

# Habilitando Suporte a Recomendação de Serviços em Redes Baseadas na Tecnologia UPnP

Ádrian Lívio Vasconcelos Guedes

Dissertação de Mestrado submetida à Coordenadoria do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Engenharia de Computação

Angelo Perkusich, Dr.  
Orientador

Campina Grande, Paraíba, Brasil  
©Ádrian Lívio Vasconcelos Guedes, Agosto de 2008

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

# Habilitando Suporte a Recomendação de Serviços em Redes Baseadas na Tecnologia UPnP

Ádrian Lívio Vasconcelos Guedes

*Dissertação de Mestrado apresentada em Agosto de 2008*

Angelo Perkusich, Dr.  
Orientador

Campina Grande, Paraíba, Brasil, Agosto de 2008

## Resumo

O surgimento de dispositivos capazes de disponibilizar seus serviços em redes locais e na Internet levou ao surgimento de protocolos que possibilitem ao usuário descobrir e utilizar estes serviços. Dentre os Protocolos de Descoberta e Disponibilização de Serviços, o UPnP permite que dispositivos que implementam a sua especificação se conectem na rede e possam disponibilizar seus serviços permitindo que um usuário ou outros dispositivos UPnP possam utilizar estes serviços. Em ambientes com um grande número de serviços disponíveis, o usuário pode ter dificuldades de localizar o serviço que para ele seja realmente relevante dentre tantos disponíveis. Os Sistemas de Recomendação têm como objetivo ajudar o usuário a lidar com uma grande quantidade de opções disponíveis, baseando-se no histórico de utilização do sistema por parte do próprio usuário ou de usuários com gostos semelhantes. Neste trabalho é proposta uma arquitetura para a utilização de Sistemas de Recomendação para a recomendação de serviços em redes baseadas na tecnologia UPnP.

# Abstract

The emergence of devices capable of providing its services in local networks and the Internet has led to the development of protocols that allow the user to discover and use these services. Among the Service Discovery Protocols, UPnP enables devices that implement its specification to connect to the network and make their services available to a user or other UPnP devices. In environments with a large number of services available, the user may have difficulties to find the service that is really important for him among many available. The Recommender Systems are designed to help the user to deal with a large number of options available, the recommendations are based on historical of use of the system from user or users with similar tastes. This work proposes an architecture for the use of Recommender Systems to enable recommendation of services in networks based on UPnP technology.

## Dedicatória

Dedico este trabalho a meus pais, Ednaldo e Rosa, que sempre me apoiaram com seu amor e dedicação, sempre incentivaram meus estudos e me ensinaram o caminho a seguir.

# Agradecimentos

Gostaria de expressar meu sentimento de gratidão a todos aqueles que direta ou indiretamente contribuíram para a conclusão deste trabalho e mesmo correndo o risco de deixar de citar alguém importante, agradeço a pessoas especiais sem as quais esta jornada teria sido muito mais difícil ou até impossível.

Agradeço a minha família, meus pais Ednaldo e Rosa a quem eu dedico este trabalho, a Alysson por ser tão trekker quanto eu e a Alan por sempre estar interessado no meu trabalho, ou simplesmente por serem os melhores irmãos do mundo.

A Aurea Lúcia e Maria do Socorro, por sempre acreditar em mim e fazerem de tudo para me ajudar a chegar até onde eu cheguei.

A Sebastião, Niça, Mayara e Daniel, por serem grandes amigos e uma segunda família para mim.

A meus Super Melhores Amigos Jonas, Paulo, Alixandra, Erik, Kesia, Jerônimo, Olympio, Zé Luis, Danilo, Daniel e Vilma. Amigos estes que me acompanham a anos e estiveram comigo durante este trabalho, ajudando, aconselhando, dando bronca, ou apenas sendo amigos.

A Aureliano, Joedna, Arthur e Jaqueline que me acolheram em sua casa como um membro de sua família.

A Rozalina por todo amor, compreensão e por simplesmente me ouvir durante os momentos mais difíceis deste trabalho.

Ao professor Angelo por toda a ajuda e orientação durante o decorrer deste trabalho.

Aos meus Amigos Mariegno, Zilminha, Marcílio, Rosa, Mineli, Josélia, Diego, Yuri, Luiz Paulo, Mateus, Marcos Moraes, Mario Hozano, Hyggo, Glauber, André Felipe, Saulo, Raul e Leandro sempre dispostos a compartilhar momentos de felicidade e a ajudar nos momentos de necessidade.

E a todas as pessoas que contribuíram com seus impostos que foram revertidos em apoio financeiro a esta pesquisa através da CAPES.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivo . . . . .	2
1.2	Relevância . . . . .	2
1.3	Estrutura da Dissertação . . . . .	3
<b>2</b>	<b><i>Universal Plug and Play (UPnP)</i></b>	<b>4</b>
2.1	Arquitetura UPnP . . . . .	5
2.2	Outros Protocolos de Descoberta de Serviços . . . . .	12
<b>3</b>	<b>Sistemas de Recomendação</b>	<b>14</b>
3.1	Filtragem Baseada em Conteúdo . . . . .	15
3.2	Filtragem Colaborativa . . . . .	16
3.3	Filtragem Híbrida . . . . .	17
3.4	Exemplos de Sistemas de Recomendação . . . . .	18
<b>4</b>	<b>Arquiterura do Sistema de Recomendação para Dispositivos UPnP</b>	<b>20</b>
4.1	Visão Geral . . . . .	20
4.2	Servidor de Recomendação . . . . .	21
4.2.1	<i>Hybrid Recommender</i> . . . . .	22
4.2.2	<i>Profiler</i> . . . . .	23
4.2.3	<i>Recommender Facade</i> . . . . .	24
4.2.4	<i>Users Databases</i> . . . . .	25
4.3	<i>Control Point</i> . . . . .	26
4.3.1	<i>Control Point API</i> . . . . .	27
4.3.2	<i>Control Point Manager</i> . . . . .	27
4.3.3	<i>Server Access Layer</i> . . . . .	27
4.3.4	<i>User Interface Facade</i> . . . . .	28
4.3.5	<i>Usage Register</i> . . . . .	28
4.3.6	<i>Device Descriptor</i> . . . . .	28
4.4	Funcionamento . . . . .	28



4.4.1	Busca . . . . .	29
4.4.2	Utilização . . . . .	30
4.4.3	Novo dispositivo UPnP entra na rede . . . . .	31
4.4.4	Dispositivo UPnP Sai da rede . . . . .	32
4.5	Implementação do Protótipo . . . . .	33
4.5.1	UPnP Media Server . . . . .	33
4.5.2	<i>Control Point</i> . . . . .	34
4.5.3	<i>Recomender Server</i> . . . . .	35
<b>5</b>	<b>Conclusão</b>	<b>37</b>
	<b>Referências Bibliográficas</b>	<b>39</b>

# Lista de Figuras

2.1	Pilha de Protocolos utilizada no UPnP. . . . .	5
2.2	Etapa de Descoberta. . . . .	7
2.3	Etapa de Descrição . . . . .	8
2.4	Etapa de Controle. . . . .	9
2.5	Etapa de Tratamento de Eventos. . . . .	10
2.6	Etapa de Apresentação. . . . .	11
4.1	Integração de um Servidor de Recomendação a uma rede UPnP . . . . .	21
4.2	Arquitetura do Servidor de Recomendação . . . . .	22
4.3	Arquitetura do <i>Control Point</i> com Suporte a Recomendação . . . . .	26
4.4	Inicialização e busca por dispositivos UPnP. . . . .	29
4.5	Utilização e alimentação dos perfis de usuário. . . . .	30
4.6	Entrada de um novo dispositivo UPnP na rede. . . . .	31
4.7	Remoção de um dispositivo UPnP da rede. . . . .	33

# Lista de Tabelas

4.1	Históricos dos usuários . . . . .	25
4.2	Perfis dos usuários . . . . .	26
4.3	Serviços do UPnP <i>Media Server</i> . . . . .	34
4.4	Ações do <i>Content Directory</i> . . . . .	34
4.5	Ações do <i>Connection Manager</i> . . . . .	35

# Glossário

**API** *Application Programming Interface*

**DHCP** *Dynamic Host Configuration Protocol*

**DNS** *Domain Name Service*

**FBC** *Filtragem Baseda em Conteúdo*

**FC** *Filtragem Colaborativa*

**GENA** *General Event Notification Architecture*

**GUI** *Graphical User Interface*

**HTTP** *HyperText Transfer Protocol*

**HTTPMU** *HTTP over Multicast*

**HTTTPU** *HTTP over UDP*

**IETF** *Internet Engineering Task Force*

**INS** *Intentional Naming System*

**IP** *Internet Protocol*

**JVM** *Java Virtual Machine*

**PC** *Personal Computer*

**SCSL** *Sun Community Source License*

**SLP** *Service Location Protocol*

**SOAP** *Simple Object Access Protocol*

**SSDP** *Simple Service Discovery Protocol*

**TCP** *Transport Control Protocol*

**UDP** *User Datagram Protocol*

**UPnP** *Universal Plug and Play*

**URL** *Universal Resource Locator*

**UUID** *Universally Unique Identifier*

**XML** *Extensible Markup Language*

# Capítulo 1

## Introdução

O número de dispositivos com capacidade de conectividade em redes locais e com acesso à Internet vem crescendo rapidamente. Estes dispositivos se tornaram capazes de consumir e disponibilizar serviços nas redes em que estão inseridos. Com isso tornou-se necessária a criação de protocolos que permitissem ao usuário destes dispositivos descobrir e utilizar estes serviços. A tecnologia UPnP [1] permite que dispositivos que implementam a sua especificação se conectem na rede e possam disponibilizar seus serviços permitindo que um usuário ou outros dispositivos possam utilizar estes serviços.

O surgimento destes protocolos e sua adoção pela indústria provoca um aumento do número de dispositivos capazes de disponibilizar serviços em rede, criando um cenário onde praticamente qualquer tipo de dispositivo pode possuir esta capacidade. O fórum UPnP [2], por exemplo, define vários tipos de serviços que podem ser disponibilizados por dispositivos, estes serviços incluem: iluminação, impressão, monitoramento por câmeras, aquecimento e climatização. Além disso, os fabricantes podem especificar seus próprios dispositivos.

Neste cenário, o grande número de serviços disponíveis pode sobrecarregar o usuário, fazendo com que seja difícil para ele localizar o serviço que deseja utilizar entre tantos disponíveis. Um problema semelhante é observado no caso da grande quantidade de informação disponível através da Internet, TV, cinema, literatura, etc. Os Sistemas de Recomendação [3–6] têm como objetivo ajudar o usuário a lidar com toda esta informação, sugerindo itens de interesse do usuário dentre um conjunto de itens disponíveis. A recomendação é feita com base nas preferências do usuário, seu no comportamento e de informações sobre os itens disponíveis.

A quantidade de informação existente nas diversas mídias pode dificultar a escolha do usuário sobre qual item de mídia, dentre os tantos disponíveis ele irá consumir. Da mesma forma, a quantidade crescente de dispositivos que disponibilizam seus serviços através de tecnologias como o UPnP fazem com que o usuário também tenha dificuldades

para encontrar um serviço que lhe será útil. Então da mesma maneira que um Sistema de Recomendação pode ajudar o usuário a encontrar um item de mídia que tenha uma maior probabilidade de lhe agradar, este tipo de sistema pode ser utilizado para ajudar o usuário a localizar os serviços que lhe serão mais úteis.

