



Trabalho de Graduação

SERVIÇO DE SUPORTE A DISSEMINAÇÃO
DE INFORMAÇÕES INDEPENDENTE DE
DISPOSITIVO EM UM AMBIENTE DE
COMPUTAÇÃO PERVASIVA

Ricardo Miotto Redin

Curso de Ciência da Computação

Santa Maria, RS, Brasil

2005

**SERVIÇO DE SUPORTE A DISSEMINAÇÃO
DE INFORMAÇÕES INDEPENDENTE DE
DISPOSITIVO EM UM AMBIENTE DE
COMPUTAÇÃO PERVASIVA**

por

Ricardo Miotto Redin

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 n^o 211

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 SUPORTE A DISSEMINAÇÃO
DE INFORMAÇÕES INDEPENDENTE DE
DISPOSITIVO EM UM AMBIENTE DE
COMPUTAÇÃO PERVASIVA**

elaborado por

Ricardo Miotto Redin

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. Antonio Marcos Candia

Prof. Dr. Benhur de Oliveira Stein

Santa Maria, 22 de dezembro de 2005.

Se as coisas são inatingíveis...ora! Não é motivo para não querê-las... Que tristes os caminhos, se não fora a mágica presença das estrelas.

(Mário Quintana)

Agradecimentos

É com grande satisfação que estou redigindo esta parte deste trabalho, pois, deste modo, fica aqui registrado a minha gratidão para com aqueles que me apoiaram durante a longa caminhada que trilhei até aqui.

Agradeço incondicionalmente aos meus pais Gilson e Norimar e o meu irmão Rodrigo que sempre foram a minha fortaleza, sempre tinham o conselho certo na hora certa, me ajudaram a levantar quando eu caí, vibraram quando venci, puxaram a minha orelha quando necessário e sempre estavam perto, mesmo longe. Sem vocês eu não iria conseguir nem mesmo dar o primeiro passo da jornada.

Agradeço a Iara e ao Caio que me proporcionaram uma chance no GMob, acreditaram no meu potencial, mostraram a direção a seguir, estavam sempre disponíveis, compreensíveis, flexíveis, companheiros e profissionais acima de tudo. Os conselhos, puxões de orelha e orientações foram e ainda serão de muita valia.

A todos os colegas do GMob pelos conselhos e prestatividade quando foi necessário. Em especial agradeço aos colegas Rafael e Rubens pelas tardes de trabalho no GMob e ao café filosófico das 4 no bar junto com o pessoal do LSC e outros colegas, onde certamente nasceram as idéias para muitas festas, empreendimentos absurdamente lucrativos(que nunca saíram da conversa...), programas revolucionários e o LoD!

Agradeço a Nelma pela paciência, prestatividade e o sorriso franco com que respondeu minhas dúvidas e resolveu os problemas que surgiam na matrícula, relatórios, projetos, etc.

Agradeço a Magda e o Guilherme pela amizade e por escutarem os meus de-

sabafos, anseios e dúvidas e o 'Guinho' por me agüentar esse tempo todo.

Agradeço a todos que de alguma forma contribuíram para a plena realização deste trabalho.

Por último, mas não menos importante, agradeço a DEUS.

Sumário

Lista de Tabelas	viii
Lista de Figuras	ix
Resumo	i
1 Introdução	1
2 Revisão de Literatura	4
2.1 Disseminação de Dados	4
2.1.1 Tecnologia <i>Push</i> de Entrega de Dados	4
2.1.2 Impacto da Mobilidade na Disseminação de Dados	6
2.2 Endereçamento Dinâmico	8
2.3 Trabalhos Relacionados	9
3 Arquitetura de Serviços pBuy	11
3.1 Serviço de Autenticação	13
3.2 Servidor do Portal de Compras	14
3.3 Serviço de Apresentação de Conteúdo	14
3.4 Serviço de Comunicação	15
3.5 Serviço de Disseminação de Informações	15
4 O Serviço de Disseminação do pBuy	16
4.1 Projeto e Implementação	17
4.1.1 Seqüência de Ações	17

4.1.2	A Mensagem	19
4.1.3	RMIManager	21
4.1.4	ControlManager	22
4.2	Adaptação do Serviço de Disseminação para o Uso do Bonjour	22
4.3	Tecnologias e Ferramentas Utilizadas	25
4.3.1	Plataforma Java e RMI	26
4.3.2	Extensible Markup Language (XML)	26
4.3.3	Bonjour	27
5	Conclusão e Trabalhos Futuros	28
	Referências Bibliográficas	30

Lista de Tabelas

4.1	Tabela de Valores e Ações Correspondentes	21
-----	---	----

Lista de Figuras

3.1	Arquitetura dos serviços	12
4.1	Diagrama de Classes do Serviço de Disseminação	18
4.2	Diagrama de Seqüência do Serviço de Disseminação	19
4.3	Arquivo XML que descreve uma mensagem	20
4.4	Diagrama de Classes com a adição BonjourManager.	23
4.5	Diagrama de Seqüência com a adição BonjourManager.	24

RESUMO

Trabalho de Graduação
Ciência da Computação
Universidade Federal de Santa Maria

SERVIÇO DE SUPORTE A DISSEMINAÇÃO DE INFORMAÇÕES INDEPENDENTE DE DISPOSITIVO EM UM AMBIENTE DE COMPUTAÇÃO PERVASIVA

AUTOR: RICARDO MIOTTO REDIN

ORIENTADOR: PROF^a. DR^a. IARA AUGUSTIN

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

A computação pervasiva cria uma visão de mobilidade global que se utiliza dos mais diversos dispositivos móveis ou estáticos que dinamicamente se conectam, reagem ao ambiente corrente e se coordenam para fornecer ao usuário as informações de uma forma transparente e realizar suas tarefas em qualquer lugar, tempo e dispositivo. Neste contexto, a disseminação de informações se torna uma forma eficiente de enviar as informações que o usuário necessita. Observa-se, porém, que os dispositivos móveis existentes no mercado não seguem um padrão de comunicação e isso dificulta o desenvolvimento de aplicações neste tipo de ambiente. Neste sentido, a contribuição deste trabalho é o projeto e implementação de um serviço de disseminação de informações que retire do desenvolvedor de aplicações pervasivas a preocupação quanto à forma de conexão com os mais diversos tipos de dispositivos e, desta forma, reduzir o tempo de desenvolvimento.

Capítulo 1

Introdução

A Computação Pervasiva [SAH 2003] [SAT 2001] é uma área relativamente nova da computação que fornece uma visão da computação futura onde o poder computacional estará disponível em qualquer lugar. Este cenário de mobilidade global emprega uma variedade de dispositivos móveis ou estáticos que dinamicamente se conectam, reagem ao ambiente corrente e se coordenam uns com os outros e com os serviços de rede para auxiliar o usuário na realização de suas tarefas. Espera-se que os dispositivos móveis descubram, dinamicamente, outros dispositivos em uma dada localização e continuem a funcionar ainda que em modo desconectado e sob condições de recursos e serviços limitados. O objetivo primário da computação pervasiva é fornecer aos usuários um acesso uniforme e imediato a informações e, transparentemente, suportar a execução de suas tarefas. A sua essência é um ambiente saturado com capacidade de computação e de comunicação integradas com as atividades do usuário [SAT 2001]. Deste modo, faz-se, no mínimo, necessário que o ente reconhecido no ambiente seja o próprio usuário ao invés de seu(s) dispositivo(s) e que as informações e ações que o usuário deseja realizar o sejam em qualquer lugar, tempo e dispositivo.

Uma aplicação pervasiva é aquela que contém algumas das características citadas acima. Para isso ela pode ou não fazer uso de uma plataforma que disponibilize as ferramentas necessárias para alcançar os objetivos da computação pervasiva. O uso desta plataforma facilita muito o desenvolvimento de tais aplicações já que vários problemas de incompatibilidade entre dispositivos são resolvidos sem a aplicação

nem mesmo necessitar saber que tais entraves existem.

A disseminação de informações é uma necessidade inerente à maioria das áreas da computação; com a computação pervasiva, esta é potencializada. O problema nesta área é o fato de que existem diversos e diferentes tipos de dispositivos em um ambiente pervasivo e, hoje, tais dispositivos não seguem um padrão de comunicação entre si, sendo que cada fabricante escolhe a forma mais adequada para implementar tal função. Outra questão que envolve a disseminação é que, muitas vezes, dentro de um ambiente pervasivo necessita-se enviar mensagens para o usuário sem que este as requisite. Isto é conhecido como tecnologia *push* de entrega de dados, ou seja, é o servidor quem decide enviar mensagens/informações para o cliente [FRA 98].

