apresentação deve considerar a heterogeneidade dos dispositivos portáteis e celulares existentes. Como não foram encontradas soluções com essas características relativas a sistemas móveis, apenas relativas à web, tornou-se necessário desenvolver uma solução própria para atender aos requisitos funcionais do serviço [BEL 2005a].

Como as aplicações do projeto pBuy têm como interface um navegador web e as requisições ao Servidor do pBuy devem ser redirecionadas ao Serviço de Comuni-

cação, o Serviço de Apresentação é o responsável por gerar documentos html onde os *hiperlinks* sejam direcionados ao servidor HTTP do Serviço de Comunicação, no dispositivo do cliente.

Capítulo 4

Serviço de Comunicação Consciente do Estado da Rede

O Serviço de Comunicação da arquietura de serviços pBuy é o foco deste trabalho. As próximas seções descrevem, respectivamente, as tecnologias utilizadas na implementação da solução e a descrição da estrutura do serviço.

4.1 Tecnologias Utilizadas

4.1.1 Bonjour

O Bonjour [BON 2005] é uma ferramenta desenvolvida pela Apple que permite criar uma rede instantânea de computadores e outros dispositivos sem a necessidade de predefinir endereços IP ou fazer configurações em servidores DNS. Ele permite que os serviços e as capacidades de cada dispositivo sejam registrados na rede, e que esses serviços sejam descobertos dinamicamente pelos outros dispositivos da rede. O Bonjour funciona sob as tecnologias de conexão mais populares do mercado, incluindo Ethernet e IEEE 802.11 (ver seção 2.1.1) e sobre o padronizado e amplamente utilizado protocolo IP (*Internet Protocol*).

Segundo a Apple, o Bonjour veio para suprir um vazio existente na forma como os dispositivos de uma rede local se comunicam. O benefício que o Bonjour traz à computação pervasiva está na facilidade de identificação de dispositivos e serviços de forma simples e descentralizada.

A utilização do Bonjour é direta. É disponibilizado pela Apple, os códigos-

fonte necessários para seu funcionamento. É suficiente adicioná-los e integrá-los com o restante do código.

4.1.2 J2ME

O J2ME (Java Micro Edition) [J2M 2005] provê um ambiente robusto e flexível para desenvolver e executar aplicações em dispositivos como telefones celulares, PDAs (*Personal Digital Assistants*) e dispositivos embarcados. Como o J2EE (*Enterprise Edition*) e o J2SE (*Standard Edition*), o J2ME é composto por uma máquina virtual Java e um conjunto de APIs (*Application Programming Interface*) padrão.

A arquitetura do J2ME engloba uma variedade de configurações, perfis e pacotes opcionais que os desenvolvedores podem combinar para construir o ambiente de execução Java adequado a determinados dispositivos e requisitos. Cada combinação é otimizada para uma categoria de tamanho de memória, poder de processamento e capacidade de entrada e saída.

Configurações abrangem uma máquina virtual e um conjunto mínimo de classes de biblioteca. Elas provêem as funcionalidades básicas para uma gama de dispositivos que compartilham características similares, como conectividade e capacidade de memória. Atualmente, existem duas configurações J2ME: CLDC (*Connected Limited Device Configuration*) e CDC (*Connected Device Configuration*).

A fim de prover um ambiente de execução completo para uma categoria específica de dispositivos, uma configuração deve ser combinada com um perfil - um conjunto de APIs de alto nível que melhor define o acesso às propriedades específicas dos dispositivos. Um perfil dá suporte a um subconjunto de dispositivos de uma determinada configuração. Um exemplo amplamente adotado é a combinação de CLDC com MIDP (*Mobile Information Device Profile*) que provê um ambiente de execução para telefones celulares e outros dispositivos com capacidade similares.

A plataforma J2ME pode ser extendida através da adição de pacotes opcionais à pilha de tecnologias que inclui tando CLDC ou CDC. Criados para resolver diversos requisitos de aplicações, os pacotes opcionais oferecem APIs padrão para usar tecnologias existentes ou emergentes como conexão a base de dados, multimídia,

Bluetooth e *web services*.

4.2 Estrutura do Serviço de Comunicação

O principal objetivo do trabalho é fornecer a funcionalidade da operação desconectada, considerando as freqüentes desconexões inerentes ao ambiente sem fio. Para que isso seja oferecido, o serviço deve ser a ponte de comunicação entre as aplicações e o exterior ao dispositivo do usuário, detectando o estado de conexão da rede e guardando em *cache* as requisições e informações providas pelo usuário durante os momentos de desconexão [PIR 2005a].

O serviço foi implementado na linguagem Java. Neste trabalho foi usado J2ME CDC/PP (*Personal Profile*) porque é suportado pela maioria dos PDAs disponíveis atualmente e tem uma API bastante parecida com a J2SE (*Java 2 Platform, Standard Edition*) 1.3.1.

A figura 4.1 mostra uma visão geral e a divisão funcional do Serviço de Comunicação.