## 1.1 Objetivo

O objetivo no contexto deste trabalho é desenvolver um suporte à recomendação de serviços em redes baseadas na tecnologia UPnP, de forma a filtrar dentre os diversos dispositivos UPnP disponíveis na rede, aqueles que o usuário geralmente utiliza, ou provavelmente irá utilizar.

O sistema deve ser composto basicamente por um Servidor de Recomendação, um ou mais *Control Points* UPnP com a capacidade de se comunicar com o Servidor de Recomendação e os dispositivos UPnP. Em uma rede UPnP o *Control Point* tem como funções: obter a descrição dos dispositivos e dos serviços por eles providos, fazer as chamadas aos serviços e monitorar o estado dos dispositivos.

O suporte a recomendação não deve interferir no funcionamento dos dispositivos UPnP. A forma com que *Control Points* e dispositivos UPnP interagem entre si deve se manter inalterada em relação à especificação UPnP. *Control Points* com e sem a capacidade de fornecer recomendações aos usuários também devem poder coexistir em uma mesma rede UPnP.

## 1.2 Relevância

Tecnologias como UPnP são responsáveis pela descoberta e disponibilização de serviços, porém a tarefa de localizar e escolher o serviço a ser usado ainda é função do usuário. Porém, em ambientes onde uma quantidade de serviços muito grande está disponível, no caso do UPnP, por exemplo, são definidos serviços para, controle de climatização, iluminação e multimídia, entre outros. A grande quantidade de serviços disponíveis pode gerar uma sobrecarga para o usuário que, dentre tantos disponíveis, não vai facilmente encontrar o serviço que deseja.

Com o desenvolvimento deste trabalho espera-se integrar o uso de Sistemas de Recomendação à tecnologia UPnP, de forma a filtrar a grande quantidade de dispositivos presentes na rede, apresentando os que sejam de maior interesse ao usuário.

## 1.3 Estrutura da Dissertação

O texto seguinte desta dissertação encontra-se organizado nos capítulos:

**Capítulo 2** Neste capítulo é apresentado a tecnologia UPnP e sua arquitetura, além de uma visão geral sobre protocolos de descoberta e disponibilização de serviços.

**Capítulo 3** Neste capítulo são apresentados os conceitos de Sistemas de Recomendação e as principais técnicas utilizadas. No capítulo 3 também são apresentados exemplos de sistemas de recomendação existentes.

**Capítulo 4** Este capítulo apresenta a arquitetura para a implementação do suporte a recomendação de serviços em redes baseadas na tecnologia UPnP.

**Capítulo 5** Apresenta as conclusões deste trabalho, assim como os trabalhos futuros.



## Capítulo 2

### *Universal Plug and Play (UPnP)*

Um dos desafios inerentes aos ambientes pervasivos consiste em como descobrir e tornar disponíveis ao usuário os serviços existentes no ambiente. Vários protocolos surgiram com o objetivo de tornar estes serviços disponíveis ao usuário, da forma mais transparente possível, nos quais o usuário deve realizar o mínimo de ações para poder localizar e utilizar o serviço desejado. Os protocolos de descoberta de serviços permitem aos dispositivos e aos serviços que se descubram, configurem-se e se comuniquem entre si [7]. Atualmente existem diversos protocolos de descoberta de serviços, desenvolvidos por empresas e institutos de pesquisa como UPnP (*Universal Plug and Play*), Jini, Rendezvous, INS (*Intentional Naming System*), Ninja, DEAPspace, Salutation, SLP (*Service Location Protocol*) [7].

A tecnologia Universal Plug and Play é um conjunto de protocolos de rede que tem como objetivo permitir a conectividade *peer-to-peer* em redes pervasivas entre PCs, aplicações inteligentes e dispositivos sem fio [8]. UPnP possui uma arquitetura de rede aberta e distribuída, criada com base em padrões amplamente adotados como IP, TCP, UDP, HTTP, SOAP e XML, sendo passível de integração com outras tecnologias, como por exemplo FireWire [9, 10], Bluetooth [11] e ZigBee [12, 13].

A tecnologia UPnP tem como objetivos:

- Prover uma conectividade entre redes *ad-hoc*, independente das suas arquiteturas, de uso fácil e flexível.
- Prover a transferência de dados entre redes de diferentes arquiteturas.
- Substituição dos *Drivers* de dispositivos por protocolos de uso comum.
- Ser independente de mídias, sistemas operacionais e de linguagem de programação.

Os dispositivos UPnP apresentam como principais características:

- Suporte à configuração zero.

- “Rede invisível”, isto é, a rede é enxergada de forma transparente para os dispositivos.
- Descoberta automática de uma ampla categoria de dispositivos dos mais diversos fabricantes.
- Um dispositivo UPnP tem a capacidade de unir-se a uma dada rede dinamicamente, obtendo um endereço IP, apresentando seus serviços e detectando a presença e as capacidades de outros dispositivos.
- Servidores de DHCP e DNS são opcionais, usados apenas se disponíveis.
- Capacidade de sair da rede sem deixar nenhuma inconsistência na mesma.

## 2.1 Arquitetura UPnP

Na arquitetura dos dispositivos UPnP [1] são definidos os protocolos para comunicação entre controladores, ou pontos de controle *Control Point* e dispositivos. Para descoberta, descrição, controle, tratamento de eventos e apresentação, o UPnP usa a pilha ilustrada na Figura 2.1.

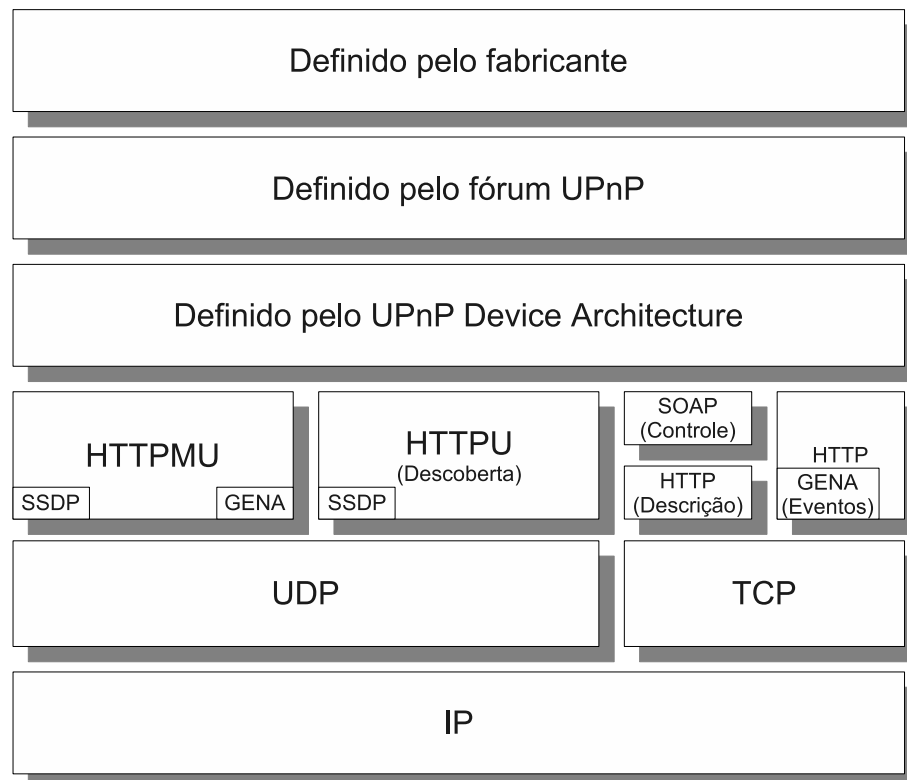


Figura 2.1: Pilha de Protocolos utilizada no UPnP.

O serviço de rede UPnP consiste dos seguintes passos:

1. **Endereçamento** Através do endereçamento o dispositivo UPnP obtém um endereço IP.
2. **Descoberta** Quando um dispositivo é adicionado na rede o UPnP permite ao dispositivo anunciar seus serviços aos *Control Points* na rede.
3. **Descrição** O *Control Point* requisita a descrição detalhada do dispositivo e de seus serviços através da URL de descrição obtida durante a descoberta.
4. **Controle** De posse da descrição do dispositivo, o *Control Point* pode requisitar ações a um serviço de um dispositivo.
5. **Tratamento de Eventos** Sempre que o valor alguma variável de estado de um dispositivo é alterada, todos os *Control Points* que se registraram interessados no estado dessa variável são notificados.
6. **Apresentação** Se um dispositivo UPnP possuir uma URL de apresentação, então o *Control Point* pode exibir esta página em um navegador web.

### Endereçamento

A base do serviço de rede UPnP é o endereçamento IP. Cada dispositivo deve ter um cliente DHCP e procurar por um servidor DHCP quando se conecta pela primeira vez na rede. Se o servidor estiver disponível o dispositivo deve usar o endereço IP fornecido. Caso contrário o dispositivo deve usar a técnica chamada de Auto IP para obter um endereço.

O Auto IP define como um dispositivo, de forma inteligente, escolhe um IP de um conjunto de endereços reservados, tornando o dispositivo capaz de se mover facilmente entre redes com ou sem gerenciamento (DHCP). Se durante a negociação com o DHCP o dispositivo recebe um nome de domínio (DNS), o dispositivo deve usar este nome em suas operações.

### Descoberta

A descoberta é o primeiro passo propriamente dito em um sistema UPnP. Quando um dispositivo é adicionado à rede, o protocolo de descoberta do UPnP permite que o dispositivo anuncie seus serviços aos pontos de controle na rede. De forma semelhante, quando um ponto de controle é adicionado à rede, o protocolo de descoberta do UPnP libera o ponto de controle para procurar por dispositivos de interesse na rede.

Em ambos os casos o ponto fundamental é a mensagem de descoberta, esta contém um pequeno conjunto de parâmetros essenciais sobre o dispositivo, como tipo, identificador e um apontador para detalhes.

Quando o novo dispositivo é adicionado na rede (*root device 1*, na Figura 2.2), ele faz *multicast* de mensagens de descoberta anunciando (*advertise*) dispositivos e serviços que contém. Qualquer ponto de controle interessado pode esperar por mensagens no endereço de *multicast* padrão. Estas mensagens informam sobre novas capacidades adicionadas à rede. De forma semelhante, quando um novo ponto de controle é adicionado à rede (*control point 3* na Figura 2.2), ele envia mensagens em *multicast* em busca dos dispositivos e serviços de seu interesse.