Tal questão gera sub-questões do ponto de vista de implementação, tais como:

- O usuário está conectado agora?
- Se não estiver conectado, deve-se guardar a mensagem para envio posterior?
- Se estiver conectado, como identificar o dispositivo e suas características?
- Qual o protocolo de comunicação entre os dispositivos?

Deve-se notar que na visão da computação pervasiva uma aplicação envia uma mensagem para um usuário e não para seu dispositivo e, portanto, não necessita estar consciente dos problemas que envolvem o envio de uma mensagem para um determinado dispositivo. Porém, a computação atual reconhece dispositivos e não o usuário. Isto é um problema que exige soluções que desloquem o foco do hardware para o usuário, de forma que este possa usar qualquer dispositivo que tiver a sua disposição e acessar o sistema da mesma forma [AUG 2004].

No grupo GMob (grupo de pesquisa em sistemas de computação móvel) está em desenvolvimento o projeto pBuy ¹, que introduz características de pervasividade em um sistema legado, chamado Portal de Compras, desenvolvido pela empresa SIG

¹Projeto financiado pela FINEP (Financiadora de Estudos e Projetos) com período de realização: 2005 e 2006.

Soluções em Informática e Gestão (www.sigbrasil.com.br). Dentre os estudos realizados para a modelagem das novas aplicações do Portal de Compras, identificou-se que as aplicações necessitarão de um conjunto de serviços que trate as características peculiares dos dispositivos portáteis e do impacto da mobilidade sobre o sistema.

A meta principal que o presente trabalho alcançou é a implementação de um serviço de disseminação de informações em um ambiente de computação pervasiva. Este serviço provê disseminação de informações às mais diversas aplicações e é uma parte importante do projeto pBuy. O serviço provê uma forma de abstração quanto à localização e ao dispositivo usado pelo cliente do sistema pervasivo a que se deseja enviar uma mensagem.

Deste modo, acredita-se que além de criar uma forma padronizada de envio de mensagens também se reduziu a complexidade e o tempo necessário para a implementação de uma aplicação pervasiva.

O restante deste texto está organizado da seguinte forma:

- O segundo capítulo contém a revisão de literatura, onde são registrados o conceito de disseminação de dados e os aspectos relativos ao ambiente pervasivo, o endereçamento dinâmico e os trabalhos relacionados.
- O terceiro capítulo trata do escopo do trabalho com uma descrição do projeto pBuy e a arquitetura de serviços sendo desenvolvida.
- O quarto capítulo detalha o foco deste trabalho, a modelagem e a implementação do serviço proposto com a apresentação de diagramas e tecnologias utilizadas.
- O quinto capítulo apresenta a conclusão deste trabalho e os trabalhos futuros que podem derivar deste.

Capítulo 2

Revisão de Literatura

Infelizmente não consta na literatura, até o momento, uma solução que atenda as necessidades do projeto pBuy. Deste modo, segue abaixo, uma revisão de literatura sobre os aspectos relacionados ao problema que este trabalho visa solucionar: a disseminação de dados independente de dispositivo.

2.1 Disseminação de Dados

Segundo Henricksen [HEN 2001], os usuários da computação pervasiva podem ser móveis e explorar as capacidades de vários dispositivos simultaneamente. Desta forma, torna-se necessário produzir mecanismos que possibilitem a distribuição e mobilidade do software também. Estes mecanismos devem ser amplamente transparentes para os desenvolvedores de software de modo que os mesmos não tenham que se preocupar com questões como migração de dados e código e/ou sincronização de componentes distribuídos.

2.1.1 Tecnologia *Push* de Entrega de Dados

O conceito de entrega de dados por disseminação não é novo, os primeiros trabalhos foram realizados nos anos 80. Recentemente, a disseminação de dados via *broadcast* tem atraído atenção devido ao suporte físico para *broadcast* fornecido pela infra-estrutura de rede. A combinação da tecnologia *push* com a Internet ¹,

¹Algumas vezes referenciado como *WebCasting*.

tem criado muitos produtos comerciais e tem gerado controvérsias com respeito a sua competência em resolver o problema de sobrecarga do servidor. O problema está na implementação que usa protocolos de comunicação *unicast* para enviar informações à (potencialmente) milhões de usuários. Estas implementações simplificam o trabalho de quem se utiliza delas, mas em contrapartida geram uma sobrecarga extra nos servidores para o gerenciamento da interface *push*. Assim, as soluções propostas para a mobilidade pressupõem que exista um canal de difusão para evitar esse problema.

Disseminação, também referenciada como tecnologia *push*, para entrega de dados é uma resposta, em evidência, para alguns dos problemas dos sistemas *pull* [FRA 98] em ambiente móvel. Um desses problemas é que o usuário deve conhecer *a priori* onde e quando obter os dados. Quando a volatilidade de dados é alta, os usuários devem buscar dados freqüentemente, gastando tempo e esforço. O segundo problema é o descompasso entre a natureza assimétrica de algumas aplicações e a infra-estrutura simétrica da comunicação ².

Neste contexto, podem ser identificados diferentes tipos de assimetria, cada qual tendo um impacto sobre os mecanismos de entrega de dados [FRA 98]. A primeira, e mais óbvia, é a assimetria de rede onde a largura de banda da rede é diferente entre o canal de *upstream* (do cliente para o servidor) e o canal de *downstream* (do servidor para o cliente).

O segundo tipo de assimetria, razão cliente/servidor, é exibido nos sistemas de informação distribuídos, devido ao desbalanceamento entre o número de clientes e o número de servidores. Tipicamente, o número de clientes é muito maior que o de servidores, a razão entre esta diferença causa atraso na entrega dos dados e sobrecarga nos servidores.

O terceiro tipo de assimetria, volume de dados, é resultado da diferença entre a quantidade de dados transmitidos entre servidores e clientes. Por exemplo, consulta de pesquisa tem um fluxo *upstream* de poucas palavras-chaves e o fluxo *downstream* do resultado pode envolver uma considerável quantidade de dados.

²Naturalmente, existem sistemas de comunicação assimétricos, como os baseados em satélites, mas a quantidade de comunicação de dados é ainda maior sobre as redes terrestres.

A volatilidade dos dados é a fonte do quarto tipo de assimetria. Se os dados e serviços que os clientes desejam acessar alteram-se freqüentemente (dados podem ser alterados ou serviços podem ser adicionados), existirá uma assimetria no controle do fluxo de dados, visto que o servidor normalmente controla a ação.

No ambiente que exhibe uma ou mais dessas assimetrias, a entrega de dados baseada em demanda não produz bons resultados, como no caso do ambiente móvel [FRA 98]. Uma solução é usar técnicas baseadas em disseminação de dados, de forma que os clientes não mais requisitam os dados, mas estes se tornam disponíveis quando o servidor os transmite.

Assim, o aspecto chave das aplicações baseadas em disseminação é a assimetria na comunicação. Isto é, a capacidade de comunicação ou o volume de dados do servidor para o cliente (*downstream*) é muito maior do que o do cliente para o servidor (*upstream*). Segundo Franklin [FRA 98], um sistema baseado em disseminação deve incorporar diferentes estilos de entrega de dados (*push* e *pull*) a fim de ser eficiente nos vários tipos de assimetria, permitindo também aplicações simétricas.

2.1.2 Impacto da Mobilidade na Disseminação de Dados

As questões relacionadas no item anterior também se aplicam à mobilidade. Devido às restrições de energia, as unidades móveis permanecem desligadas ou no modo de economia de energia, o que indica, por exemplo, que o escalonamento do envio das informações para os clientes deve ser regular, uma vez que as unidades devem ser *acordadas* para receber o dado relevante.

O movimento das unidades requer que o conteúdo do dado a ser difundido seja dinâmico e adaptativo. A abordagem *cache no ar* [STA 97] dinamicamente altera o conteúdo disseminado, baseando-se na freqüência de acesso aos dados. O fator mobilidade novamente complica esta questão. Como a unidade se move de célula a célula³(célula é a denominação da área de cobertura de um ponto de acesso sem-fio e de telefones celulares), as necessidades do cliente movem-se com ele. Quando o usuário se move para outra célula, este deve adicionar os dados difundidos por esta

³(

nova célula, e possivelmente descartar os anteriores. Assim, o conteúdo do dado a ser difundido deve se adaptar a este movimento.

Existem dois conjuntos de questões a serem tratadas: uma do lado do servidor e outra do lado do cliente.

O principal desafio da perspectiva do servidor é organizar e escalonar os dados a serem disseminados. A natureza serial do meio impede o acesso aleatório; portanto, a distribuição dos dados no canal de difusão é crítica para o desempenho do sistema. A difusão sem fio fornece uma nova forma de dados que pode ser vista como um outro meio de memória - *public air memory* [IMI 97]. O acesso não requer transmissão, mas uma *escuta*, onde a consulta envolve uma simples filtragem dos dados recebidos, de acordo com o filtro especificado pelo usuário. Desta forma, os dados devem ser organizados a fim de facilitar o uso do mecanismo de recuperação dos dados.