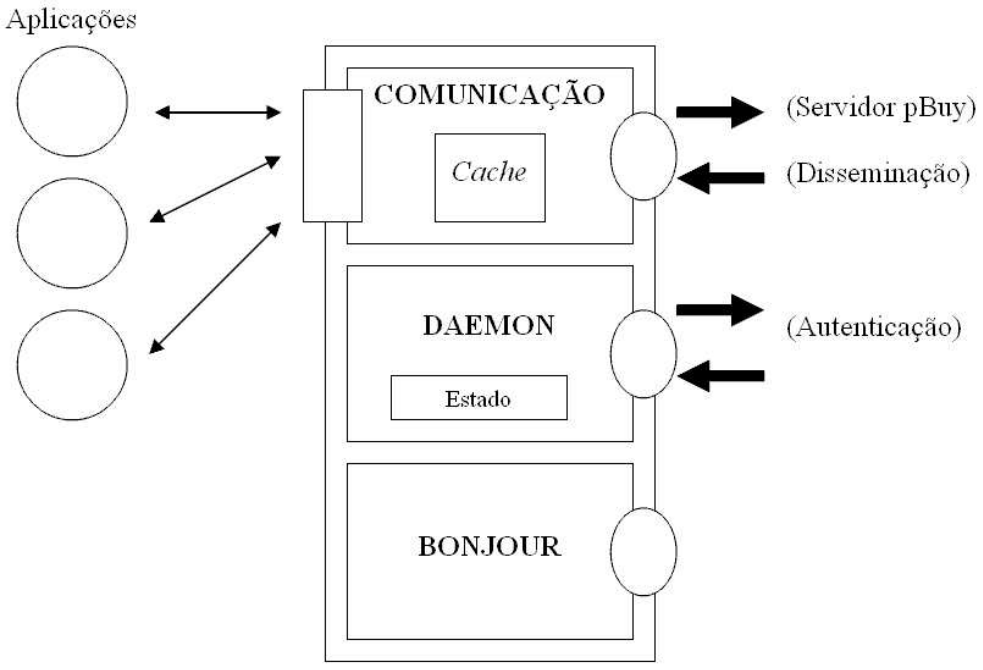

Figura 4.1: *Overview* dos módulos funcionais do Serviço de Comunicação

O módulo *Comunicação* é responsável pela gerência das portas de comunicação e do fluxo de mensagens, tanto vindas e destinadas às aplicações quanto aos outros serviços exteriores ao cliente. O módulo intitulado *Daemon* é responsável por prover e manter as informações sobre o estado da rede bem como comunicar ao módulo de *Comunicação* quando o usuário não estiver autenticado no sistema. O módulo *Bonjour* (seção 4.1.1) serve para o anúncio e descoberta de serviços na rede. É através dele que o Serviço de Comunicação econtra e identifica o endereço dos serviços com quem ele se comunica.

As próximas seções descrevem os principais pontos e o modo de funcionamento do serviço.

4.2.1 Autenticação no Sistema

Para que o usuário possa realizar operações no sistema, ele precisa identificar-se e receber o *ticket* que confirma sua autenticação (seção 3.1). Assim que o usuário prover as informações de identificação pela interface no navegador, o Serviço de Comunicação as envia ao Serviço de Autenticação, através de um socket TCP (*Transfer Control Protocol*).

A escolha da comunicação com o Serviço de Autenticação pelo protocolo TCP justifica-se pela confiabilidade e controle da conexão que ele oferece. No caso da ocorrência de problemas no envio ou **desconexões**, exceções são lançadas e podem ser tratadas, como por exemplo, com a tentativa do reenvio posterior.

Além das informações do nome e tipo de usuário e senha, também é necessário enviar ao Serviço de Autenticação uma informação sobre o tipo de dispositivo. Essa informação serve para que os documentos enviados ao usuário sejam adaptados ao dispositivo que ele porta no momento, pelo Serviço de Apresentação. O ideal, em sistemas mais amplos, seria descobrir em tempo de execução informações sobre o modelo, fabricante e arquitetura do dispositivo, para que fosse possível fazer uma adaptação bastante específica àquele aparelho. Como os sistemas operacionais não fornecem todas essas informações e as informações disponibilizadas são suficientes para os requisitos atuais do projeto pBuy, optou-se em usar as propriedades sobre o

tipo de processador (*os.arch*) e versão do sistema operacional (*os.version*) obtidas pela API Java.

Como desconexões freqüentes são previstas, o Serviço de Comunicação armazena tanto as informações de identificação quanto o *ticket* gerado com o sucesso do processo de autenticação. A detecção da reconexão ocorre quando o módulo intitulado *Daemon* volta a receber respostas do Serviço de Autenticação. Essas respostas confirmam ou não se o usuário continua autenticado. Caso a confirmação for positiva, o *ticket* da autenticação anterior à desconexão permanece válido. Caso contrário, o Serviço de Comunicação envia novamente os dados de identificação para que o usuário seja autenticado novamente.

4.2.2 Detecção do Estado de Conexão

Terminais móveis estão expostos a diversos ambientes. A mobilidade desses terminais faz com que aumente a freqüência das desconexões. Diante da impossibilidade de prever desconexões com as ferramentas disponíveis (seção 2.5), da necessidade do Serviço de Disseminação estar ciente do estado da conexão dos dispositivos portáteis para que seja postergado o envio de mensagens e da possibilidade do dispositivo continuar a operar mesmo desconectado, foi adotada uma solução simples: o Serviço de Comunicação periodicamente envia ao Serviço de Autenticação uma mensagem para dizer que está acessível e o Serviço de Autenticação responde com a confirmação ou não de que o usuário está autenticado. Com o não recebimento da mensagem por qualquer uma das partes durante um tempo pré-definido infere-se que o outro ponto de comunicação não está acessível e, portanto, o dispositivo portátil encontra-se desconectado.

A mensagem enviada pelo Serviço de Comunicação ao Serviço de Autenticação para mostrar que está acessível será, antes da autenticação ocorrer, uma mensagem sem conteúdo semântico, e simplesmente será respondida pelo Serviço de Autenticação com uma mensagem com significado semelhante a "Não estás autenticado". Depois de devidamente autenticado e ter recebido o *ticket* que confirma a autenticação, a mensagem periódica ao Serviço de Autenticação conterá esse *ticket* para que

ele saiba identificar qual usuário está enviando a mensagem. Se segundo o Serviço de Autenticação o usuário estiver autenticado ele responde com uma confirmação ao Serviço de Comunicação.