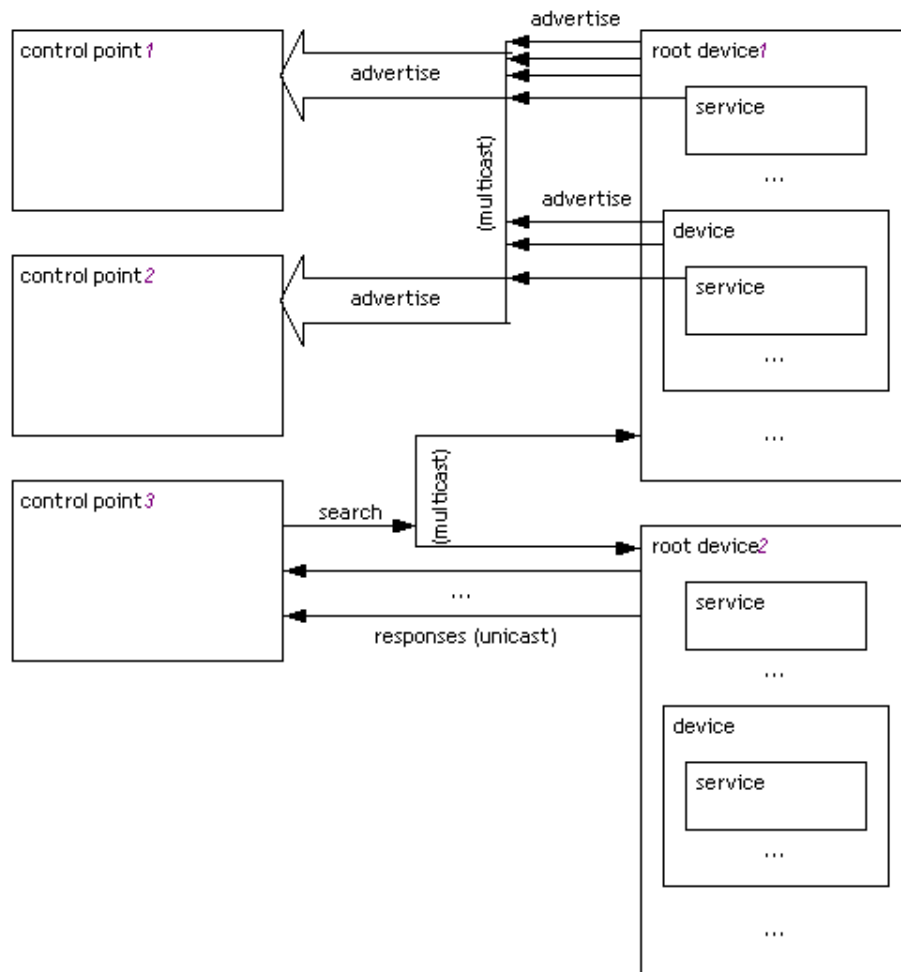


Figura 2.2: Etapa de Descoberta.

Todos os serviços devem esperar por mensagens vindas do endereço de *multicast* padrão e devem responder caso algum de seus dispositivos ou serviços correspondam ao critério de busca da mensagem (*unicast responses* do *root device 2* na Figura 2.2).

Quando um dispositivo é removido da rede, ele deve novamente enviar mensagens em *multicast* informando que aquele dispositivo está sendo retirado da rede e que ele e seus serviços não estarão mais disponíveis.

O endereço de *multicast* padrão, assim como os mecanismos de anúncio, busca e remoção de dispositivos são definidos pelo *Simple Service Discovery Protocol* (SSDP).

O SSDP é usado sobre HTTPMU, responsável pelo *multicast* sobre UDP, e HTTPU, responsável por *unicast* sobre UDP.

## Descrição

No momento em que um ponto de controle descobre um dispositivo ele ainda sabe muito pouco sobre ele. Para que isto seja sanado, deve haver uma requisição da descrição através da URL passada ao ponto de controle no passo anterior, como ilustrado na Figura 2.3.

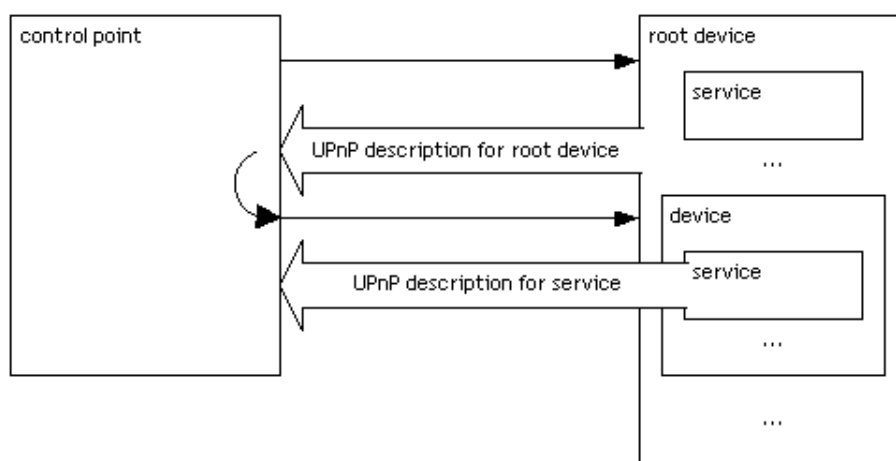


Figura 2.3: Etapa de Descrição

Um dispositivo UPnP pode conter outros dispositivos lógicos ou serviços. Estas informações estão contidas na descrição do dispositivo. A descrição UPnP para um dispositivo é expressa em XML e é composta de:

- Especificações do fabricante como modelo, número de série, nome e URL do fabricante;
- Uma lista de todos dispositivos embutidos ou serviços;
- URLs para controle, tratamento de eventos e apresentação.

Após receber a descrição do dispositivo, o ponto de controle pode fazer uma nova requisição para cada serviço ou dispositivo embutido (Figura 2.3). Para cada serviço, a descrição inclui uma lista de comandos ou ações, parâmetros ou argumentos e variáveis.

## Controle

Após a etapa de descrição, o ponto de controle pode enviar ações a um serviço de um dispositivo remoto. Para enviar uma ação, o ponto de controle envia uma mensagem de controle ao endereço contido na URL de controle previamente adquirida (*action request*,

na Figura 2.4). Uma mensagem de controle é enviada em XML, o protocolo usado é SOAP (*Simple Object Access Protocol*).

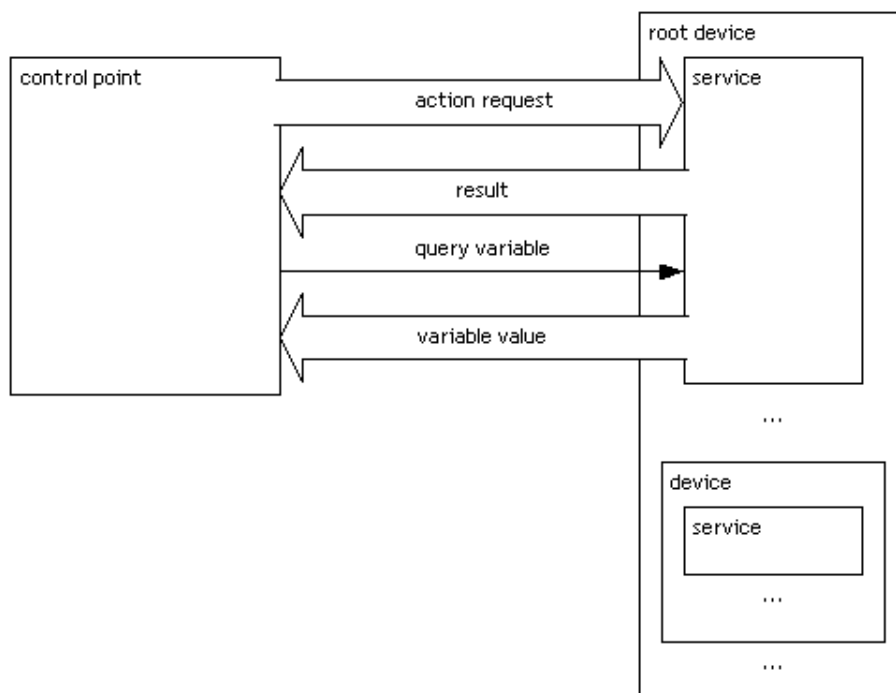


Figura 2.4: Etapa de Controle.

Como resposta a uma mensagem de controle, o serviço pode retornar algum resultado ou um erro. Uma chamada a um serviço também pode provocar uma alteração nas variáveis de estado do serviço. Uma alteração no valor destas variáveis provoca a emissão de eventos para todos os pontos de controles interessados, como descrito na Seção 2.1.

Caso um ponto de controle deseje determinar o estado de alguma variável de estado ele pode requisitar o valor desta variável, o serviço então fornece o valor da variável (*query variable* na Figura 2.4).

Enquanto a publicação feita na descoberta ainda estiver válida o ponto de controle pode assumir que o dispositivos e seus serviços ainda estão disponíveis.

### Tratamento de Eventos

Após um ponto de controle ter descoberto um dispositivo (seção 2.1) e recebido a sua descrição (seção 2.1), o ponto de controle possui as informações essenciais para o tratamento de eventos. A descrição UPnP para um serviço inclui uma lista das ações e uma lista das variáveis que modelam o serviço em tempo de execução.

Uma mensagem especial é enviada assim que o ponto de controle se une à rede. Esta mensagem contém os nomes e valores de todas as variáveis que podem ser tratadas, permitindo a inicialização do modelo de estado do serviço. Em uma rede com múltiplos

pontos de controle, o tratamento de eventos é destinado a manter todos os pontos de controle igualmente informados sobre os efeitos de qualquer ação.

O serviço publica, via mensagens, as atualizações quando as variáveis são alteradas. Qualquer ponto de controle pode observar esta mudança. Estas mensagens contém o nome e o valor atual da variável ou variáveis. As mensagens são expressas em XML e formatadas pelo GENA (*General Event Notification Architecture*).

Para poder observar os eventos, um ponto de controle pode “assinar” os eventos de um determinado serviço, isto é, o ponto de controle se tornará um observador do estado daquele serviço e sempre que uma alteração numa variável de estado do serviço ocorrer o ponto de controle será informado sobre essa mudança. Isto é feito através de uma mensagem (*subscription message*, na Figura 2.5).

Periodicamente a requisição deve ser renovada (*renewal request*, na Figura 2.5) de forma que, caso o ponto de controle ou o serviço venha a ser retirado de forma inesperada da rede, a rede consiga identificar que aquele componente não está mais disponível.

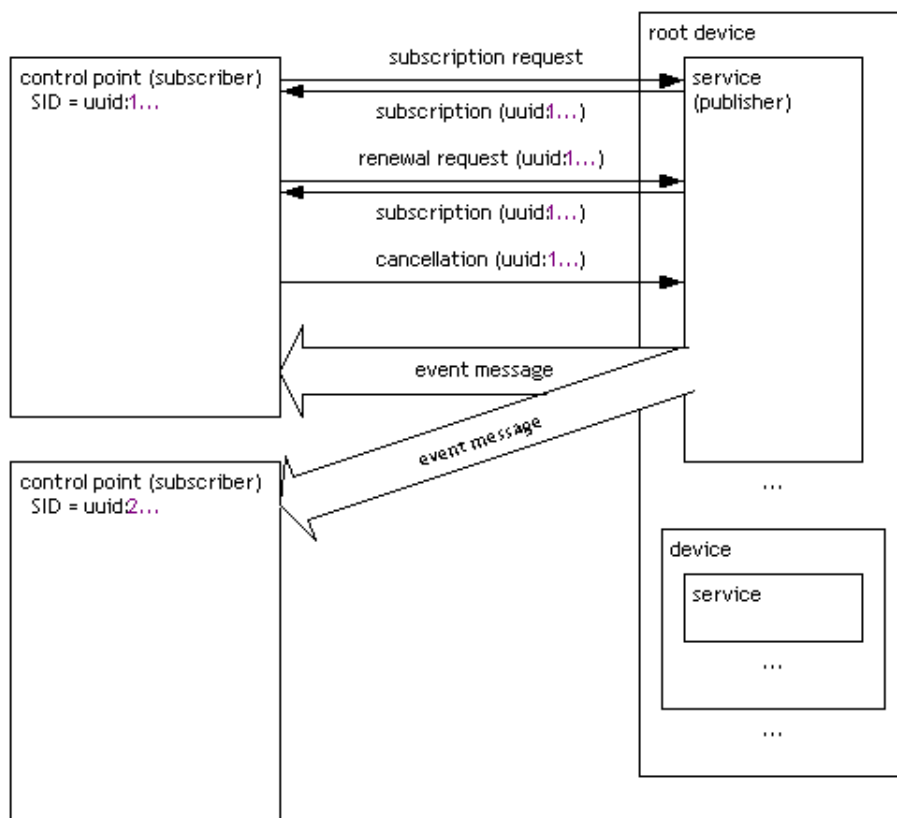


Figura 2.5: Etapa de Tratamento de Eventos.