Difundir dados levanta outras questões importantes, já que o controle é transferido do usuário para o fornecedor de informações (servidor). A mais relevante é o escalonamento dos dados a serem transmitidos, de forma a atender as necessidades dos usuários (os dados transmitidos devem ser relevantes). Outro ponto é que o servidor deve alterar os dados sendo difundidos, de acordo com a mudança dos interesses dos clientes.

Do lado do cliente, os problemas são relacionados à recuperação dos dados, à substituição de páginas e ao *prefetching*. A recuperação dos dados se dá a partir do conhecimento da organização (estrutura) dos dados. Porém, o escalonamento global, realizado pelo servidor para escolher os dados a disseminar, pode não atender completamente o cliente em seu padrão de acesso local. Este problema conduz à necessidade de uma *cache* no cliente, a fim de minimizar o tempo médio de espera para acessar o dado. Existe ainda o problema de propagar as atualizações dos dados para todos os clientes. Assim, a grande quantidade de dados que devem ser gerenciados é a raiz dos benefícios e das desvantagens desta solução.

2.2 Endereçamento Dinâmico

No início da Internet os endereços dos dispositivos conectados a ela eram controlados por um arquivo de *hosts* do qual cada máquina da rede tinha uma cópia. Com o passar do tempo esse sistema de mapear nomes em endereços tornou-se extremamente suscetível à falhas e impraticável devido à baixa escalabilidade do modelo e à necessidade de manutenção contínua a curtos intervalos. Para solucionar este problema foi proposto, e implantado com sucesso, o sistema de DNS (*Domain Name System*), onde recursivamente cada domínio e sub-domínio cuidam da tradução dos nomes dos *hosts* em seus endereços correspondentes.

Infelizmente, esse sistema não contempla uma classe de dispositivos: aqueles que possuem o seu endereço atribuído dinamicamente por um servidor DHCP a cada inicialização e/ou tempo pré-determinado. Estes dispositivos somente seriam alcançáveis pelo seu endereço atual e não poderiam oferecer os seus serviços na Internet de uma forma contínua através de um nome. Desta forma, o protocolo DNS recebeu um adendo através da RFC2136 [RFC 2005] que definiu uma forma eficaz de mapear dinamicamente um nome em um endereço junto ao protocolo DNS. Este adendo ficou conhecido como DNS dinâmico.

Deve-se considerar o DNS dinâmico como um grande avanço para a dissociação do endereço físico de um dispositivo de sua função já que o mesmo permite que um nome no espaço de nomes DNS mude de dispositivo quase que automaticamente. Esta dissociação é uma característica que auxilia a computação pervasiva a alcançar os seus objetivos. Apesar disso, o DNS dinâmico possui um grande problema para computação pervasiva no que diz respeito à relativamente baixa frequência com que se pode mudar um mapeamento de um mesmo registro.

A Apple, no entanto, seguiu outro caminho, definindo uma gama de protocolos próprios de rede chamado AppleTalk [WIK 2005] em 1984. O Appletalk possuía um protocolo que era utilizado para descoberta de serviços e resolução e nomes nos dispositivos Apple. Com o tempo a Apple criou evoluções deste protocolo para que o mesmo fosse interoperável com o sistema corrente na Internet. Desta evolução tem-

se, hoje, o protocolo Bonjour [APP 2005] que contempla uma forma de descoberta de serviços e resolução de nomes em um protocolo, o qual realiza estas tarefas de uma forma robusta e transparente para o usuário. Além destas características, o protocolo Bonjour é compatível com o DNS dinâmico e, assim sendo, o torna um protocolo com capacidades para ser utilizado através da Internet, da mesma forma que ele é utilizado em uma rede lógica local.

Segundo a Apple, o Bonjour veio para suprir um vazio existente na forma como os dispositivos de uma rede local se comunicam de maneira semelhante ao utilizado na Internet e sem a necessidade de uma etapa de configuração por parte do usuário e/ou administrador.

O maior benefício que a computação pervasiva obtém do protocolo Bonjour é uma forma de livrar o desenvolvedor da preocupação com o endereço do dispositivo e deixá-lo livre para se preocupar em servir o usuário, ente que deve ser o centro das atenções em tal ambiente. Outra característica relevante do ponto de vista da computação pervasiva é a descoberta de serviços de forma simples e descentralizada.

2.3 Trabalhos Relacionados

Poucos trabalhos discutem estes problemas. Notadamente, o grupo de mobilidade da Universidade Rutgers, comandado pelo professor Imielinski [DAT 2005], tem realizado significativo trabalho desde 1993 em disseminação de dados para ambientes móveis no Laboratório DATAMAN. Alguns dos resultados de suas pesquisas foram discutidos acima e segue abaixo uma descrição de alguns dos projetos desenvolvidos pelo grupo.

- ***Digital Sprinklers*** - Tem como meta preencher o espaço local com dispositivos agregadores e dispersores de informações importantes para os usuários que passam por perto se baseando para isso em suas preferências previamente selecionadas.
- ***Dataspace*** - Trata-se de um banco de dados massivamente distribuído que agrega fortemente espaço físico com as informações produzidas. Isto é alcan-

gado fazendo com que os produtores das informações sejam obrigados a guardarem as mesmas. As requisições dos usuários por informações são respondidas perguntando para a localização física em que a informação foi produzida.

- ***Non-Interactive WEB*** - A meta do projeto é criar uma arquitetura de operações desconectadas para a WEB, provendo requisições em segundo plano, descritor uniforme de recursos, serviço de diretórios, etc.
- ***Infostations*** - O projeto tem como meta o projeto, avaliação e implementação de serviços de informação usando o conceito de estações de informação. A arquitetura consiste em uma rede *wireless* de pequena banda ubíqua que é aumentada com acessos convenientes e freqüentes a pontos isolados com banda-larga que são chamadas InfoStations.

Capítulo 3

Arquitetura de Serviços pBuy

O grupo GMob atualmente desenvolve o projeto pBuy, financiado pela FINEP, com período de execução de fev/2005 a dez/2006. Este projeto tem como objetivo a adequação de novas tecnologias ao sistema de Portal de Compras da empresa interveniente com vistas à inovação tecnológica, mercado de exportação de software e padronização segundo o processo do *Capability Maturity Model Integration* (CMMI). O aporte tecnológico da instituição executora será na área de computação pervasiva e padronização de processo de engenharia de software para alcançar certificação CMMI. A inovação tecnológica introduzida pela computação pervasiva é o diferencial para a empresa aumentar seu grau de competitividade e alcançar novos mercados.

O desafio de pesquisa para o grupo GMob é o de inserir tecnologia móvel com acesso pervasivo em uma aplicação real. Grande parte das aplicações existentes na área da computação pervasiva é experimental e interna aos grupos de pesquisa. Este projeto 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.

A arquitetura do projeto pBuy é organizada em serviços que realizam separadamente as tarefas necessárias para prover as informações requisitadas ou que devem

ser enviadas aos clientes. Para tanto, foram modelados os serviços de autenticação, disseminação e apresentação do lado do servidor. Do lado do cliente, existe o serviço de comunicação.

Entre os serviços que residem no lado servidor, a comunicação é feita por invocação remota de métodos (RMI). A comunicação entre o cliente e os serviços é feita de dois passos: (1) o cliente requisita algo relativo à aplicação envolvida; (2) a resposta lhe será enviada através do serviço de disseminação.

Por outro lado, é necessária uma forma dos clientes saberem como comunicar suas requisições para os serviços do pBuy. Para isso é usada a tecnologia Bonjour que fornece um meio de endereçamento e uma forma de anúncio e descoberta de serviços de maneira automática, sem a necessidade de uma etapa de configuração por parte do usuário do dispositivo.