A comunicação com o Serviço de Autenticação é feita por datagramas, pelo protocolo UDP (*User Datagram Protocol*), porque não necessita estabelecer e manter uma conexão. Os datagramas são enviados pela rede sem que haja confirmação ou garantia da entrega em ordem, esta funcionalidade é suficiente para o uso pretendido. Como é dado suporte a desconexões freqüentes, o fato dos pacotes enviados não chegarem ao destino não é considerado como uma falha, mas como indicativo de que o dispositivo encontra-se desconectado.

4.2.3 Os estados de Operação

Semelhante ao sistema de arquivos Coda (ver seção 2.3), o Serviço de Comunicação opera em 3 estados: conectado, desconectado e reintegrado. As próximas seções detalham cada um deles.

4.2.3.1 Estado Conectado

O estado conectado caracteriza-se pela presença de conexão à rede. Nele, quando é feita uma requisição, ela é guardada em *cache* e é verificada a existência da conexão. Com a confirmação, é enviada ao Servidor do pBuy e excluída da *cache* (figura 4.2). A inclusão na *cache*, mesmo estando conectado, justifica-se pela possibilidade de ocorrer uma desconexão durante o envio da requisição.

Pelo mesmo motivo que na autenticação no sistema, a transferência de requisições é feita utilizando-se soquetes TCP. Desta forma, caso haja problemas durante a transferência das requisições, eles serão detectados e a requisição não será excluída da *cache*, para que possa ser retransmitida posteriormente.

4.2.3.2 Estado Desconectado

Durante o período de desconexão, cada requisição, depois de ser armazenada em *cache* e a consulta do estado atual apontar a desconexão, permanecerá guardada,

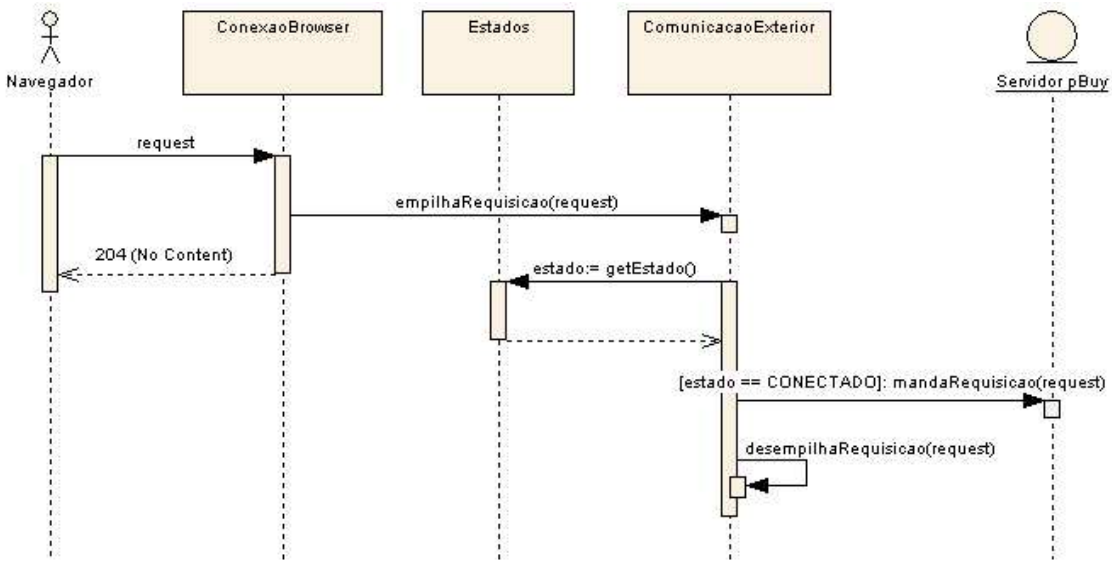

Figura 4.2: Diagrama de seqüência do atendimento à uma requisição em estado conectado

sem que haja uma tentativa imediata do seu envio. Essa tentativa ocorrerá no momento em que for detectado o restabelecimento da conexão. A estrutura de dados da *cache* é uma fila (FIFO - *First In First Out*), para que as requisições sejam enviadas na ordem em que foram solicitadas.

A maioria dos PDAs, quando a bateria está se esgotando, entra em modo de economia de energia, desligando o *display*, salvando o contexto e restaurando-o na próxima vez em que o dispositivo for ligado. Este procedimento faz com que, caso haja informações armazenadas em *cache* quando o dispositivo entrar em modo de economia de energia, elas sejam mantidas quando o aparelho for ligado novamente.

Para que as requisições e informações providas pelo usuário não sejam perdidas caso ocorra o esgotamento total da bateria ou se o dispositivo não salva o contexto, quando o serviço de comunicação está no estado desconectado, as mensagens são também armazenadas em memória persistente, e excluídas depois de terem sido enviadas no estado de reintegração. Quando o serviço for iniciado posteriormente no mesmo dispositivo e o usuário for autenticado, o serviço fará uma checagem se existem requisições não enviadas daquele usuário.

Quando o desligamento ocorrer depois de requisições terem sido enviadas, mas

antes das suas respostas serem recebidas, essas respostas ficarão armazenadas no Serviço de Disseminação para que sejam entregues quando o usuário conectar-se novamente, no disposito em que estiver. O tempo de permanência do armazenamento dessas respostas na *cache* do Serviço de Disseminação é dependente do tipo de mensagem (ver seção 3.3) [RED 2005].