No momento em que o ponto de controle não necessita mais dos dados de um determinado serviço ele pode mandar uma mensagem de cancelamento (*cancellation* na Figura 2.5)). Desta forma o ponto de controle não irá mais receber eventos do serviço.

## Apresentação

Um dispositivo pode ter uma URL para apresentação. Caso isso ocorra, o ponto de controle pode carregar esta URL em um *browser* (Figura 2.6) e dependendo das capacidades da página, ela permite ao usuário controlar o dispositivo e/ou ver o status do mesmo. O grau de controle permitido depende das capacidades específicas da página de apresentação e do dispositivo.

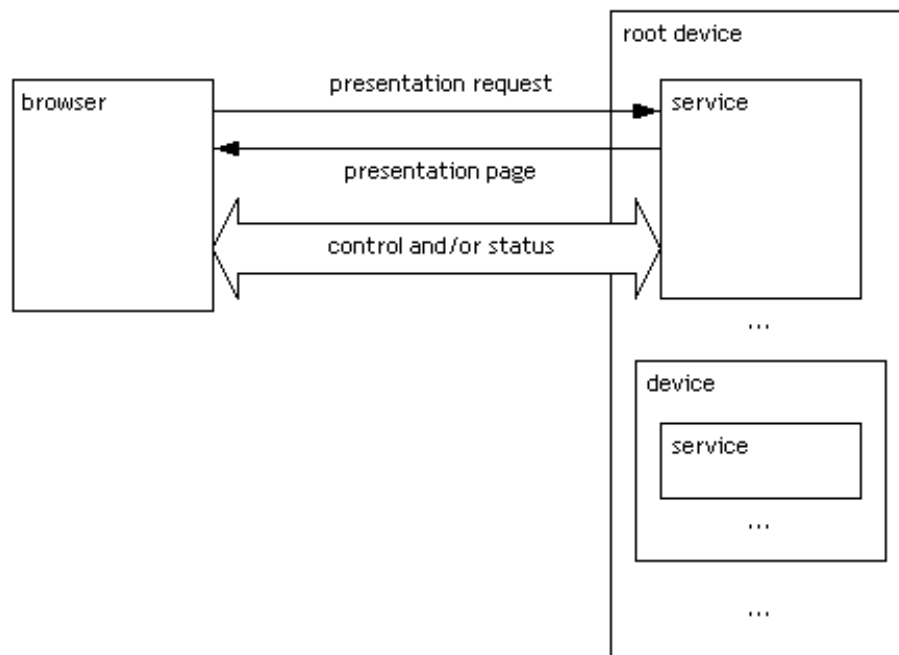


Figura 2.6: Etapa de Apresentação.



## 2.2 Outros Protocolos de Descoberta de Serviços

Durante a fase inicial deste trabalho, outros protocolos de descoberta e disponibilização de serviços foram estudados. Estes protocolos apresentam diferentes soluções para localização e disponibilização de serviços.

### Zeroconf

Zeroconf ou *Zero Configuration Network* [14] é um conjunto de técnicas que permitem a criação de uma rede IP de forma automática sem a configuração de servidores específicos. Isto permite que usuários sem conhecimento técnico conectem computadores, impressoras e outros dispositivos em rede de forma que estes possam ser utilizados.

A especificação do padrão é feita pelo *IETF Zeroconf Working Group* criado em setembro 1999. Em maio de 2002 uma versão de Zeroconf foi lançada pela Apple com o nome comercial de Rendezvous. Em agosto do mesmo ano é lançada uma versão para Windows, e, em abril de 2005, Apple anuncia "Rendezvous 2" com um novo nome comercial, Bonjour. Bonjour inclui APIs para Java, além de uma versão completa para Windows [14].

O IETF apresenta como requisitos para o Zeroconf [15]:

- Alocação automática de endereços sem a presença de um servidor DHCP (*Link-Local Addressing*).
- Tradução entre nomes e IP sem a presença de um servidor DNS (*Multicast DNS*)
- Localizar serviços, como impressoras, sem um servidor de diretório (*DNS Service Discovery*)

Um outro requisito é que as soluções acima devem coexistir em redes já configuradas. Os protocolos Zeroconf não podem causar dano à rede quando uma máquina Zeroconf é conectada a uma rede maior.

Zeroconf e UPnP possuem algumas características semelhantes em termos de configuração de rede, obtendo um IP de maneira automática quando o dispositivo é adicionado na rede, ao contrário de Jini que presume a existência de um mecanismo de configuração de rede já definido, como DHCP por exemplo. Zeroconf e UPnP diferem porém no fato de Zeroconf ser focado em serviços e UPnP em dispositivos.

## Jini

A tecnologia de rede Jini [16] é uma arquitetura aberta que permite aos desenvolvedores em Java criar serviços centrados em rede facilmente adaptados a mudanças. Jini pode ser usada para criar sistemas com escalabilidade e flexibilidade.

Jini foi originalmente criada pela Sun Microsystems, tendo contribuições da comunidade de desenvolvedores Jini desde 1999 [17]. O código fonte de Jini é disponível sob a *Sun Community Source License* (SCSL).

Jini define uma arquitetura distribuída baseada na idéia de grupos federados de usuários e de recursos requisitados por estes usuários. O objetivo geral é transformar a rede em uma ferramenta flexível e facilmente administrável, onde os recursos possam ser encontrados pelos clientes. Estes recursos podem ser implementados tanto por dispositivos de hardware como por programas ou uma combinação de ambos [18].

Jini é construída sobre Java de forma que os objetos Java possam circular pela rede, estendendo os benefícios da programação orientada a objetos à rede. Desta forma, os objetos podem interagir entre si através de interfaces bem definidas.

Jini e UPnP modelam os serviços e dispositivos de maneira bastante semelhante, usando a idéia de objetos. Jini porém assume que estes objetos sejam objetos Java, desta forma atrelando o sistema a existência de máquinas virtuais Java (JVM). UPnP, por outro lado, apenas define um protocolo comum, que pode ser implementado em qualquer linguagem.

## Capítulo 3

# Sistemas de Recomendação

A grande quantidade de informação disponível através da Internet e outras mídias como TV, cinema e música, podem fazer com que os usuários dessas mídias fiquem sobrecarregado com o excesso de informação, tornando difícil encontrar um item que seja de sua preferência dentre tantos disponíveis. Muitas vezes as pessoas utilizam mecanismos para filtrar toda essa informação e procurar o que lhe seja mais relevante segundo suas preferências. Um destes mecanismos é a recomendação. Por exemplo, uma pessoa pode pedir a opinião sobre um determinado item (música, livro, filme, etc.) a outras pessoas com gostos semelhantes aos seus. Os Sistemas de Recomendação [4–6] são uma tentativa de capturar esta e outras formas de interação entre as pessoas e automatizar este processo, de forma a disponibilizar este serviço a um grande número de usuários para um grande número de itens.

De maneira geral, os sistemas de recomendação têm como objetivo ajudar o usuário na escolha entre um número muito grande de itens disponíveis. As técnicas de recomendação geralmente consistem em calcular a provável nota que um usuário daria para um determinado item. Esta nota reflete a satisfação do usuário em relação a aquele item.

Os estudos sobre os sistemas de recomendação levaram ao surgimento de dois paradigmas principais: a Filtragem Baseada em Conteúdo (FBC) e a Filtragem Colaborativa (FC) [3,19]. Na Filtragem Baseada em Conteúdo, a avaliação do item a ser recomendado é feita através da comparação do conteúdo deste item com os itens que o usuário avaliou no passado. A Filtragem Colaborativa leva em conta as opiniões dos outros usuários do sistema sobre aquele item, sendo as opiniões mais relevantes as dos usuários com preferências mais parecidas com a do usuário que procura a recomendação. Uma outra abordagem é a utilização de uma combinação das duas técnicas chamada de Filtragem Híbrida.

### 3.1 Filtragem Baseada em Conteúdo

A Filtragem Baseada em Conteúdo recomenda itens ao usuário com base nos itens para os quais ele tenha indicado satisfação no passado [19]. A técnica consiste em obter uma descrição dos itens e compará-las com o perfil do usuário de forma a determinar se este item lhe é relevante. O perfil do usuário pode ser criado a partir de um questionário, onde o usuário responde sobre suas preferências. O perfil também pode ser melhorado a partir do registro dos itens que são escolhidos pelo usuário.

Cada item é descrito por um vetor  $D = (w_1, w_2, \dots, w_k)$  com  $k$  elementos [20, 21], onde cada elemento é definido a partir da análise do conteúdo do item. No caso de um texto, podem ser usadas as palavras chaves, no caso de um arquivo de música, podem ser extraídas informações como artista, estilo ou nome do álbum. Cada elemento  $w_i$  indica o peso do elemento  $i$  de acordo com a relevância deste elemento em relação ao item e ao conjunto dos outros itens disponíveis no sistema. Este vetor é comparado com o perfil do usuário de forma a obter a relevância do item para o usuário.

O cálculo dos elementos  $w_i$  do vetor de descrição do item  $D$  é feito utilizando o algoritmo *tf-idf*:

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (3.1)$$

$$idf_i = \log \frac{n}{df_i} \quad (3.2)$$

$$w_i = tf_i \cdot idf_i \quad (3.3)$$

onde  $tf_i$  é a frequência de ocorrência do termo  $i$  no item  $D$ .  $idf_i$  é a frequência com que o termo  $i$  ocorre em todos os itens disponíveis no sistema.

Assim como os itens, o perfil do usuário é representado por um vetor  $P = (u_1, u_2, \dots, u_l)$ , com  $l$  elementos, onde cada termo  $u_i$  mede a relevância da característica  $i$  para o usuário. O perfil do usuário é alimentado pela opinião que o usuário fornece sobre um dado item. Esta opinião pode ser obtida de forma explícita, onde o usuário vota no item, ou de forma implícita, o sistema automaticamente obtém a opinião do usuário através da utilização do sistema.

A atualização do perfil do usuário é feita através da equação:

$$P' = \alpha P + \beta D \quad (3.4)$$

onde as constantes  $\alpha$  e  $\beta$  definem quanto o perfil anterior  $P$  e a descrição do documento avaliado  $D$  influenciam no perfil atualizado  $P'$ .

A relevância do item para o usuário é calculada através da similaridade entre o vetor do item e o vetor do perfil do usuário. A similaridade pode ser obtida através da medida do cosseno entre os vetores  $P$  e  $D$  dada por:

$$\text{sim}(D, P) = \frac{D \cdot P}{\|D\| \cdot \|P\|} = \frac{\sum_k u_k w_k}{\sqrt{\sum_k u_k^2 \sum_k w_k^2}} \quad (3.5)$$

A Filtragem Baseada em Conteúdo pode ter sua eficiência comprometida caso o conteúdo dos itens não possa ser identificado facilmente ou não esteja disponível. Outro problema acontece quando um item é muito diferente dos outros itens existentes no sistema, pois não existirão recomendações para itens com conteúdo semelhante [6].