A figura 3.1 mostra os passos de uma requisição do usuário:

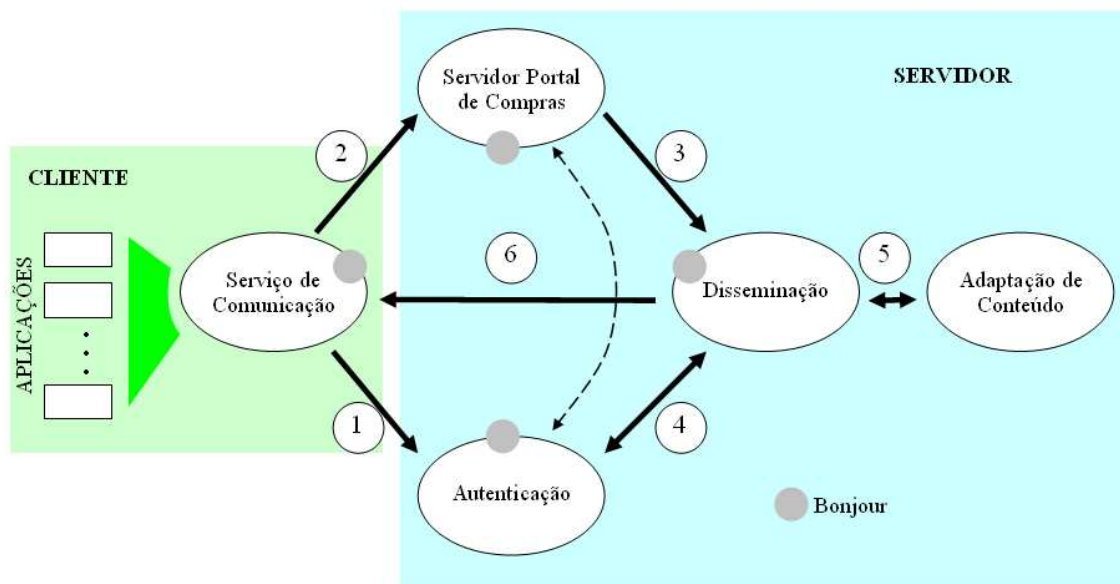


Figura 3.1: Arquitetura dos serviços

Inicialmente, quando o serviço de comunicação for ativado no cliente, ele descobrirá, através do Bonjour, os serviços de autenticação, disseminação e o servidor do portal de compras.

O usuário deverá então se autenticar no sistema através do serviço de autenti-

cação (vide seção 3.1) (1). Em seguida, a aplicação do usuário, provavelmente, fará uma solicitação a alguma informação do Portal (2), que, por sua vez, enviará o documento solicitado (em formato XML - *eXtensible Markup Language*) e informações do usuário destino ao serviço de disseminação (3).

O serviço de disseminação deverá verificar, consultando o serviço de autenticação (4), se o usuário está conectado e autenticado, no caso da informação não ser pública. Essas informações, juntamente com o documento XML, são, então, enviadas ao serviço de apresentação (5) para que sejam transformadas em algo visualizável pelo dispositivo do usuário. Finalmente, as informações solicitadas pelo dispositivo serão entregues pelo serviço de disseminação (6).

A seguir serão brevemente explicados os serviços que fazem parte do projeto.

3.1 Serviço de Autenticação

Este serviço, como o nome sugere, faz a autenticação dos usuários cadastrados no sistema. Além disso, ele guarda informações, como o 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.

Este serviço serve como um *proxy* para o módulo da aplicação legada que exige que o usuário autentique-se antes que qualquer ação possa ser executada pelo usuário. Para isso ele recebe as informações de autenticação dos usuários e confronta as mesmas com as informações contidas no módulo de autenticação do Portal de Compras. Este módulo pode atuar independentemente ou acoplado ao Portal de Compras, isto é, as informações de autenticação podem ser guardadas e verificadas internamente a ele ou confrontadas com as que constam no Portal de Compras.

Para manter a informação do estado da conexão do dispositivo, o serviço de autenticação enviará mensagens periódicas ao usuário (*heartbeats*) [AGU 97]. O não envio, por um determinado intervalo (*timeout*), caracteriza o dispositivo como desconectado.

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.

Para que as mensagens sejam entregues aos destinatários corretos o servidor do Portal de Compras envia ao serviço de disseminação um identificador único do usuário e a mensagem, além de outros dados.

3.3 Serviço de Apresentação de Conteúdo

A adaptação de conteúdo [BEL 2005] 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. No sistema pBuy, o serviço de apresentação é utilizado pelo serviço de disseminação de informações que informará o tipo do dispositivo e os dados que sofrerão o processo de adaptação automática. Os dados adaptados ao dispositivo deverão 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 as características citadas acima relativas a sistemas móveis, apenas relativas à WEB, concluiu-se que seria necessário desenvolver uma solução própria para atender aos requisitos funcionais do serviço.

Para o desenvolvimento desta solução é utilizada a tecnologia XSL, acrônimo de *eXtensible Stylesheet Language* (Linguagem de Folha de Estilos Extensível). O XSL foi desenvolvido pela *World Wide Web Consortium* (W3C) pela necessidade de uma linguagem de estilos para documentos XML (*eXtensible Markup Language*), permitindo assim a transformação de documentos XML em outros tipos de documentos. Neste caso, é armazenada uma única versão com o conteúdo a ser apresentado, enquanto diferentes formas de apresentação são geradas automaticamente a partir

deste conteúdo de acordo com o tipo de dispositivo-alvo. Para isso, o armazenamento do conteúdo é no padrão XML de forma a ser convertido no formato apropriado ao dispositivo-alvo.

Num primeiro momento, é realizada a adaptação considerando os formatos de dados: HTML, WML, PDF, XML e TXT que atendem as necessidades imediatas das aplicações pBuy. Outros formatos poderão ser tratados na seqüência de execução do projeto.

3.4 Serviço de Comunicação

O serviço de comunicação [PIR 2005] 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 para os momentos de desconexões.

3.5 Serviço de Disseminação de Informações

O serviço de disseminação é responsável por enviar as mensagens que o portal de compras gera ao usuário móvel e é executado do lado do servidor. Ele deve realizar o envio da mensagem para o usuário sem que o mesmo tenha que requisitá-la explicitamente. A mensagem deve ser entregue adaptada ao dispositivo que o usuário está utilizando para acessar o ambiente pervasivo, por isso o serviço de disseminação interage com o serviço de Apresentação que faz a tradução do conteúdo da mensagem. No próximo capítulo o serviço de disseminação será explicado em seus detalhes.

Capítulo 4

O Serviço de Disseminação do pBuy

Este capítulo apresenta os passos e decisões que foram necessários para a construção do serviço de disseminação do projeto pBuy. O serviço atua como uma forma de enviar mensagens para usuários móveis e facilitar o trabalho dos desenvolvedores de aplicações pervasivas.

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

No pBuy, este serviço interage com os outros serviços de suporte às aplicações com comportamento pervasivo; em especial, com o serviço de Apresentação, que realiza a adaptação do conteúdo ao dispositivo utilizado no momento pelo usuário que deverá receber a mensagem, e o serviço de autenticação, para confirmar que o destinatário está *online* e autenticado no sistema.

As mensagens enviadas ao serviço de disseminação são endereçadas ao usuário e não a um dispositivo ou endereço IP (*Internet Protocol*). O serviço de disseminação solicita ao serviço de apresentação a transformação da mensagem de acordo com o dispositivo do usuário, com base nos dados fornecidos pelo serviço de autenticação.

A atribuição de um endereço para o usuário independentemente do dispositivo

que ele ocupe para acessar o ambiente é uma das questões mais relevantes deste serviço, a qual permite a entrega da mensagem ao usuário independente do dispositivo em uso, no momento.

A necessidade de um padrão de endereçamento é característica do ambiente pervasivo. A forma de endereçamento ideal teria que contemplar a total mobilidade do usuário como, por exemplo, transparência na troca de endereços quando o usuário muda de ambiente. No entanto, atualmente, poucos protocolos de comunicação suportam as necessidades desse ambiente.

4.1 Projeto e Implementação

Internamente, o serviço de disseminação está dividido em módulos independentes que colaboram para realizar a tarefa que foi destinada ao serviço. Isto também facilita a programação e facilita o entendimento do que deve ocorrer em cada momento da execução do programa. A implementação foi inteiramente programada em Java [SUN 2005] na sua versão J2SE.

Na figura 4.1 podemos ver um diagrama de classes que mostra todos os módulos criados para a implementação do serviço de disseminação. Os módulos relevantes serão explicados no decorrer do texto.

Na fase de projeto ficou definido que o serviço de disseminação deveria receber uma mensagem, o nome do seu destinatário e mais algumas informações (vide seção 4.1.2) e enviá-la para o usuário sem a necessidade de que o mesmo tenha que fazer requisições para recebê-la. Fica evidente que é necessário alguma forma de monitorar quando os usuários ficam ativos. Outros meios para saber quando o usuário está ativo são possíveis (um exemplo é o caso onde o usuário poderia avisar quando se torna ativo) mas têm as suas desvantagens.

4.1.1 Seqüência de Ações

O diagrama de seqüência (Figura 4.2) ilustra os passos seguidos quando uma requisição de envio de mensagem chega ao módulo de disseminação de dados.