4.2.3.3 Estado de Reintegração

O estado de reintegração ocorre quando é detectado que o dispositivo está conectado novamente. É o estado de transição entre os estados desconectado e conectado. No momento da detecção da reconexão, é feita a sincronização de mensagens: enviadas as eventuais requisições feitas durante o período de desconexão, que estão em *cache*, e recebidas as que estão na *cache* do Serviço de Disseminação, se houver.

Dependendo do tempo em que o usuário esteve desconectado, o Serviço de Autenticação pode considerar o usuário como não autenticado, pois poderia ter sido uma desconexão definitiva. Neste caso, o Serviço de Autenticação responderá aos *heartbeats* com uma mensagem que indique que o usuário não está autenticado e o Serviço de Comunicação enviará os dados de identificação novamente para pegar um novo ticket válido (ver seção 3.1).

4.2.4 O Acesso ao Serviço pelas Aplicações

Como no sistema em funcionamento atualmente a interface de acesso ao conteúdo do Portal de Compras é a web e os PDA's disponíveis possuem navegadores semelhantes, optou-se em manter o acesso pelos PDA's através de uma interface web. Para que isso ocorresse, o Serviço de Comunicação deveria oferecer uma interface suportada pelo navegador do dispositivo, um servidor HTTP.

Na aplicação teste, a partir do navegador, o usuário deve inicialmente digitar o endereço local para que o navegador faça uma requisição HTTP para o Serviço de Comunicação, que já deve ter sido disparado explicitamente. Ele, então, responde com a tela inicial de autenticação no sistema. Após o sucesso na autenticação, o

Serviço de Comunicação solicita ao Servidor do pBuy a página incial do sistema para que o usuário processe sua navegação.

Devido à natureza assíncrona do sistema, o Serviço de Comunicação possui algumas diferenças dos sistemas web tradicionais. Como as requisições feitas pelo usuário não são respondidas imediatamente e o dispositivo é passível de sofrer desconexões durante o atendimento às suas requisições, cada requisição feita pelo navegador é respondida com o código HTTP 204 (*No Content*) e nada muda na visualização da página (RFC 2616).

A forma encontrada para que o usuário recebesse suas requisições ou mensagens geradas pelo sistema foi a consulta periódica do navegador ao Serviço de Comunicação, utilizando *javascripts*. Para que isso fosse possível, o Serviço de Apresentação de Conteúdo teve de acrescentar esses *scripts* às páginas HTML (*HyperText Markup Language*) geradas, como forma de adaptação aos PDA's. Portanto, as respostas às requisições são armazenadas até a próxima inquirição do navegador pela existência de documentos que chegaram.

A figura 4.3 ilustra a ordem em que os eventos descritos acima acontecem. Após a digitação por parte do usuário dos dados de identificação no navegador, eles são enviados ao Serviço de Comunicação que os envia ao Serviço de Autenticação. Com o sucesso na autenticação, o serviço de comunicação faz a solicitação da página inicial ao servidor do pBuy. Enquanto ocorre o processamento dessa requisição, o navegador consulta periodicamente o serviço de comunicação sobre a chegada de mensagens. Quando o serviço de comunicação recebe a resposta do serviço de disseminação, ele a armazena até a próxima consulta do browser, quando é finalmente entregue.

Algumas páginas HTML referenciam outros arquivos, como imagens, que devem ser exibidos junto com o restante da página. Depois de receber um documento HTML com referência a outro arquivo que deve ser mostrado junto, o navegador faz uma nova requisição daquele arquivo ao servidor HTTP. Como o sistema é assíncrono e poderia haver desconexões no atendimento à segunda requisição do navegador, estourando seu tempo de espera, e o Servidor do Portal de Compras tem consciência de quando ocorre documentos desse tipo (que refeceriam outros arquivos), todos