## 3.2 Filtragem Colaborativa

Filtragem Colaborativa é uma abordagem utilizada em sistemas de recomendação baseada na avaliação dos itens por parte dos usuários [5]. Ela foi inicialmente desenvolvida de forma a suprir necessidades deixadas pela deficiência da Filtragem Baseada em Conteúdo [5]. A Filtragem Colaborativa ignora os dados referentes à descrição do item e utiliza como base para as recomendações as opiniões dos usuários sobre um determinado item. A Filtragem Colaborativa se baseia no fato de encontrar pessoas com gostos semelhantes.

A Filtragem Colaborativa faz uso de um banco de dados contendo os perfis dos usuários. Cada perfil contém as opiniões de um usuário sobre um conjunto de itens. Estas opiniões são representadas por notas dadas a cada item. As notas podem ser obtidas de maneira explícita ou implícita, isto é, o sistema pode requisitar ao usuário que este atribua uma nota ao item ou pode inferir a nota com base na utilização do sistema por parte do usuário.

Os perfis dos usuários são comparados uns com os outros de forma a identificar pessoas com gostos semelhantes. Quando a recomendação para um item é requisitada para um usuário, chamado de usuário ativo, as opiniões dos usuários com preferências semelhantes são usadas para determinar a recomendação.

A recomendação é calculada como o valor da nota que provavelmente o usuário ativo atribuirá ao item, com base na opinião dos outros usuários sobre aquele item. O valor da recomendação  $p_{a,j}$  para o usuário ativo  $a$  do item  $j$ , é calculado por uma soma ponderada das notas dos outros usuários através da equação 3.6 [22]:

$$p_{a,j} = \bar{v}_a + k \sum_{i=1}^n w(a, i)(v_{i,j} - v_i) \quad (3.6)$$

onde,  $n$  é o número de usuários com peso  $w(a, i)$  diferente de zero,  $k$  é um fator de normalização tal que o valor absoluto da soma dos pesos seja igual a unidade, e a nota média  $\bar{v}_i$  do conjunto dos itens  $I_i$  do perfil do usuário  $i$  é calculada por:

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_j} v_{i,j} \quad (3.7)$$

Os pesos  $w(a, i)$  são uma medida da similaridade entre dois usuários e podem ser calculados através do coeficiente de Pearson:

$$w(a, i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (3.8)$$

O coeficiente de Pearson foi definido inicialmente no contexto do projeto GroupLens [22, 23], sendo a maneira mais comumente utilizada para calcular os pesos [6].

Os principais problemas da Filtragem Colaborativa são [19, 24]:

**Problema do novo usuário** Como um usuário novo ainda não fez ou fez poucas avaliações de itens, o sistema ainda não é capaz de determinar suas preferências de maneira eficiente.

**Problema do novo item** Se um novo item for adicionado à base de dados, ele não poderá ser recomendado até que alguma informação sobre ele seja obtida através da recomendação de outros usuários.

**A qualidade depende do tamanho do histórico** Assim como no caso do novo usuário e do novo item, um sistema com um pequeno histórico gera recomendações menos eficientes.

**Problema da “Ovelha negra”** Se um usuário tem preferências extremamente discrepantes dos outros usuários, não existirão usuários com preferências semelhantes nos quais basear a recomendação.

### 3.3 Filtragem Híbrida

A Filtragem Híbrida é uma tentativa de minimizar os problemas inerentes à Filtragem Baseada em Conteúdo e à Filtragem Colaborativa. A Filtragem Híbrida é na verdade uma combinação das duas técnicas anteriores com o objetivo de aproveitar o que cada uma tem de melhor. Como a Filtragem Colaborativa independe do conteúdo do item a ser avaliado, ela se torna mais eficiente do que a Filtragem Baseada em Conteúdo quando são tratados itens com conteúdos que diferem muito entre si, ou dos quais não se pode obter

uma descrição mais detalhada. Por outro lado, a Filtragem Baseada em Conteúdo, por avaliar os itens a partir de sua descrição, torna-se mais eficiente quando são tratados itens novos no sistema ou de usuários com gostos incomuns [6]. Um sistema de recomendação criado a partir destas duas técnicas, entretanto, ainda depende da existência de um banco de dados contendo as opiniões do usuário para o qual se deseja fazer a recomendação. Contudo, é possível em ambos os casos criar um perfil inicial do usuário de forma a provê-lo com recomendações iniciais que vão melhorando à medida em que ele utiliza o sistema e faz novas avaliações.

Em um sistema de recomendação híbrido, a recomendação pode ser feita diretamente através da escolha entre um dos resultados obtidos através de outras técnicas. Esta escolha é feita através de algum critério definido pelo sistema. Por exemplo um sistema pode tentar fazer a recomendação através da Filtragem Colaborativa, caso não seja possível, o sistema apresenta o resultado da Filtragem Baseada em Conteúdo [24].

Outra metodologia consiste em combinar os resultados obtidos por outras técnicas. Neste caso os resultados podem ser apresentados simultaneamente, porém separados, ou podem ser agrupados como uma única recomendação, neste caso é possível definir qual dos resultados é mais relevante de forma a enfatizar uma das técnicas utilizadas.

Também é possível combinar diretamente duas técnicas, de forma que os resultados de uma influenciam diretamente no resultado de outra. O resultado de uma técnica pode, por exemplo, ser usado para reduzir o número de itens a serem avaliados antes de se utilizar outra técnica [24].

### 3.4 Exemplos de Sistemas de Recomendação

As técnicas de recomendação descritas até aqui são aplicadas em diversas aplicações. Nesta seção são apresentados alguns exemplos de sistemas de recomendação que utilizam essas técnicas.

Schafer *et al* [25] apresentam diversas empresas de comércio eletrônico e como elas usam os sistemas de recomendação como ferramenta de marketing para aumentar as suas vendas. Segundo o trabalho, os sistemas de recomendação aumentam as vendas de três maneiras: fazendo com que visitantes ocasionais do site se tornem compradores; aumentando a venda cruzada, isto é, um usuário que apenas estava interessado em um produto acaba comprando outro; e criando fidelidade do cliente com a loja. Empresas de comércio eletrônico que utilizam sistemas de recomendação também são apresentadas por Schafer *et al* em [26] e [25], como por exemplo: Amazon.com, CDNOW, eBay, Levis, Reel.com e Drugstore.com. A Amazon.com, por exemplo, utiliza um sistema de recomendação híbrido baseado no histórico de compras e na avaliação dos produtos por parte do usuário.

WikiLens [27] é um software que permite a qualquer usuário criar um sistema de recomendação para um tipo de item qualquer. Os sistemas de recomendação criados são mantidos pela própria comunidade de usuários do WikiLens.

Cazella [6] apresenta o W-RECMAS, um sistema de recomendação híbrido que leva em consideração a relevância da opinião dos usuários, para recomendar artigos científicos.

Ringo um sistema de recomendação para músicas, é apresentado por Shardanand *et al* em [28]. No mesmo trabalho, é feita uma comparação entre quatro algoritmos de recomendação.

No contexto educacional Reategui *et al* [29] utilizam um sistema de recomendação como componente do Endowing, um personagem virtual que pode ser usado em ambientes educacionais. Tendo a capacidade de obter informações dos alunos, o personagem utiliza estas informações para alimentar um sistema de recomendação para recomendar conteúdos a serem estudados e atividades a serem desenvolvidas pelos alunos.

Os exemplos acima mostram áreas de pesquisa em que os sistemas de recomendação podem ser aplicados. Porém não foram encontrados trabalhos que tentam utilizar esta técnica em associação com os protocolos de descoberta e disponibilização de serviços com o objetivo de filtrar os serviços mais relevantes ao usuário.



# Capítulo 4

## Arquiterura do Sistema de Recomendação para Dispositivos UPnP

Neste capítulo é apresentada a arquitetura para a habilitação de recomendação de serviços em redes baseadas na tecnologia UPnP. A Seção 4.1 apresenta uma visão geral da arquitetura. As Seções 4.2 e 4.3 descrevem os dois componentes principais da arquitetura, respectivamente, o Servidor de Recomendação e o *Control Point* com suporte a recomendação.

### 4.1 Visão Geral

A arquitetura descrita neste trabalho, ilustrada na Figura 4.1, é composta basicamente por um Servidor de Recomendação, um ou mais *Control Points* UPnP com a capacidade de se comunicar com o Servidor de Recomendação e os dispositivos UPnP. Em uma rede UPnP o *Control Point* tem como funções: obter a descrição dos dispositivos e dos serviços por eles providos, fazer as chamadas aos serviços e monitorar o estado dos dispositivos. A forma com que os *Control Points* interagem com os dispositivos UPnP se mantém inalterada em relação à especificação do padrão UPnP.

Quando um *Control Point* é ativado ele faz uma busca na rede por dispositivos UPnP (1 na Figura 4.1). Geralmente, um *Control Point* convencional apresenta todos os dispositivos que ele pode controlar. O *Control Point* apresentado neste trabalho, antes de apresentar os dispositivos ao usuário, submete a lista dos dispositivos encontrados ao Servidor de Recomendação (2 na Figura 4.1). O Servidor retorna uma lista dos dispositivos ordenada com base na recomendação feita por ele (3 na Figura 4.1). Esta lista é direcionada ao usuário que fez a requisição (4 na Figura 4.1), o *Control Point* deve então

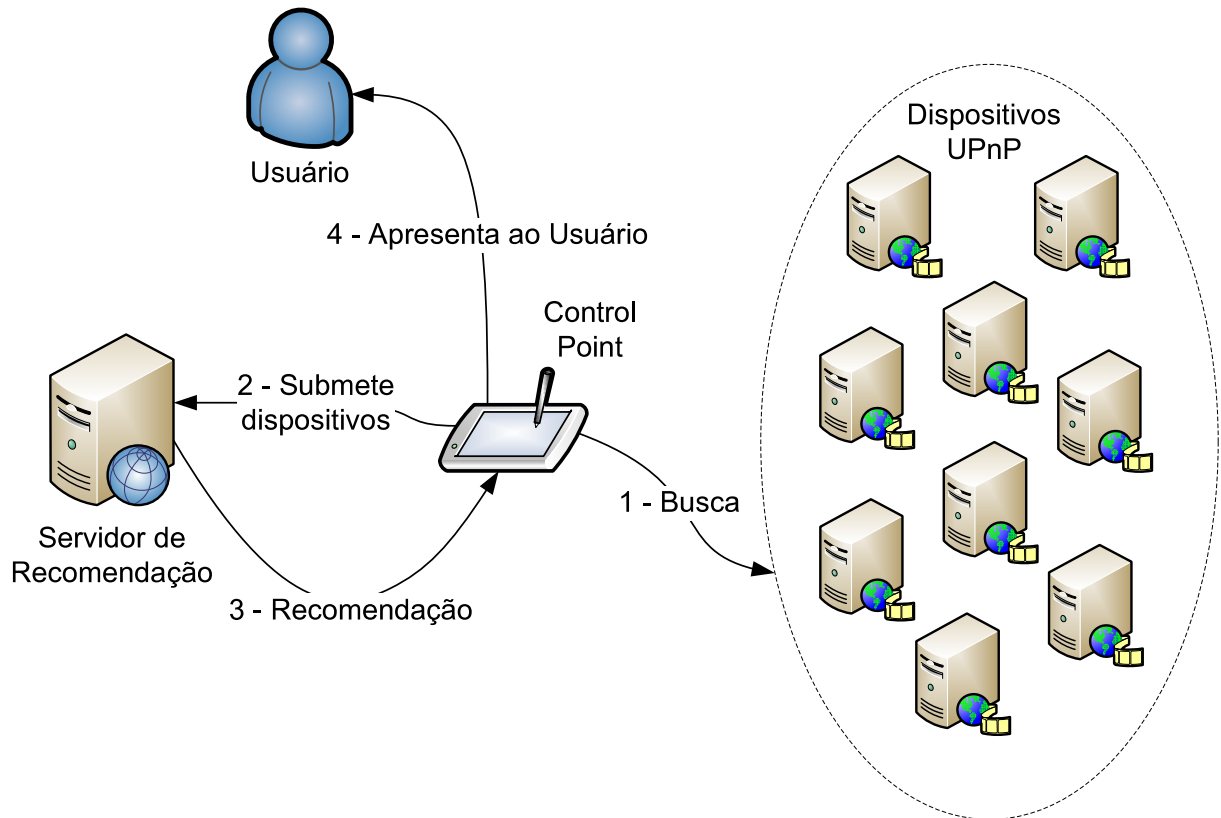


Figura 4.1: Integração de um Servidor de Recomendação a uma rede UPnP

autenticar o usuário no servidor antes de fazer a requisição. A lista leva em consideração as estatísticas de utilização dos dispositivos por parte do próprio usuário autenticado e dos outros usuários que utilizaram o sistema anteriormente.