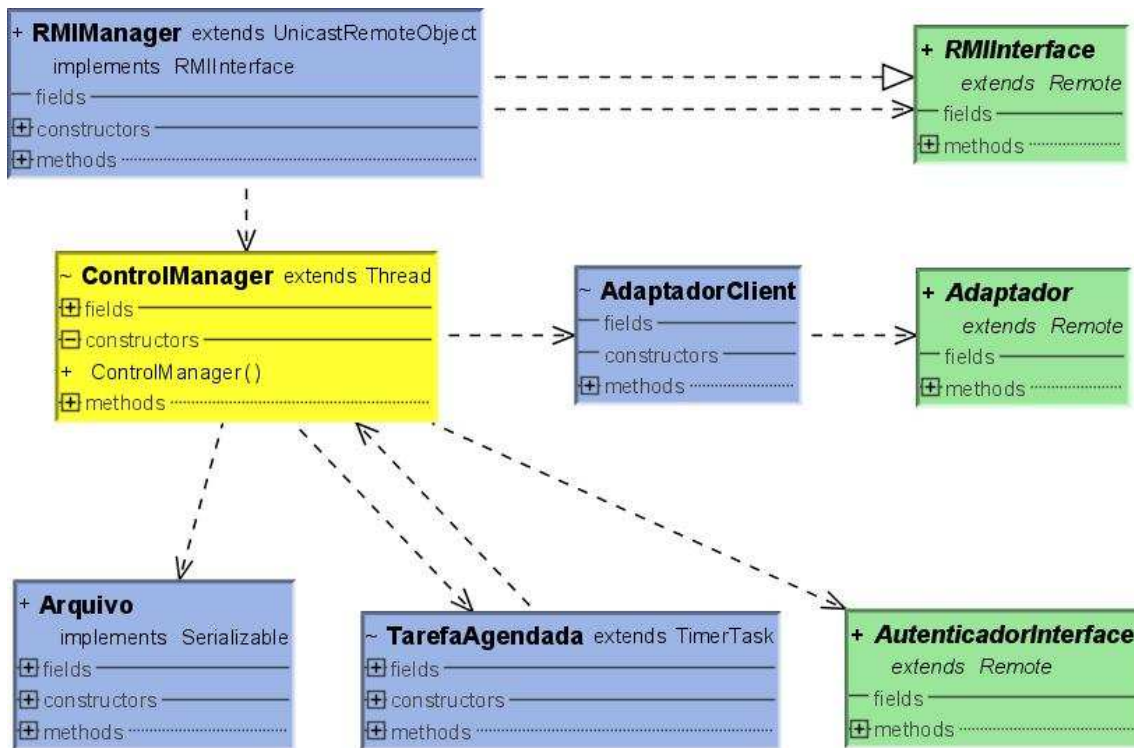


Figura 4.1: Diagrama de Classes do Serviço de Disseminação

A aplicação envia a mensagem por RMI e esta é recebida pelo `RMIManager`, que é o módulo responsável por esta tarefa. O `RMIManager`, por sua vez, valida a mensagem através de um arquivo XSD (*XML Schema Definition*) que define o formato sintaticamente correto das mensagens. Se esta for inconsistente ele lança uma exceção remota como um aviso à aplicação que enviou a mensagem de que a mesma não poderá ser enviada. No caso de uma mensagem válida o `RMIManager` instancia um objeto da classe `ControlManager` para cuidar do resto do processo de envio. O `ControlManager` pergunta então para o Serviço de Autenticação se o usuário está autenticado e qual seu dispositivo ¹. De posse desta resposta, o `ControlManager` solicita ao Serviço de Apresentação que formate a mensagem para o dispositivo em questão. Em seguida, este solicita ao Serviço de Autenticação o endereço físico do usuário-destino da informação e cria um *socket* TCP com o Serviço de Comunicação do dispositivo do usuário para a efetivação do envio. Caso

¹Cabe salientar que somente o último dispositivo no qual o cliente efetuou autenticação será mantido como ativo

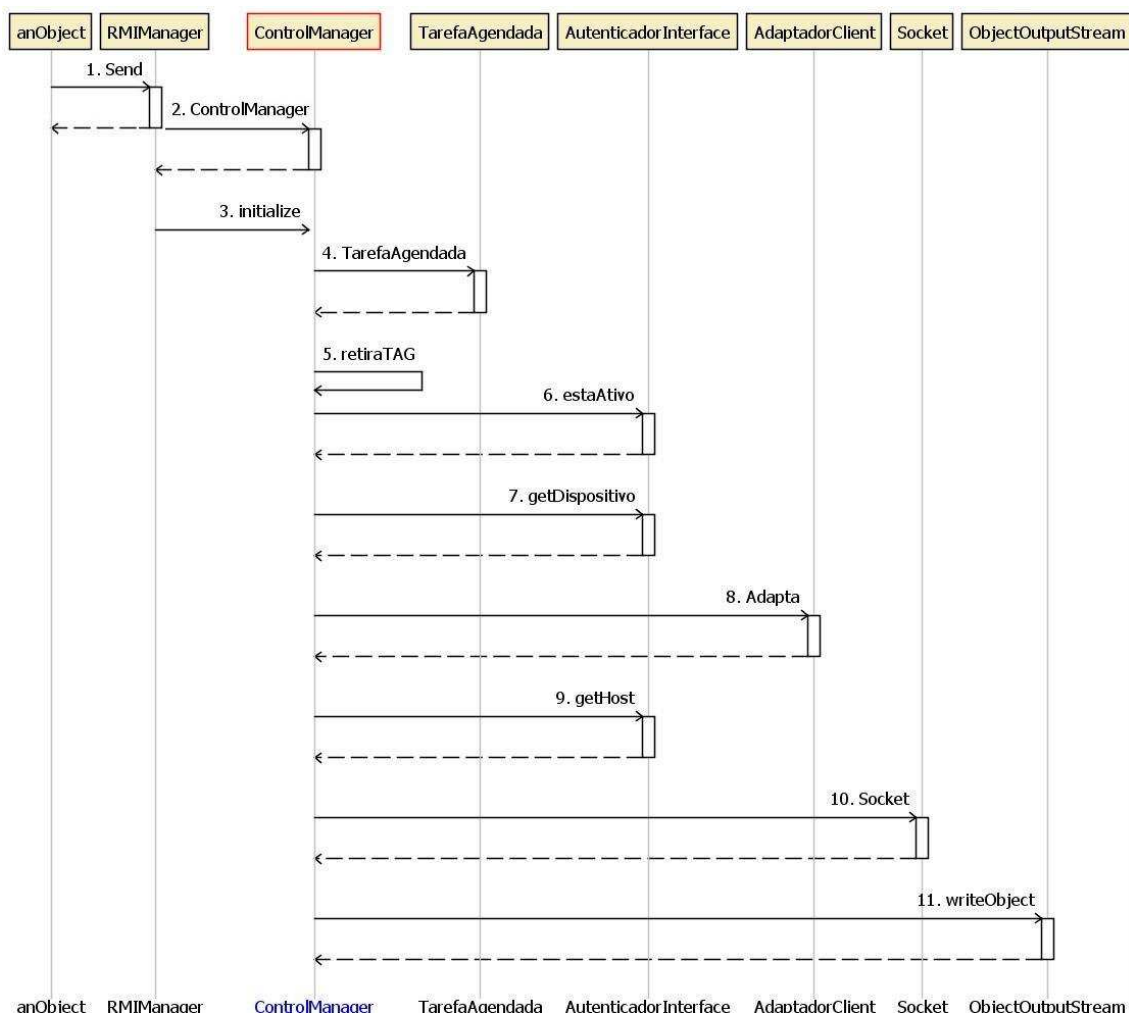


Figura 4.2: Diagrama de Seqüência do Serviço de Disseminação

a mensagem seja de acesso público não há a necessidade de consultar o serviço de autenticação para confirmar a identidade do usuário.

As próximas seções descrevem a sintaxe e semântica do arquivo que representa a mensagem e os módulos do serviço de disseminação.