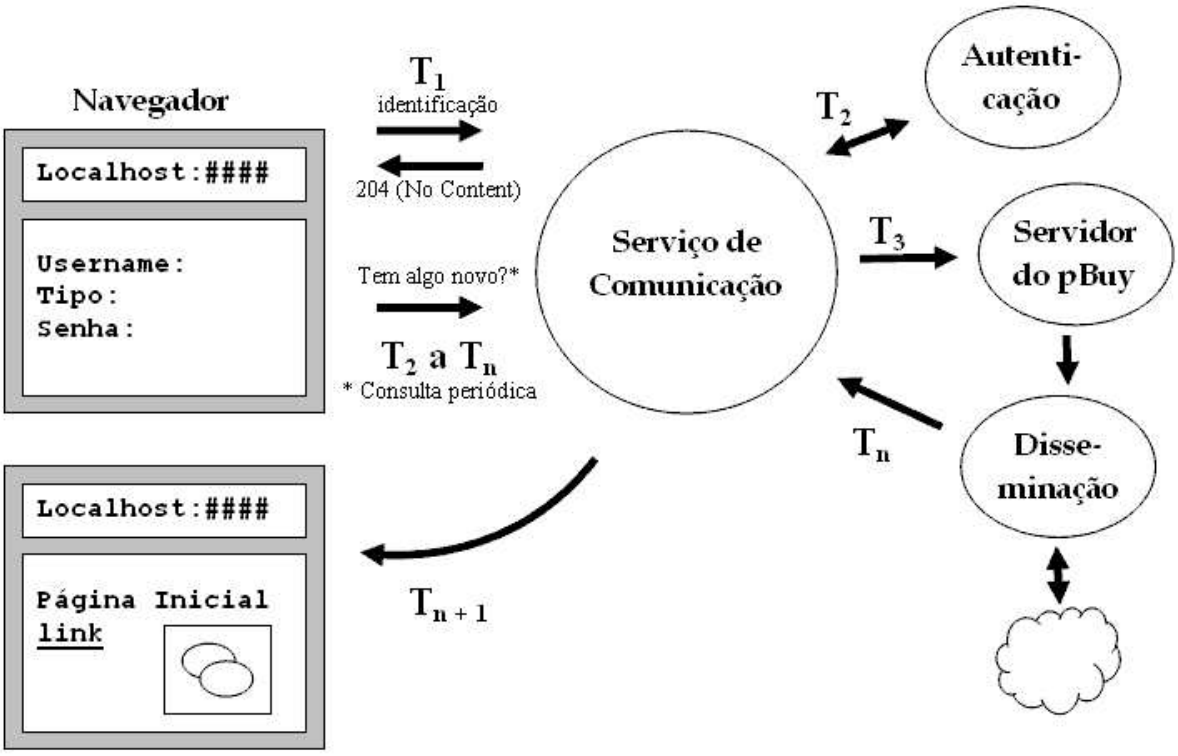

Figura 4.3: Etapas e interação entre a aplicação e o Serviço de Comunicação

os arquivos devem ser enviados na mesma resposta ao Serviço de Comunicação. Dessa maneira, quando o navegador solicitar um arquivo referenciado em uma página HTML, ele já estará disponível para ser entregue, não necessitando fazer uma nova requisição ao Servidor do pBuy.

4.2.5 Estrutura Interna

O diagrama de classes do serviço de comunicação (figura 4.4) mostra a organização interna e relação entre suas classes. As próximas seções descrevem, conforme a divisão da figura 4.1, a estrutura interna do serviço.

4.2.5.1 Daemon

Com o disparo da aplicação, pela classe `Daemon`, que contém o método `main`, são iniciadas as *threads* `ComunicacaoBrowser`, que aguarda conexões do navegador e `ComunicacaoExterior`, que aguarda conexões do serviço de Disseminação. Além

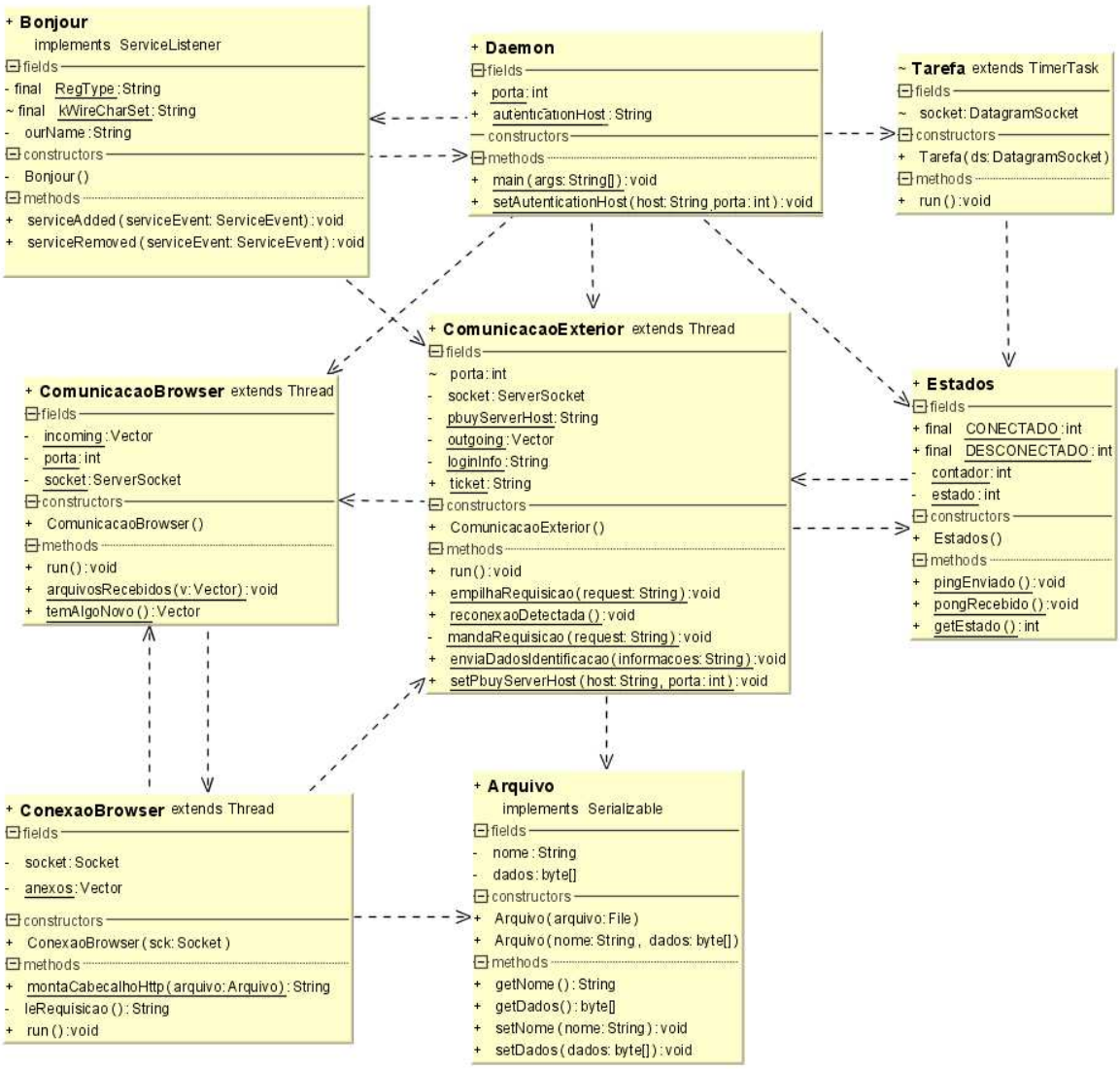

Figura 4.4: Diagrama de Classes do Serviço de Comunicação

disso, a classe `Tarefa` é agendada para executar periodicamente e a classe `Bonjour` é instanciada.