Nas seções seguintes a arquitetura aqui apresentada é descrita de forma mais detalhada.

## 4.2 Servidor de Recomendação

O Servidor de Recomendação (*Recommender Server*) é o elemento da arquitetura responsável por, a partir da lista de dispositivos encontrados pelo *Control Point*, fornecer as recomendações de dispositivos para o usuário. O Servidor de Recomendação também é responsável por receber, armazenar e processar as informações referentes a utilização dos dispositivos UPnP por parte dos usuários. A partir destas informações o Servidor de Recomendação cria o perfil de cada usuário. A arquitetura responsável pela disponibilização destas funcionalidades é ilustrada na Figura 4.2.

Os módulos que compõem o Servidor de Recomendação são:

- **Hybrid Recommender** Um sistema de recomendação híbrido.

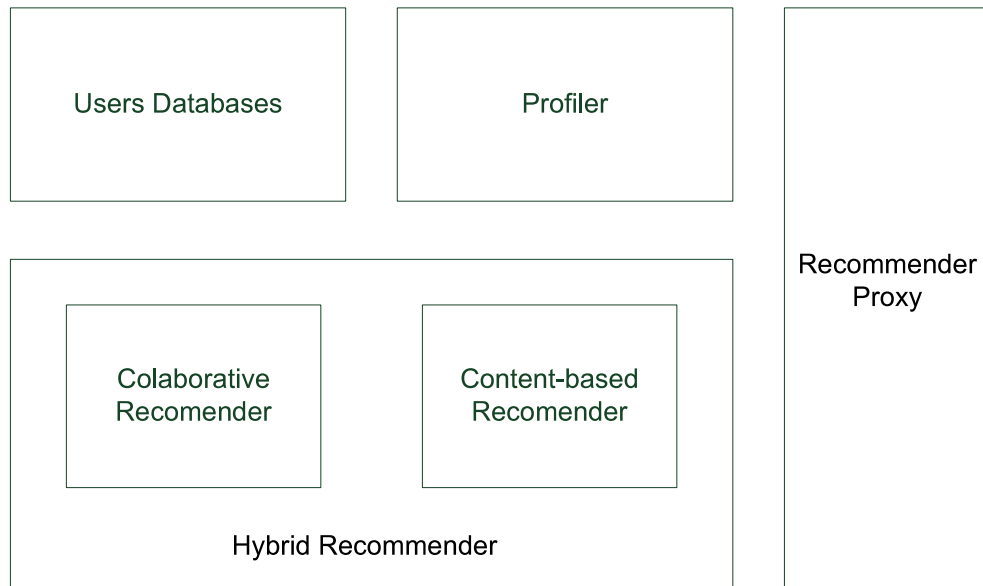


Figura 4.2: Arquitetura do Servidor de Recomendação

- **Profiler** Coleta as informações de utilização dos dispositivos UPnP por parte do usuário e as armazena nos bancos de dados.
- **Recommender Facade** Faz a interface entre o Servidor de Recomendação e os *Control Points*.
- **Users Databases** Bancos de dados contendo as informações de históricos e perfis dos usuários.

Nas subseções seguintes estes itens são descritos mais detalhadamente.

#### 4.2.1 *Hybrid Recommender*

O módulo *Hybrid Recommender* é o elemento principal do Servidor de Recomendação. Ele utiliza as estatísticas de uso e os perfis dos usuários, criados pelo *Profiler* e armazenados nos *Users Databases*, para gerar recomendações para os dispositivos disponíveis na rede UPnP e as disponibiliza através da *Recommender Facade*. O *Hybrid Recommender* consiste de um Sistema de recomendação Híbrido que utiliza Filtragem Colaborativa e Filtragem Baseada em Conteúdo conforme descrito na Seção 3.3.

O *Hybrid Recommender* é composto por um Sistema de Recomendação Colaborativo (*Colaborative Recommender*) e um Sistema de Recomendação Baseado em Conteúdo (*Content-Bases Recommender*). Cada um dos sistemas gera a recomendação para dispositivos disponíveis na rede para o usuário ativo. Estas recomendações são combinadas e apresentadas ao usuário ativo.

O *Content-Bases Recommender* gera recomendações comparando as descrições dos dispositivos com o perfil do usuário como apresentado na Seção 3.1. A descrição de cada dispositivo é representada por um vetor  $D = (w_1, w_2, \dots, w_k)$  com  $k$  elementos, cada elemento do vetor representa a relevância (peso) de uma determinada característica do dispositivo em relação às outras características do mesmo dispositivo e às características de todos os outros dispositivos disponíveis na rede UPnP. O cálculo dos elementos do vetor de descrição do dispositivo  $D$  são realizados pelo *Profiler*.

As características levadas em consideração em um dispositivo UPnP para a determinação da recomendação através da Filtragem Baseada em Conteúdo são obtidas através do *Device Descriptor*, descrito na Subseção 4.3.6. Estas características variam de acordo com o dispositivo a ser recomendado, no caso dos componentes da *UPnP AV Architecture* [30] um *UPnP Media Server* [31] é caracterizado através do seu conteúdo de mídia (artistas, albums, músicas, filmes, gênero do filme, etc.), enquanto que um *UPnP Media Render* [32] pode ser caracterizado pelos tipos de mídia que é capaz de reproduzir.

O vetor que descreve o perfil do usuário  $P = (u_1, u_2, \dots, u_l)$  possui  $l$  elementos e cada elemento  $u_i$  representa a relevância da característica  $i$  que um determinado dispositivo UPnP possa possuir. Este vetor é criado pelo *Profiler* e é usado para a recomendação para um determinado item através da similaridade do cosseno calculada pela Equação 3.5.

O *Collaborative Recommender* faz as recomendações comparando o histórico do usuário com os históricos dos outros usuários cadastrados no sistema, como apresentado na Seção 3.2. O histórico de cada usuário é composto por uma lista contendo todos os dispositivos UPnP que já foram utilizados pelo usuário e o número de vezes que cada dispositivo foi utilizado. Cada dispositivo corresponde a um item, identificado através do UUID (*Universally Unique Identifier*), definido na Arquitetura UPnP [1] e a nota atribuída ao item é obtida de forma implícita, fazendo com que o valor da nota do dispositivo seja igual ao número de vezes que ele foi utilizado. A recomendação é então obtida calculando a nota que provavelmente o usuário dará a cada um dos dispositivos disponíveis na rede UPnP, isto é feito utilizando-se a Equação 3.6, onde os pesos são calculados através do coeficiente de Pearson definido na Equação 3.8.

Os resultados obtidos através da Filtragem Baseada em Conteúdo e da Filtragem Colaborativa são combinados pelo *Hybrid Recommender*. Os resultados de ambas as técnicas são agrupados como uma única recomendação que é apresentada ao usuário.

### 4.2.2 *Profiler*

O *Profiler* coleta as informações de utilização dos dispositivos UPnP por parte do usuário e as armazena nos bancos de dados. O *Profiler* é responsável por, a partir das características

do dispositivo obtidas pelo *Device Descriptor* e recebidas através do módulo *Recommender Facade*, gerar o vetor de descrição de um dispositivo, e atualizar os perfis de usuário no *Users Databases* cada vez que um dispositivo for usado por um usuário do sistema.

O vetor de descrição  $D$  de um dispositivo é determinado através do algoritmo *tf-idf* apresentado na Seção 3.1, onde cada um dos pesos  $w_i$  é calculado através das Equações 3.1, 3.2 e 3.3.

O *Profiler* é notificado pela *Recommender Facade* cada vez que um usuário ativo utiliza um dispositivo UPnP disponível na rede. O *Profiler* então faz a atualização do perfil do usuário no *Users Databases*, registrando o número de utilizações de um dispositivo e atualizando o vetor de descrição do perfil do usuário  $P$ . A atualização do vetor  $P$  é feita através da Equação 3.4.

Os perfis criados são usados pelo *Hybrid Recommender* para a determinação das recomendações para o usuário através da Filtragem Colaborativa, como descrito na Seção 3.2 e para a Filtragem Baseada em Conteúdo como mostrado na Seção 3.1. Os perfis criados pelo *Profiler* são armazenados nos *Users Databases*.

### 4.2.3 *Recommender Facade*

A *Recommender Facade* tem como função prover um ponto de acesso único através do qual os *Control Points* com Suporte a Recomendação de Dispositivos UPnP possam acessar o Servidor de Recomendação. A *Recommender Facade* fornece uma interface através da qual um *Control Point* utilizando a *Server Access Layer* pode solicitar recomendações ao *Hybrid Recommender* e alimentar o *Profiler* com estatísticas de uso dos dispositivos por parte do usuário ativo.

A *Recommender Facade* disponibiliza através de uma API de *webservices* os seguintes serviços:

```
recommend(username, auth, itens_to_evaluate)
```

Através de uma chamada a este serviço, o *Control Point* pode solicitar a recomendação dos dispositivos para o usuário ativo autenticado através de um nome de usuário `username` e um código de autenticação `auth`. A lista de todos os dispositivos disponíveis na Rede UPnP é enviada através do parâmetro `itens_to_evaluate`, esta lista de dispositivos contém o UUID e as características de cada dispositivo, como descrito anteriormente em 4.2.1.

```
submit_use(username, auth, item)
```

Através deste serviço o *Control Point* pode atualizar o *Recomender Server* quando um dispositivo é utilizado, de forma a atualizar o perfil do usuário ativo identificado através dos parâmetros `username` e `auth`.

```
batch_submit_use(username, auth, items)
```

Este serviço, assim como o `submit_use`, permite ao *Control Point* atualizar o perfil do usuário no *Recomender Server*. O serviço `batch_submit_use`, entretanto, permite que o registro de várias utilizações de dispositivos possam ser submetidas em uma única chamada, diminuindo o número de chamadas a *Recommender Facade*.

#### 4.2.4 Users Databases

O módulo *Users Databases* é um conjunto de bancos de dados contendo as informações de históricos e perfis dos usuários. As informações armazenadas neste bancos de dados são utilizadas pelo *Hybrid Recommender* para a determinação dos itens recomendados. Os bancos de dados são atualizados com as estatísticas de uso dos dispositivos por cada usuário pelo *Profiler*.

Os bancos de dados que compõem este módulo são apresentados nas tabelas 4.1 e 4.2.