4.1.2 A Mensagem

A mensagem é um arquivo XML (vide figura 4.3) com *tags* que informam as condições em que a mensagem deve ser enviada. Este arquivo XML é encapsulado por um objeto da classe *Arquivo* e guardado junto com os demais arquivos (também encapsulados em objetos da classe *Arquivo*) que podem vir anexos à mensagem no

vetor que encapsula a mesma. Este vetor é o objeto repassado entre os serviços que têm acesso à mensagem. Toda mensagem é gerada pelo Servidor do Portal de Compras e repassada para o serviço de disseminação onde cada mensagem é validada através de um arquivo XSD sendo que o arquivo XML que define a mensagem deve ser o primeiro arquivo do vetor.

```

- <Mensagem>
- <!--
  xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
-->
<!-- xsi:noNamespaceSchemaLocation="mensagem.xsd" -->
<instantaneo>true</instantaneo>
<!-- true, false -->
<time_out>0</time_out>
<!-- milisegundos para expirar o envio -->
<retorno>null</retorno>
<!-- html, wml, pdf, texto, null -->
<dispositivo>desktop</dispositivo>
<!-- desktop, pda, celular -->
<destino>redin</destino>
<titulo>Concorrência</titulo>
<!-- Concorrência, leilão, tomada de preços, convite... -->
- <corpo>
  Contratação de pessoa jurídica especializada na prestação de serviços técnicos de informática e telecomunicações,
  com vistas a desenvolver, otimizar e manter em perfeito funcionamento os sistemas, serviços e a infra-estrutura
  computacional do Ministério da Educação (MEC). Entrega das propostas: 03/05/2005 às 09h30.
</corpo>
</Mensagem>

```

Figura 4.3: Arquivo XML que descreve uma mensagem

A maioria das *tags* utilizadas na mensagem não é utilizada pelo serviço de disseminação mas sim por outros serviços do projeto pBuy como o serviço de Apresentação. Como podemos ver na Tabela 4.1 as *tags* *instantaneo*, *timeout* e *delay* definem ações distintas por parte do disseminador para o envio da mensagem. Caso o valor da *tag* *instantaneo* seja verdadeiro, a mensagem tem prioridade sobre as outras e deve ser enviada imediatamente. O valor de *timeout* indica se a mensagem tem um prazo de validade ou o serviço de disseminação deve tentar entregá-la indefinidamente. O valor da *tag* *delay* indica quanto à mensagem deve ficar em espera antes da primeira tentativa de envio. O valor da *tag* *delay* deve ser obrigatoriamente menor que o valor da *tag* *timeout*, sob pena de não envio da mensagem.

As outras *tags* que o serviço de disseminação utiliza para enviar a mensagem

instantaneo	timeout	delay	ação executada
true	>0	X	Enviar urgente até o timeout expirar.
true	0	X	Enviar urgente indefinidamente.
false	>0	>0	Enviar a partir de delay até timeout.
false	0	>0	Enviar a partir de delay indefinidamente.
false	>0	0	Enviar agora até timeout.
false	0	0	Enviar a partir de agora indefinidamente.

Tabela 4.1: Tabela de Valores e Ações Correspondentes

são a *tag* destino que guarda o(s) usuário(s) destino da mensagem e a *tag* dispositivo que o Serviço de disseminação deve preencher com o valor repassado pelo serviço de autenticação antes de solicitar ao serviço de apresentação a adaptação do conteúdo.

4.1.3 RMIManager

Este é o módulo responsável pela recepção das mensagens que devem ser enviadas. Ele implementa o servidor RMI e também confere se o arquivo XML que descreve a mensagem é válido. Quando o servidor do portal de compras faz uma requisição de envio (chamando o método `Send()` da interface `RMIInterface`) ele ativa o servidor RMI que utiliza um XSD para validar a sintaxe da mensagem. Se algum erro no conteúdo da mensagem é detectado o `RMIManager` causa uma exceção à aplicação que o chamou como forma de indicar que a mensagem não poderá ser enviada.

Em seguida o `RMIManager` inicializa um objeto `ControlManager` através do método `Initialize()` e daí por diante o `ControlManager` será o módulo ativo que cuidará do envio da mensagem.

Os dados recebidos pelo `RMIManager` estão contidos em uma estrutura `Vector` e encapsuladas por um tipo `Arquivo`. Pelo uso do `Vector` temos uma independência do número de arquivos anexos que uma mensagem pode conter, da mesma forma encapsular todos os arquivos sob a classe `Arquivo` provê uma forma padronizada de acesso aos mesmos. O arquivo XML que descreve a mensagem é o primeiro arquivo da estrutura `Vector` e os demais arquivos não seguem qualquer ordem, desde que todos os arquivos citados no XML estejam presentes no `Vector`.

4.1.4 ControlManager

Este é o módulo principal do serviço de disseminação. Suas funções incluem instanciar a fila de mensagens, fazer requisições (como cliente RMI) de autenticação de usuário, de endereço de dispositivo e adaptação da mensagem. Também é este módulo que realiza o envio das mensagens seguindo os parâmetros especificados pelo Portal de Compras. Quando um objeto `ControlManager` é instanciado pela primeira vez ele cria a fila que guardará as mensagens que devem ser enviadas no futuro. Para a implementação da fila foi usada a classe `Timer` do pacote `java.util` disponível por padrão no J2SE. Esta classe provê serviços de realização de tarefas temporizadas. A classe `Timer` usa a classe abstrata `TimerTask` para guardar as tarefas na fila de execução, desta forma foi necessária à criação de uma classe `TarefaAgendada` que implementou `TimerTask` e serviu de encapsulamento a mensagem e seus atributos.

Quando uma das mensagens que está presente na fila deve ser enviada (seguindo as restrições de tempo impostas pelo Portal de Compras) o método `Run()` do `ControlManager` é chamado, iniciando uma tentativa do processo de envio que inclui verificar autenticação do usuário (quando necessário), pedir o tipo do dispositivo do usuário para o serviço de autenticação para a adaptação, adaptação da mensagem, resolução do endereço do usuário e o envio propriamente dito. Se por algum motivo (usuário não está ativo no momento, autenticação falhou, etc.) o envio não foi possível, a mensagem é reagendada para tentativas posteriores de envio.

4.2 Adaptação do Serviço de Disseminação para o Uso do Bonjour

Esta seção descreve as modificações realizadas no serviço de disseminação para que o mesmo utilize o protocolo Bonjour para a descoberta de usuários e serviços. O protocolo Bonjour foi escolhido por ser um padrão aberto que permite interoperabilidade sem a necessidade de qualquer configuração por parte do usuário. Ele age de um modo transparente e automatizado na resolução de nomes e descoberta de serviços suportando tanto redes locais como também endereços em sub-redes

diferentes.

A modificação necessária foi a criação de uma classe adicional para tratar dos eventos que devem ser gerados para o Bonjour e também para processar as respostas destes eventos. Outras modificações foram feitas no `ControlManager` para passar a utilizar a nova classe criada para descobrir e resolver o endereço do dispositivo do usuário ao invés do serviço de autenticação.

A figura 4.4 mostra o novo diagrama de classes com a adição do módulo que faz uso do Bonjour para o registro e descoberta de serviços e usuários. Na figura podemos ver que o módulo `BonjourManager` tem uma dependência para com o `ControlManager` e também que o primeiro deve estender três interfaces do protocolo Bonjour. Continua sendo necessário pedir ao Serviço de Autenticação se o usuário está autenticado e qual o tipo do dispositivo que o mesmo usa no momento.

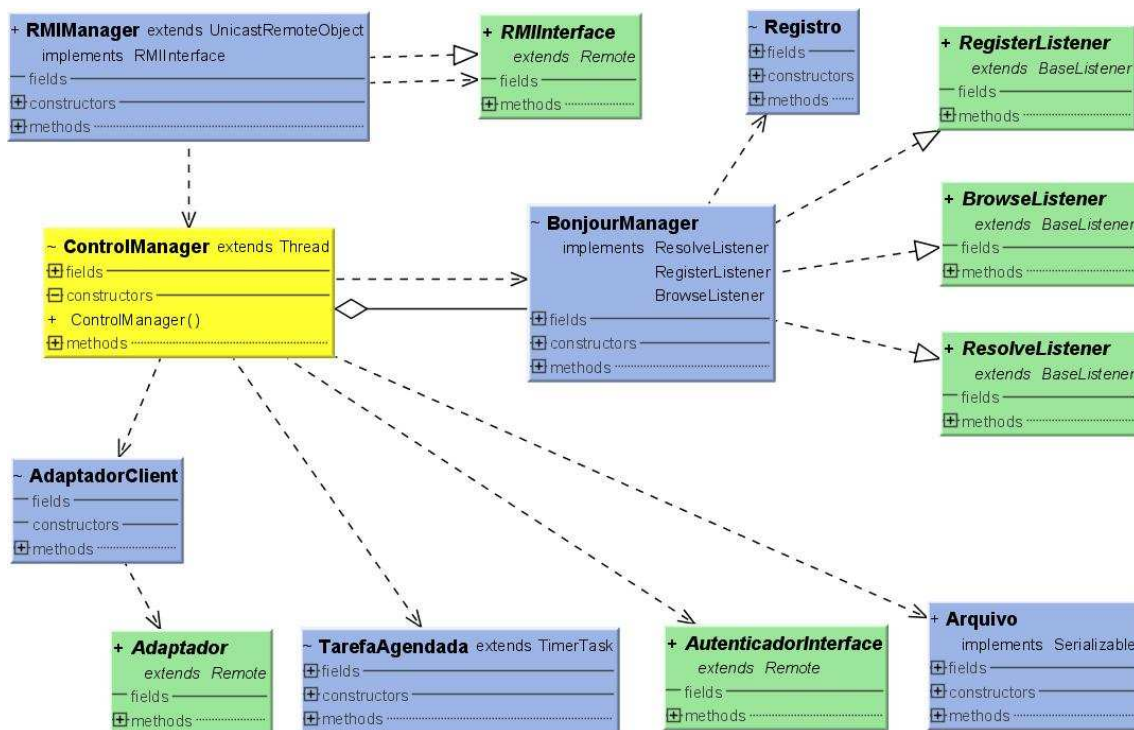


Figura 4.4: Diagrama de Classes com a adição BonjourManager.