Depois de fazer a inicialização do Serviço, a classe `Daemon` passa a escutar uma porta UDP, onde virá as respostas do serviço de autenticação às mensagens periódicas enviadas pela classe `Tarefa`. A cada resposta recebida do serviço de autenticação, a classe `Daemon` chama o método `pongRecebido` da classe `Estados`. Da mesma fora, a classe `Tarefa`, que é executada periodicamente, chama o método

`pingEnviado` também da classe `Estados` quando envia a mensagem com a finalidade de mostrar que está acessível. De posse dessas informações, a classe `Estados` tem condições de manter a informação do estado do dispositivo. Se, para um determinado número de mensagens enviadas não foi obtido nenhum retorno do serviço de autenticação, o estado passa a ser `DESCONECTADO`.

Na classe `Estados`, quando o estado estava marcado como `DESCONECTADO` e o método `pongRecebido` é chamado, o estado passa a ser `CONECTADO` e a classe `ComunicacaoExterior` é comunicada, com a chamada do método `reconexaoDetectada`, para que sejam tomadas as ações necessárias com a detecção da re-conexão (seção 4.2.3.3).

4.2.5.2 Comunicação

A classe `ComunicacaoBrowser` aguarda conexões do navegador do usuário e, para cada uma delas instancia um objeto da classe `ConexaoBrowser`, que extende a classe `Thread`. A classe `ConexaoBrowser` é responsável por ler e responder às solicitações feitas através do navegador. Essas solicitações podem ser requisições de documentos, que são então enviadas para a classe `ComunicacaoExterior` através da chamada do método `empilhaRequisicao`, ou consultas para saber se chegou algo novo (ver seção 4.2.4), que são feitas à classe `ComunicacaoBrowser` através da chamada do método `temAlgoNovo`. Caso tenha algum documento na fila de entrada `incoming`, na classe `ComunicacaoBrowser` ele é enviado ao navegador pelo protocolo HTTP. A estrutura de dados é uma lista, da classe `Vector`, de objetos da classe `Arquivo`. A lista justifica-se pela possibilidade de documentos HTML referenciarem outros arquivos que devem ser mostrados junto (seção 4.2.4).

A classe `ComunicacaoExterior`, que também extende a classe `Thread`, é responsável por aguardar conexões do serviço de disseminação para o recebimento de documentos, enviar as requisições do usuário ao Servidor do pBuy quando houver conexão, consultando a classe `Estados`, ou sendo avisado por ela quando houver detecção de re-conexão, e por enviar os dados de identificação ao serviço de autenticação. É nela também onde ficam guardados os dados de identificação e o *ticket*

nos momentos de desconexão (seção 4.2.1).

Quando houver dados a serem enviados e a consulta à classe `Estados` indicar que o dispositivo está desconectado, esses dados são armazenados em *cache*, referenciada por `outgoing` e em armazenamento persistente (ver seção 4.2.3.2).

A mensagem recebida do serviço de disseminação é uma lista serializada, da classe `Vector`, de arquivos, conforme comentado acima. Esta lista é enviada à classe `ComunicacaoBrowser` para que seja entregue na próxima demanda do navegador por dados novos.

4.2.5.3 Bonjour

No construtor da classe `Bonjour`, chamado logo que o serviço é iniciado, é instanciado um objeto que controla todo anúncio e descoberta de serviços. A classe desse objeto é importada de um arquivo `.jar` com os *bytecodes* que implementam o protocolo. A classe `Bonjour` implementa a classe `ServiceListener`, que é ligada àquele objeto para que ela seja notificada quando os serviços forem descobertos e removidos.

Quando o serviço de Autenticação ou o Servidor do pBuy, serviços para o qual o serviço de comunicação envia informações, são descobertos, as classes responsáveis por esse envio, `Daemon` e `ComunicacaoExterior` respectivamente, são notificadas pelos métodos `setAutenticationHost` e `setPbuyServerHost`.

4.3 Testes e Resultados

Além dos testes locais a cada módulo e dos casos de uso do serviço, foram também realizados testes de integração com os demais serviços do sistema. Apesar do Servidor do pBuy e do Serviço de Autenticação não estarem completamente funcionais, foi possível constatar a integração e o fluxo correto dos dados entre os demais serviços.

O dispositivo portátil utilizado nos testes foi um PDA Sharp Zaurus SL-5600, com memória flash interna de 32 MB, sistema operacional linux embedix (kernel

2.4.18). Após o Serviço de Comunicação ser executado no disposito e ter descoberto os demais serviços anunciados no Bonjour (Autenticação, Servidor pBuy e Disseminação), foi aberto o navegador já instalado no sistema, Opera 6.0 [1], e digitado o endereço local do servidor HTTP interno ao Serviço de Comunicação. Ele responde com a tela inicial onde são digitados os dados de identificação e, de posse dos dados, ele envia ao Serviço de Autenticação.