Usuário	Dispositivo 1	Nota 1	...	Dispositivo n	Nota n
$username_1$	UUID <sub>11</sub>	$v_{11}$	...	UUID <sub>1n</sub>	$v_{1n}$
$username_2$	UUID <sub>21</sub>	$v_{21}$	...	UUID <sub>2n</sub>	$v_{2n}$
...	...	...	...	...	...
$username_k$	UUID <sub>k1</sub>	$v_{k1}$	...	UUID <sub>kn</sub>	$v_{kn}$

Tabela 4.1: Históricos dos usuários

Os Históricos dos usuários são utilizados na Filtragem Colaborativa, neste banco são armazenados, para cada usuário, os  $n$  UUIDs de todos os dispositivos UPnP por ele utilizados e submetidos ao *Recommender Server* e a nota  $v_n$  atribuída a cada um deles, dada pelo número de utilizações de cada um destes dispositivos pelo usuário.

Os Perfis dos usuários são utilizados na Filtragem Baseada em Conteúdo, neste banco são armazenados para cada um dos  $k$  usuários o peso  $u_n$  de de cada característica  $n$  na opinião do usuário.

Usuário	Característica 1	Peso 1	...	Característica n	Peso n
$username_1$	Característica <sub>11</sub>	$u_{11}$	...	Característica <sub>1n</sub>	$u_{1n}$
$username_2$	Característica <sub>21</sub>	$u_{21}$	...	Característica <sub>2n</sub>	$u_{2n}$
...	...	...	...	...	...
$username_k$	Característica <sub>k1</sub>	$u_{k1}$	...	Característica <sub>kn</sub>	$u_{kn}$

Tabela 4.2: Perfis dos usuários

### 4.3 *Control Point*

Um *Control Point* convencional tem como funções: obter a descrição dos dispositivos e dos serviços por eles providos, fazer as chamadas aos serviços e monitorar o estado dos dispositivos. O *Control Point* com Suporte a Recomendação de Dispositivos UPnP é o elemento da arquitetura responsável pelas funções de um *Control Point* convencional, além de: obter uma descrição do dispositivo contendo as informações relevantes à Filtragem Baseada em Conteúdo; submeter os dispositivos encontrados e suas descrições para o servidor de recomendação e receber a lista de dispositivos recomendados e apresentá-la ao usuário. O *Control Point* com Suporte a Recomendação também é responsável por monitorar as atividades do usuário e alimentar o Servidor de Recomendação com as estatísticas de utilização dos dispositivos. A arquitetura responsável pela disponibilização destas funcionalidades é mostrada na Figura 4.3.

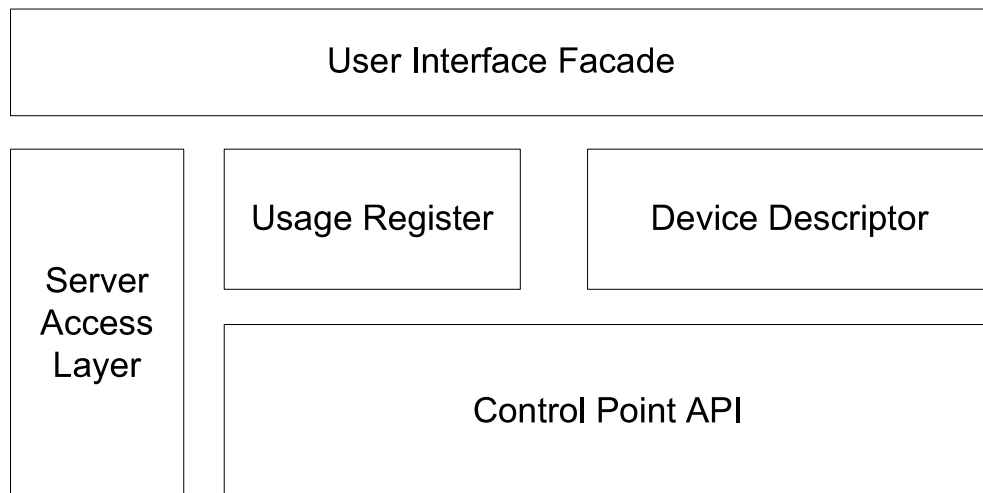


Figura 4.3: Arquitetura do *Control Point* com Suporte a Recomendação de Dispositivos UPnP

Os componentes principais do *Control Point* com Suporte a Recomendação são:

- ***Control Point* API** Permite acesso às funcionalidades do *Control Point* UPnP.

- **Control Point Manager** Gerencia o *Control Point* com Suporte a Recomendação de Dispositivos UPnP.
- **Server Access Layer** Permite o acesso ao servidor de recomendação.
- **User Interface Facade** Permite que a Interface com o usuário se comunique com o *Control Point* com Suporte a Recomendação.
- **Usage Register** Monitorar as atividades do usuário e alimentar o Servidor de Recomendação com as estatísticas de utilização dos dispositivos.
- **Device Descriptor** Obtém a descrição dos item disponíveis.

Nas Subseções seguintes estes itens são descritos mais detalhadamente.

### 4.3.1 *Control Point API*

A *Control Point API* é responsável pelo serviço de rede UPnP ela possui todas as funcionalidades de um *Control Point* convencional e as disponibiliza para os outros módulos do *Control Point* com Suporte a Recomendação de Dispositivos UPnP.

### 4.3.2 *Control Point Manager*

O *Control Point Manager* é o elemento central do *Control Point* com Suporte a Recomendação de Dispositivos UPnP. O *Control Point Manager* obtém a lista dos dispositivos disponíveis na rede através da *Control Point API*, obtém as estatísticas de uso e as descrições dos dispositivos da rede através do *Usage Register* e do *Device Descriptor* e as submete ao Servidor de Recomendação através da *Server Access Layer*. Ele também é responsável por enviar as informações necessárias para o usuário através da *User Interface Facade*.

### 4.3.3 *Server Access Layer*

A *Server Access Layer* permite o acesso ao servidor de recomendação através da *Recommender Facade*. Ela é responsável por receber as chamadas provenientes do *Control Point Manager*, chamar os serviços apropriados da *Recommender Facade* e enviar os resultados de volta para o *Control Point Manager*.



#### 4.3.4 *User Interface Facade*

A *User Interface Facade* permite que a interface com o usuário se comunique com o *Control Point* com Suporte à Recomendação. A *User Interface Facade* é uma camada que abstração que fornece os mecanismos necessários para que a Interface Gráfica com o Usuário (*Graphical User Interface* ou GUI) acesse e controle os dispositivos UPnP disponíveis para o *Control Point* e apresente os dispositivos recomendados entre os dispositivos disponíveis.

#### 4.3.5 *Usage Register*

O *Usage Register* é responsável por monitorar as atividades do usuário e alimentar o Servidor de Recomendação com as estatísticas de utilização dos dispositivos. O *Usage Register* intercepta as chamadas provenientes do *Control Point Manager*, registra estas chamadas, verificando quais devem ser consideradas interações que denotam interesse do usuário por um determinado dispositivo e as envia para o *Recommender Server* através do *Control Point Manager*.

#### 4.3.6 *Device Descriptor*

O *Device Descriptor* obtém a descrição dos item disponíveis na rede UPnP. As descrições são utilizadas pelo *Recommender Server* na determinação de recomendações com base na Filtragem Baseada em Conteúdo. Conforme o tipo de dispositivo a ser recomendado, as características obtidas podem variar, como dito anteriormente na Subseção 4.2.1. O *Item Descriptor* recebe do *Control Point Manager* a lista de dispositivos que precisam de uma descrição, ele então utiliza a *Control Point API* para explorar os dispositivos através de seus serviços UPnP em busca de informações relevantes a sua descrição.

### 4.4 Funcionamento

Como dito no início deste Capítulo, a arquitetura para disponibilização de recomendação de serviços em redes baseadas na tecnologia UPnP definida neste trabalho, é composta por um Servidor de Recomendação, um ou mais *Control Points* UPnP com a capacidade de se comunicar com o Servidor de Recomendação e os dispositivos UPnP. Os módulos que compõem o Servidor de Recomendação e os *Control Points* são apresentados nas Seções 4.2 e 4.3. Concluindo a definição desta arquitetura, nesta Seção é apresentado como os módulos descritos anteriormente interagem entre si de forma a prover a recomendação de dispositivos UPnP para o usuário.

### 4.4.1 Busca

A Busca por dispositivos UPnP disponíveis na rede é o procedimento inicial para a recomendação de dispositivos. A etapa de busca é ilustrada na Figura 4.4, seguindo os seguintes passos:

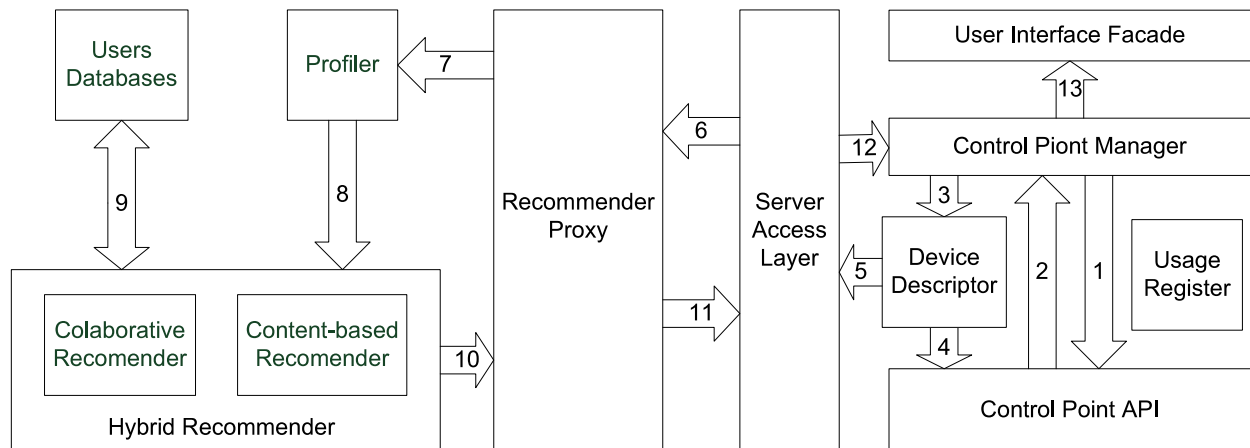


Figura 4.4: Inicialização e busca por dispositivos UPnP.

1. O *Control Point Manager* solicita à *Control Point API* a lista dos dispositivos UPnP existentes na rede.
2. A *Control Point API* faz uma busca por dispositivos na rede, como descrito nas etapas de Descoberta e Descrição na rede UPnP, apresentadas na Seção 2.1. A *Control Point API* então informa a lista de dispositivos encontrados ao *Control Point Manager*.
3. Conhecendo a lista de dispositivos UPnP disponíveis na rede, o *Control Point Manager* pede a descrição de cada um destes dispositivos ao *Device Descriptor*.
4. Através das ações dos dispositivos UPnP o *Device Descriptor* coleta as informações relevantes para a Filtragem Baseada em Conteúdo. O *Device Descriptor* obtém as características dos dispositivos acessando as ações disponíveis nos serviços dos dispositivos upnp através da *Control Point API*, como mostrado na etapa de Controle apresentada na Seção 2.1.
5. Com a lista de Dispositivos e suas características o *Device Descriptor* envia as informações para o *Recommender Server* através da *Server Access Layer*.
6. A *Server Access Layer* envia a lista dos dispositivos e suas características para o *Recommender Server* através de uma chamada remota para a *Recommender Facade*.