A adição deste módulo também afeta a interação entre os módulos de forma que o diagrama de seqüência da figura 4.5 é o novo encadeamento de eventos. Inicialmente a aplicação envia a mensagem por RMI e esta é recebida pelo `RMIManager`,

que é o módulo responsável por esta tarefa. O `RMIManager`, por sua vez, valida a mensagem através de um arquivo XSD e se esta for inconsistente ele lança uma exceção remota como um aviso à aplicação que enviou a mensagem de que a mesma não poderá ser enviada. No caso de uma mensagem válida o `RMIManager` instancia um objeto da classe `ControlManager` para cuidar do resto do processo de envio. O `ControlManager` pergunta então para o Serviço de Autenticação se o usuário está autenticado e qual seu dispositivo. De posse desta resposta, o `ControlManager` solicita ao Serviço de Apresentação que formate a mensagem para o dispositivo em questão. Em seguida, este solicita ao `BonjourManager` o endereço físico do usuário-destino da informação e cria um *socket* TCP com o Serviço de Comunicação do dispositivo do usuário para a efetivação do envio. Caso a mensagem seja de acesso público não há a necessidade de consultar o serviço de autenticação para confirmar a identidade do usuário.

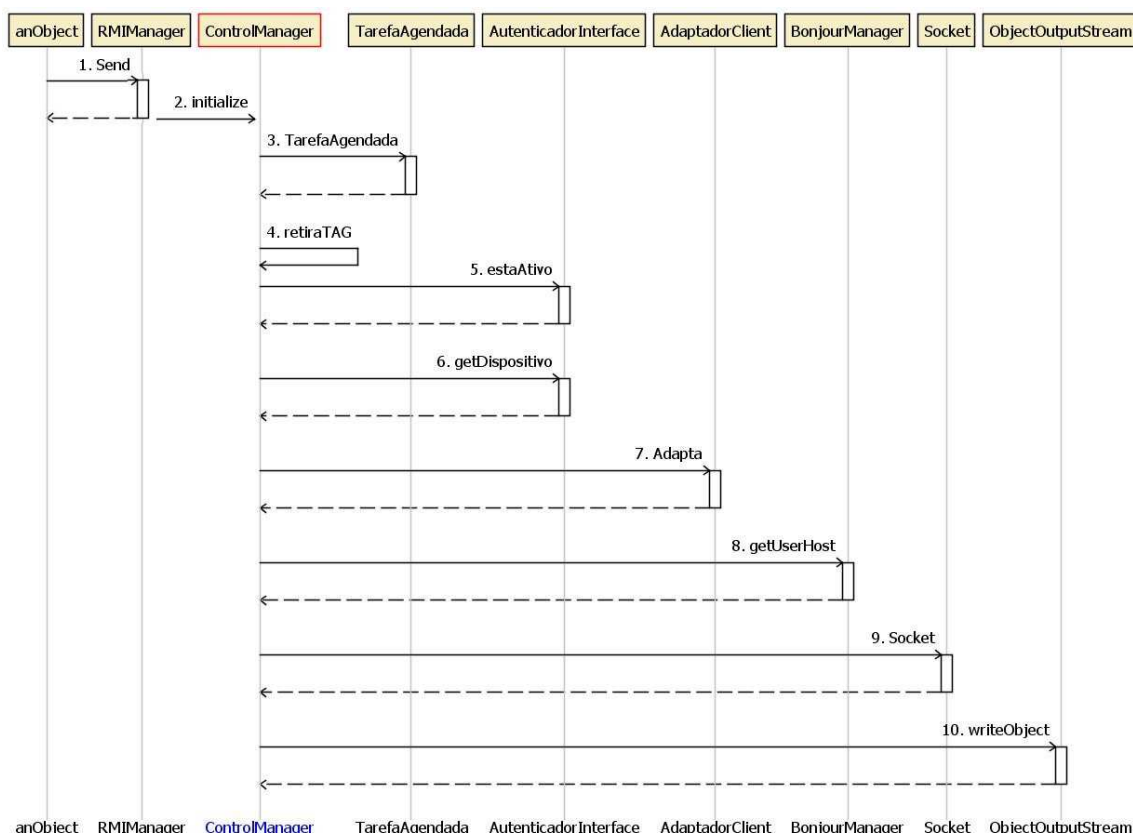


Figura 4.5: Diagrama de Seqüência com a adição BonjourManager.

BonjourManager é o módulo responsável pela descoberta e resolução de endereço dos dispositivos dos usuários pervasivos bem como dos outros serviços do projeto pBuy. O BonjourManager implementa as *Interfaces* e *Listeners* requisitados pelo mDNSResponder para realizar as tarefas acima citadas. O BonjourManager disponibiliza os meios para encontrar tanto os serviços do pBuy como os usuários. Este módulo implementa as interfaces RegisterListener(), BrowseListener() e ResolveListener() e recebe os eventos que resultam do registro, procura e resolução de referências a serviços anunciados através do mDNS.

O BonjourManager está continuamente procurando por usuários ativos que são guardados em uma lista. Desta forma quando houver alguma mensagem para ser enviada uma simples consulta a esta lista é necessária para saber se usuário encontra-se *online*. Caso isto não fosse feito, uma requisição de procura para o Bonjour seria necessária e desta forma além de um aumento no tempo de envio das mensagens teríamos uma sobrecarga extra sobre a rede e um desperdício de banda.

Cabe salientar que o sistema legado exige a autenticação do usuário e este é o motivo da existência do serviço de autenticação. O projeto pBuy prevê a substituição do serviço de autenticação pelo uso do Bonjour também para autenticação. O Bonjour ficará responsável pelas tarefas de autenticação, descoberta de serviços e endereçamento lógico.

4.3 Tecnologias e Ferramentas Utilizadas

Esta seção cita e descreve brevemente algumas das tecnologias relevantes que foram utilizadas na implementação deste serviço. Inicialmente são apresentadas a Plataforma Java e a Invocação de Método Remoto (*Remote Method Invocation*) [SUN 2005a], que permite a portabilidade desejada ao serviço e ao sistema como um todo. Em seguida falamos sobre o *eXtensible Markup Language* (XML)[XML 2005], que é utilizado neste trabalho como forma de guardar as informações que devem ser enviadas ao usuário. Finalmente é realizada uma explanação sobre o Bonjour, utilizado neste projeto para permitir um padrão de endereçamento e uma forma de

descoberta de serviços.

4.3.1 Plataforma Java e RMI

Em um ambiente de computação pervasiva, o uso dos mais diversos tipos de dispositivos gera uma heterogeneidade nas arquiteturas e, desta forma, desencadeia o problema de portabilidade de código. Para resolver este problema, optou-se pelo uso da plataforma Java, pois a mesma possibilita, através de código interpretado, a portabilidade necessária para resolver o problema citado.

Outro aspecto relevante e pertinente é a comunicação através de Invocação Remota de Métodos (RMI) que existe na plataforma Java. Esta forma de comunicação orientada a objetos deriva da Chamada Remota de Procedimentos (RPC) e permite que um objeto localizado em uma máquina remota invoque a execução de um método de outro objeto, de forma que esta seja quase como uma invocação local. Esta forma de comunicação é útil para a comunicação entre os serviços do lado do servidor, no projeto pBuy, por simplificar a implementação da comunicação entre os serviços.

4.3.2 Extensible Markup Language (XML)

Segundo o W3C, XML (*eXtensible Markup Language*) [XML 2005] é um formato de texto derivado do SGML (ISO 8879) que é simples e muito flexível. O XML foi inicialmente projetado para superar os desafios da publicação eletrônica em larga escala, e está tendo um importante papel na troca de uma vasta gama de dados na Internet. Ele é utilizado para descrever semântica e o conteúdo dos mais diversos tipos de dados, desde documentos de texto simples até arquivos de áudio.