Nos testes realizados, o Serviço de Autenticação era capaz de simplesmente confirmar o sucesso na autenticação para quaisquer dados e insucesso quando os campos estivessem vazios. Neste caso, a tela de identificação é mostrada novamente para reinserção dos dados, naquele, é feita uma requisição ao Servidor do pBuy pela página inicial. Assim que o Servidor recebe a requisição ele envia o documento XML ao Serviço de Disseminação, endereçado ao nome e tipo de usuário.

O nome e o tipo de usuário destino deveriam ser obtidos através de uma consulta ao Serviço de Autenticação enviando como parâmetro o *ticket* recebido do Serviço de Comunicação no momento da requisição. Como no momento em que os testes foram realizados, estas funcionalidades não estavam implementadas no Servidor do pBuy, o Serviço de Comunicação teve de enviar diretamente a ele o nome e tipo de usuário para que fossem usados como destino junto ao arquivo XML solicitado, no momento do envio ao Serviço de Disseminação.

De posse do arquivo, o Serviço de Disseminação faz uma consulta ao Serviço de Autenticação para averiguar se o dispositivo portado pelo usuário destino encontra-se acessível no momento e, se estiver, solicita também o tipo de dispositivo que ele porta. Caso o usuário esteja acessível, o Serviço de Disseminação envia ao Serviço de Apresentação o arquivo e o tipo de dispositivo para o qual ele deve ser adaptado. Depois de receber o arquivo adaptado, ele o envia ao dispositivo do usuário. A figura 4.5 mostra a tela do PDA depois de ter recebido do Serviço de Disseminação o documento adaptado pelo Serviço de Apresentação.

As situações descritas acima foram testadas entre os serviços funcionando em conjunto. Os testes foram em parte prejudicados por nem todos os serviços estarem

[1] Build 342 (de 29/01/2003), plataforma Qt/Embedded 2.3.2/ARM

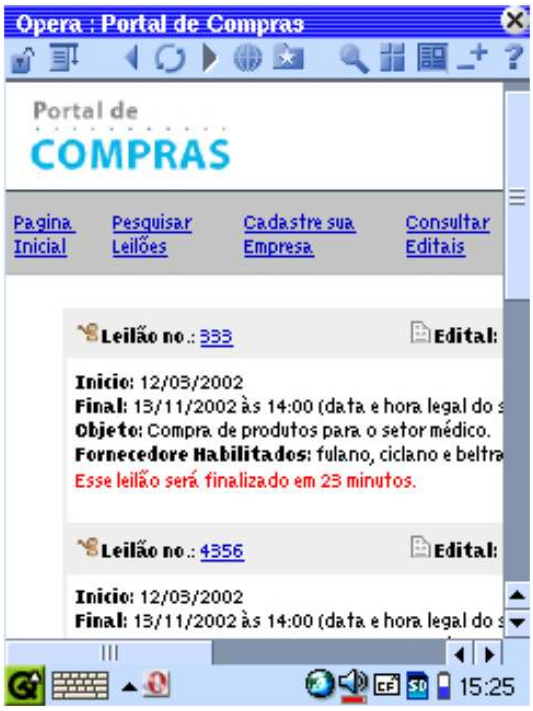

Figura 4.5: *Screenshot* da tela do PDA após receber o arquivo adaptado do Serviço de Disseminação.

completamente funcionais. De qualquer forma, pôde-se comprovar a viabilidade do modelo.

Os testes da operação em estado desconectado e posterior reintegração ocorreram de forma satisfatória. O teste de desconexão foi feito desconectando fisicamente o dispositivo da rede. Durante o tempo em que estava sem rede, foram feitas algumas requisições, que, com a volta da conexão, foram enviadas e atendidas corretamente.

Capítulo 5

Conclusão

A computação pervasiva é uma área de pesquisa promissora que envolve vários desafios. Este trabalho abordou o tratamento de alguns deles através da modelagem e implementação de um serviço de comunicação, do ponto de vista dos dispositivos portáteis, para o projeto pBuy.

A implementação do trabalho foi executada fazendo uso da plataforma Java J2ME CDC / Personal Profile, suportada pela grande maioria dos PDAs disponíveis atualmente. O principal objetivo era fornecer a funcionalidade da operação desconectada e posterior sincronização das informações, considerando desconexões constantes, situação potencialmente comum em redes sem fio, freqüentes no ambiente pervasivo. Outra característica do serviço é a descoberta dinâmica dos serviços com a qual ele se comunica, fazendo uso da ferramenta Bonjour, desenvolvida pela Apple, para o anúncio e descoberta de serviços.

Dentre as dificuldades encontradas, pode-se citar a modelagem do sistema no que diz respeito às funções e protocolos de comunicação entre os serviços. Problemas menores foram encontrados na inclusão e adaptação do código necessário para o funcionamento do Bonjour, bem como no tratamento da comunicação de baixo nível, por sockets.

A arquitetura do sistema pBuy está em fase de conclusão da prototipação do sistema. As próximas etapas contemplam a inclusão de robustez, segurança e de novas funcionalidades aos serviços.

No que diz respeito ao Serviço de Comunicação, algumas evoluções são possí-

veis: o disparo do serviço no dispositivo do usuário, os protocolos de comunicação com os outros serviços e aumento na segurança do sistema como um todo.

Uma extensão que deverá ser feita a este trabalho é a adaptação do serviço para o uso em dispositivos de menor capacidade, como telefones celulares. Dentre as considerações relevantes para realizar essa adaptação, podem-se citar: maior restrição no uso de threads, re-codificação do Bonjour ou o encontro de uma forma alternativa que exerça sua função e suporte às aplicações pelo protocolo WAP (*Wireless Application Protocol*).

O trabalho foi concluído atingindo satisfatoriamente os objetivos propostos. Suas contribuições ocorrem tanto na evolução do projeto, quanto como fonte bibliográfica no desenvolvimento de trabalhos futuros que contemplem, mesmo que tangencialmente, alguns dos apectos abordados.

Referências Bibliográficas

[AGU 97] AGUILERA, M. K.; CHEN, W.; TOUEG, S. Heartbeat: A timeout-free failure detector for quiescent reliable communication. In: LECTURE NOTES IN COMPUTER SCIENCE: DISTRIBUTED ALGORITHMS, PROC. OF 11TH INTERNATIONAL WORKSHOP, WDAG'97, 1997, Saarbrücken, Germany. **Anais...** Springer, 1997. v.1320, p.126–140.

[AUG 2004] AUGUSTIN, I. **Abstrações para uma linguagem de programação visando aplicações móveis em um ambiente de pervasive computing**. 2004. Tese de Doutorado — UFRGS - Universidade Federal do Rio Grande do Sul.

[BEL 2005a] BELUSSO, R. C. et al. Análise de alternativas para a apresentação consciente do dispositivo em um ambiente pervasivo. In: XV SEMINáRIO REGIONAL DE INFORMáTICA, 2005, Santo Ângelo - RS. **Anais...** [S.l.: s.n.], 2005.

[BEL 2005] BELUSSO, R. C. **Serviço de apresentação consciente do dispositivo em um ambiente pervasivo**. 2005. Monografia (Bacharel em Ciência da Computação) — UFSM - Universidade Federal de Santa Maria.

[BON 2005] BONJOUR. Site oficial: http://developer.apple.com/networking/bonjour/, acessado em agosto / 2005.

[CHA 96] CHANDRA, T. D.; TOUEG, S. Unreliable failure detectors for reliable distributed systems. **Journal of the ACM**, v.43, n.2, March 1996.

[CON 2002] CONAN, D. et al. Disconnected operations in mobile environments. In: IPDPS WORKSHOP ON PARALLEL AND DISTRIBUTED COMPUTING ISSUES IN WIRELESS NETWORKS AND MOBILE COMPUTING, 2., 2002, Florida (USA). **Anais...** [S.l.: s.n.], 2002.

[FIS 85] FISCHER, M. J.; LYNCH, N. A.; PATERSON, M. S. Impossibility of distributed consensus with one faulty process. **Journal of the ACM**, v.32, n.2, p.374–382, April 1985.

[GEL 85] GELERNTER, D.; CARRIERO, N. Generative communication in linda. **ACM Transactions on Programming Languages and Systems**, New York, NY, USA, v.7, n.1, p.80–112, 1985.

[IBM 2005] IBM. **Site sobre pesquisa em computação pervasiva**. http://www.research.ibm.com/thinkresearch/pervasive.shtml.

[J2M 2005] J2ME - java 2 platform, micro edition. Site oficial: http://java.sun.com/j2me/, acessado em agosto / 2005.

[KIS 92] KISTLER, J.; SATYANARAYANAN, M. Disconnected operation in the coda file system. **ACM Transactions on Computer Systems**, v.10, n.1, February 1992.

[MUM 95] MUMMERT, L. B.; EBLING, M. R.; SATYANARAYANAN, M. Exploiting weak connectivity for mobile file access. In: CM PRESS, 1995, New York, USA. **Anais...** [S.l.: s.n.], 1995.

[MUR 2001] MURPHY, A. L.; PICCO, G. P.; ROMAN, G.-C. Lime: a middleware for physical and logical mobility. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, 21., 2001. **Anais...** [S.l.: s.n.], 2001. p.524–533.

[PIR 2005] PIRES, R. P. et al. Arquitetura de serviços para o portal de compras pervasivo. In: IV SIMPóSIO DE INFORMáTICA DA REGIãO CENTRO DO RS, 2005, Santa Maria - RS. **Anais...** [S.l.: s.n.], 2005.

[PIR 2005a] PIRES, R. P. et al. Comunicação entre componentes da aplicação em ambiente pervasivo. In: XV SEMINáRIO REGIONAL DE INFORMáTICA, 2005, Santo Ângelo - RS. **Anais...** [S.l.: s.n.], 2005.

[RED 2005a] REDIN, R. M. et al. Análise de alternativas de entrega de dados independente do dispositivo em um ambiente de computação pervasiva. In: SEMINáRIO DE INFORMáTICA RS - JORNADA INTEGRADA DE TECNOLOGIA E COMPUTAçãO, 2005, Torres - RS. **Anais...** [S.l.: s.n.], 2005.

[RED 2005] REDIN, R. M. **Serviço de suporte à dissemimação de informações independente de dispositivo em um ambiente de computação pervasiva**. 2005. Monografia (Bacharel em Ciência da Computação) — UFSM - Universidade Federal de Santa Maria.

[SAH 2003] SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21^{st} century. **IEEE Computer Society**, v.36, n.3, p.25–31, Março 2003.

[SAT 2001] SATYANARAYANAN, M. Pervasive computing: vision and challenges. In: IEEE PERSONAL COMMUNICATIONS, 2001, New York. **Anais...** [S.l.: s.n.], 2001.

[WEI 91] WEISER, M. The computer of the 21^{st} century. **Scientific American**, v.265, n.9, September 1991.

[ZEL 2004] ZELETIN, R. P.; STEGLICH, S.; ARBANOWSKI, S. Pervasive communication a human-centered service architecture. **10^{th} IEEE International Workshop on future Trends of distributed Computing systems**, 2004.