A mais importante característica do XML se resume em separar a interface com o usuário (a forma como apresentamos) dos dados estruturados. Como exemplo o HTML especifica como os dados devem ser apresentados na tela do navegador e o XML define o conteúdo do documento. No HTML as *tags* são utilizadas para dizer o tamanho da letra, fonte, cor do texto, etc. No XML as *tags* são utilizadas para descrever os dados presentes no arquivo, título, assunto, data, etc. No projeto

pBuy o principal uso do XML é para guardar as mensagens aos usuários enquanto transitam pelos serviços.

O padrão XML define ainda outros recursos interessantes como XSL (eXtensible Style Language) [XSL 2005] e CSS (Cascading Style Sheets) [CAS 2005] para apresentação dos dados, XSD (XML Schema Definition) para definição e validação dos arquivos XML, entre outros.

4.3.3 Bonjour

Segundo a Apple o Bonjour é uma tecnologia de rede baseada em padrões e aberta, que conecta automaticamente dispositivos eletrônicos em uma rede, permitindo interoperabilidade sem a necessidade de qualquer configuração por parte do usuário.

Quando um dispositivo é adicionado, a rede Bonjour configura o dispositivo usando uma técnica chamada endereçamento de rede local (*link-local addressing*). Se há um servidor DHCP disponível será usado o endereço designado por ele; caso contrário, o dispositivo seleciona aleatoriamente um IP de uma faixa pré-determinada pela IANA (*Internet Assigned Numbers Authority*) para este fim e assume este endereço para si (os endereços estão na faixa 169.254.xxx.xxx). O dispositivo, então, envia uma mensagem pela rede para determinar se já existe alguém que esteja utilizando o mesmo endereço. Caso o endereço já esteja em uso, ele volta a selecionar aleatoriamente outro endereço até que encontre um disponível.

Após estes passos, o dispositivo está configurado para usar a rede. Ainda existe a necessidade de descobrir os serviços oferecidos e anunciar os seus serviços, mesmo sem a infra-estrutura de resolução de nomes. Para isso, o Bonjour utiliza o mDNS-SD [MUL 2005], uma forma alternativa de resolução de nomes baseada no sistema DNS que funciona em redes não infra-estruturadas. Para descobrir um serviço, uma requisição mDNS é lançada e recebe como respostas os nomes de todos os serviços que se encaixam na requisição. Quando o autor da requisição mDNS for usar um dos serviços que respondeu, será feita a tradução do nome para o endereço e obtidas as demais informações de conexão necessárias.

Capítulo 5

Conclusão e Trabalhos Futuros

A computação pervasiva é um novo cenário computacional que está crescendo com o desenvolvimento dos dispositivos móveis e sem-fio. Este novo cenário computacional pode ser resumido por 3 A's: *Anywhere, Anytime, Any device*, e visa disponibilizar o ambiente computacional do usuário em qualquer tempo, lugar e com qualquer dispositivo.

Apesar dos avanços realizados no hardware e software no sentido de tornar realidade a computação pervasiva, muito ainda falta para que esta se torne presente no nosso dia-a-dia. Além disso, o novo paradigma da computação pervasiva cria a necessidade de saber o contexto em que o usuário está acessando o sistema, isto é, com que dispositivo, que características tal dispositivo possui, que padrões de comunicação suporta, etc.

Desde modo este trabalho apresenta a implementação de um serviço de disseminação de dados que suporte os requisitos impostos pela computação pervasiva para os desenvolvedores de aplicações que necessitam de tais características e mais especificamente visa servir ao projeto pBuy no envio de mensagens aos usuários móveis e pervasivos.

Este serviço foi construído com o uso do J2SE e utiliza os serviços disponibilizados pelo protocolo Bonjour. Foi testado utilizando-se os demais serviços que fazem parte do projeto pBuy, demonstrando assim sua integração e funcionalidades. Com o uso do Bonjour temos a vantagem de que o serviço se torna capaz de encontrar os usuários independentemente de estarem na mesma rede local ou espalhados ge-

ográfica e logicamente porque o Bonjour (como implementado pela Apple) permite a tradução de endereços NAT transparente para o serviço e abstrai essa dificuldade para o utilizador do mesmo (no caso o serviço de disseminação).

O trabalho realizado até agora deixa margem para melhoramentos futuros. Entre os quais podemos citar:

- O desenvolvimento de um módulo de acompanhamento da fila de mensagens, para que seja possível algum nível de consciência do estado do serviço por parte dos administradores.
- Uma forma de confirmação de envio e recebimento de mensagens para garantir que o usuário está ciente de alguma informação de extrema importância e/ou urgência.
- Modificação da prioridade de uma mensagem na fila de envio, de forma que uma mensagem que se torna urgente após ser repassada ao disseminador mas que ainda não foi enviada (possui um tempo de espera por exemplo) receba uma modificação para que seja enviada imediatamente ou vice-versa.
- O Bonjour faz uso de registros TXT do protocolo DNS. Tais registros podem ser usados para guardar informações acerca de características exclusivas do dispositivo, preferências do usuário, redirecionamento de mensagens, etc. O uso futuro destas funcionalidades fornecidas pelo Bonjour daria um suporte à pervasividade em um nível mais avançado para o serviço de disseminação e este seria beneficiado pelo uso de um padrão internacionalmente reconhecido e implementado.

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 WORLD INTERNATIONAL WORKSHOP, WDAG'9, 1997, Saarbrücken, Germany. **Anais. . .** Springer, 1997. v.1320, p.126–140.
- [APP 2005] APPLE developer connection - bonjour. Disponível em <http://developer.apple.com/networking/bonjour/index.html> - acessado em 02/08/2005.
- [AUG 2004] AUGUSTIN, I. **Abstrações para uma linguagem de programação visando aplicações móveis em um ambiente de pervasive computing**. 2004. Tese de Doutorado — Universidade Federal do Rio Grande do Sul.
- [BEL 2005] BELUSSO, R. **Serviço de apresentação consciente do dispositivo em um ambiente pervasivo**. 2005. Monografia (Bacharel em Ciência da Computação) — Universidade Federal de Santa Maria.
- [CAS 2005] CASCADING style sheets 2.1 specification. w3c candidate recommendation. Disponível em <http://www.w3.org/TR/CSS21/>, acessado em agosto / 2005.

- [DAT 2005] DATAMAN - mobile computing laboratory. Disponível em <http://www.cs.rutgers.edu/dataman/index.html> - acessado em 03/08/2005.
- [FRA 98] FRANKLIN, M.; ZDONIK, S. Data in your face: push technology in perspective. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON THE MANAGEMENT OF DATA, 1998, Seattle USA. **Anais...** [S.l.: s.n.], 1998.
- [HEN 2001] HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY, A. Infrastructure for pervasive computing: challenges. In: WORKSHOP ON PERVASIVE COMPUTING, INFORMATIK, 2001, Vienna, Austria. **Anais...** [S.l.: s.n.], 2001.
- [IMI 97] IMIELINSKI, T.; VISWANATHAN, S.; BRADINATH, B. R. Data on air: organization and access. **IEEE Transactions on Knowledge and Data Engineering**, v.9, n.3, may 1997.
- [MUL 2005] MULTICAST dns. Disponível em <http://www.multicastdns.org/> - acessado em 04/10/2005.
- [PIR 2005] PIRES, R. **Serviço de comunicação consciente do estado da rede em um ambiente pervasivo**. 2005. Monografia (Bacharel em Ciência da Computação) — Universidade Federal de Santa Maria.
- [RFC 2005] RFC 2136 (rfc2136) - dynamic updates in the domain name system (dns update). <http://www.faqs.org/rfcs/rfc2136.html> - acessado em 03/08/2005.
- [SAH 2003] SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21st century. **IEEE Pervasive Computing**, v.36, n.3, p.25–31, mar 2003.

- [SAT 2001] SATYANARAYANAN, M. Pervasive computing; vision and challenges. **IEEE Personal Communications Magazine**, v.8, n.4, p.10–17, aug 2001.
- [STA 97] STATHATOS, K. Adaptive data broadcast in hibrid networks. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 23., 1997. **Anais...** [S.l.: s.n.], 1997.
- [SUN 2005] SUN microsystems. Java 2 Platform, Standard Edition (J2SE). Disponível em <http://java.sun.com/j2se/index.jsp>, acessado em agosto/2005.
- [SUN 2005a] SUN microsystems. Java Remote Method Invocation (Java RMI). Disponível em <http://java.sun.com/products/jdk/rmi>, acessado em agosto/2005.
- [WIK 2005] WIKIPEDIA, the free encyclopedia - appletalk. Disponível em <http://en.wikipedia.org/wiki/AppleTalk> - acessado em 04/10/2005.
- [XML 2005] XML, extensible markup language. Disponível em <http://www.w3.org/XML/>, acessado em agosto / 2005.
- [XSL 2005] XSL, extensible stylesheet language. Disponível em <http://www.w3.org/Style/XSL/>, acessado em agosto / 2005.