7. A *Recommender Facade* envia os itens e suas características para o *Profiler*, que gera o vetor de descrição  $D$  para cada dispositivo.
8. O *Profiler* envia a lista de dispositivos e suas descrições para o *Hybrid Recommender*.
9. O *Hybrid Recommender* usa os dados armazenados nos *Users Databases* e as informações sobre os dispositivos recebidas da *Recommender Facade* para calcular as recomendações.
10. O *Hybrid Recommender* retorna então a lista dos dispositivos ordenada de acordo com o resultado da recomendação para a *Recommender Facade*.
11. A *Recommender Facade* envia o resultado para a *Server Access Layer*.
12. A *Server Access Layer* repassa o resultado para o *Control Point Manager*.
13. O *Control Point Manager* disponibiliza a recomendação para a *User Interface Facade*.

### 4.4.2 Utilização

A etapa de Utilização ocorre quando o usuário começa a utilizar os dispositivos UPnP disponíveis na rede. Enquanto o usuário faz a utilização dos dispositivos, o *Control Point* com Suporte a Recomendação atualiza o *Recommemnder Server* com as estatísticas de uso do dispositivo por parte do usuário. A etapa de Utilização é ilustrada na Figura 4.5, seguindo os seguintes passos:

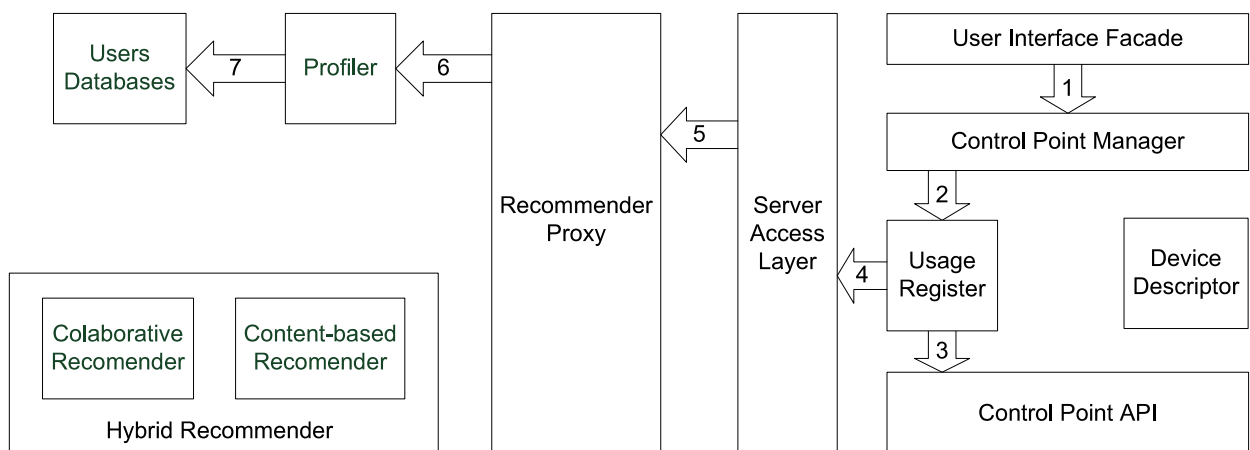


Figura 4.5: Utilização e alimentação dos perfis de usuário.

1. Através da *User Interface Facade* o usuário ativo escolhe qual dispositivo dentre os disponíveis ele vai controlar.

2. Cada vez que o usuário faz o uso de uma das Ações existentes nos Serviços disponibilizados pelo Dispositivo UPnP, o *Control Point Manager* informa ao *Usage Register*.
3. O *Usage Register* repassa a chamada para *Control Point API*, que aciona o serviço desejado.
4. O *Usage Register* verifica qual Serviço foi utilizada e se esta utilização deve ser considerada como um registro da utilização do serviço para o sistema de recomendação. Se a ação realizada pelo usuário denotar interesse na utilização daquele dispositivo (o usuário mandou o Media Render reproduzir uma música, por exemplo), *Usage Register* envia o registro da utilização do dispositivo através da *Server Access Layer*, de forma a atualizar o perfil do usuário no *Recommender Server*.
5. A *Server Access Layer* envia o registro da utilização para o *Recommender Server* através da *Recommender Facade*.
6. A *Recommender Facade* então passa o registro da utilização para o *Profiler*.
7. O *Profiler* faz a atualização do perfil do usuário no *Users Databases*, registrando o número de utilizações do dispositivo UPnP e atualizando o perfil do usuário ativo no sistema.

### 4.4.3 Novo dispositivo UPnP entra na rede

Quando um novo dispositivo UPnP entra na rede, ele é adicionado na lista de dispositivos disponíveis e o *Recommender Server* calcula a recomendação para o novo item. Esta etapa é ilustrada na Figura 4.6, seguindo os seguintes passos:

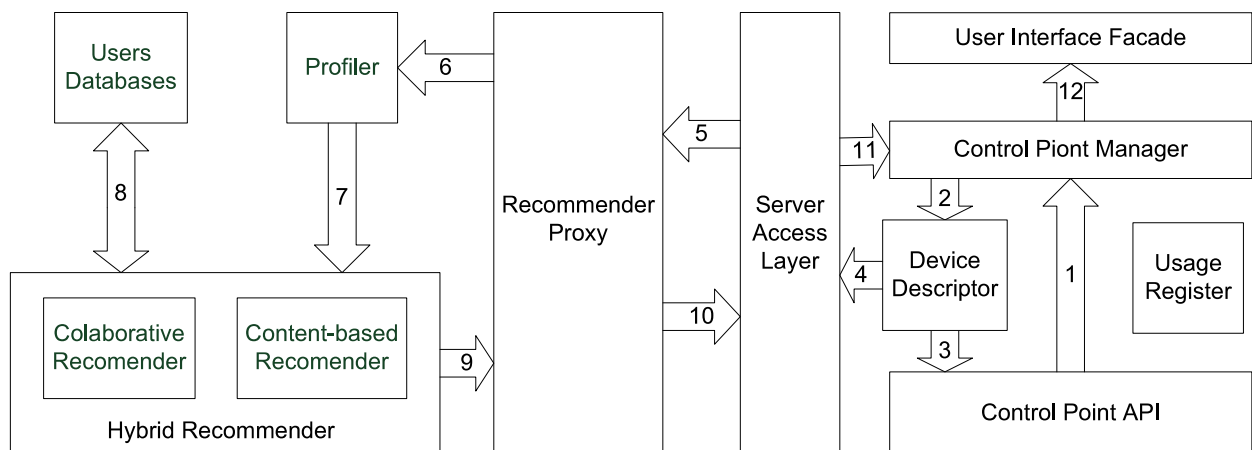


Figura 4.6: Entrada de um novo dispositivo UPnP na rede.

1. Quando um novo dispositivo UPnP entra na rede a *Control Point* API pode identificá-lo através do serviço de Descoberta do Padrão UPnP, como descrito na Seção 2.1. A *Control Point* API então informa o *Control Point Manager* da descoberta do novo dispositivo.
2. O *Control Point Manager* pede a descrição do novo dispositivo encontrado ao *Device Descriptor*.
3. Através das ações do novo dispositivo UPnP encontrado, o *Device Descriptor* coleta as informações relevantes para a Filtragem Baseada em Conteúdo. O *Device Descriptor* obtém a descrição dos dispositivos acessando as ações disponíveis nos serviços dos dispositivos upnp através da *Control Point* API, como mostrado na etapa de Controle apresentada na Seção 2.1.
4. Com a nova lista de Dispositivos e suas descrições o *Device Descriptor* envia as informações para o *Recommender Server* através da *Server Access Layer*.
5. A *Server Access Layer* envia a lista dos dispositivos e suas descrições para o *Recommender Server* através de uma chamada remota para a *Recommender Facade*.
6. A *Recommender Facade* envia os itens e suas descrições para o *Hybrid Recommender*.
7. O *Hybrid Recommender* usa os dados armazenados nos *Users Databases* para calcular as recomendações.
8. O *Hybrid Recommender* retorna então uma nova lista dos dispositivos para a *Recommender Facade* ordenada de acordo com o resultado da recomendação levando em conta a adição de um novo dispositivo UPnP na rede.
9. A *Recommender Facade* envia o resultado para a *Server Access Layer*.
10. A *Server Access Layer* repassa o resultado para o *Control Point Manager*.
11. O *Control Point Manager* disponibiliza a nova recomendação para a *User Interface Facade*.

#### 4.4.4 Dispositivo UPnP Sai da rede

Quando um dispositivo UPnP sai da rede ele não pode deixar nenhuma inconsistência e a rede UPnP e o Sistema de Recomendação devem continuar a funcionar normalmente. Esta etapa é ilustrada na Figura 4.7, seguindo os seguintes passos:

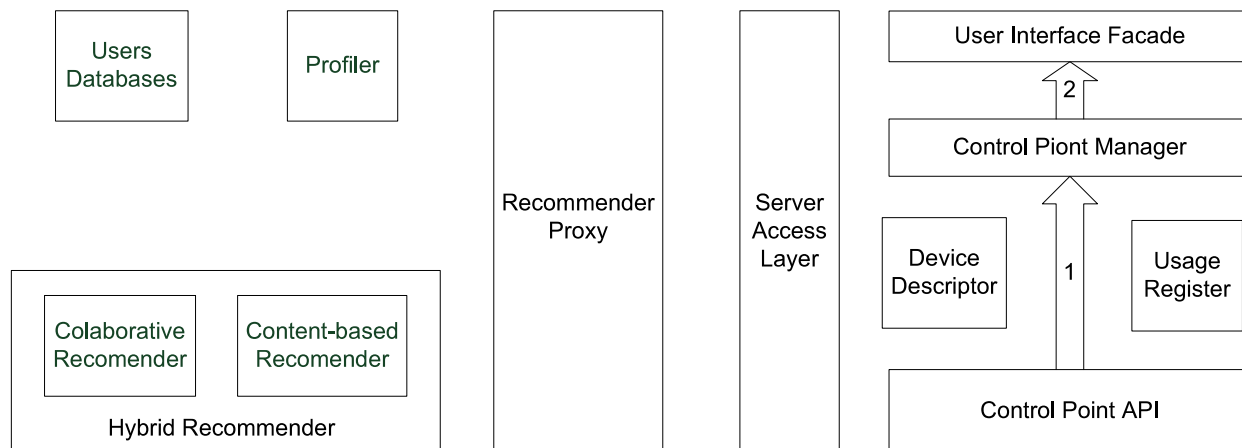


Figura 4.7: Remoção de um dispositivo UPnP da rede.

1. Quando um Dispositivo UPnP é removido da rede a *Control Point* API informa o *Control Point Manager*. A saída do dispositivo da rede é tratada normalmente pelo UPnP.
2. O *Control Point Manager* solicita então que a *User Interface Facade* que remova o Dispositivo da Interface com o Usuário.

## 4.5 Implementação do Protótipo

A arquitetura proposta neste trabalho pode ser implementada para as várias categorias de dispositivos UPnP disponíveis. Como forma de validação do trabalho, esta arquitetura foi implementada em um sistema de recomendação de dispositivos que implementem o padrão UPnP *Media Server Device* descrito na subseção a seguir. A implementação do protótipo foi feita utilizando a linguagem de programação Python [33]. Nesta seção são apresentadas as especificidades referentes a implementação da Arquitetura de Recomendação de Dispositivos UPnP para dispositivos que implementam o padrão UPnP *Media Server Device*.

### 4.5.1 UPnP Media Server

O UPnP *Media Server Device* [31] é definido na Arquitetura UPnP AV [30]. A Tabela 4.3 mostra os serviços necessários a implementação de um *Media Server*.

O serviço *Content Directory* permite aos clientes deste serviço acessar, de uma maneira semelhante ao acesso a uma estrutura de diretórios, os itens de mídia disponíveis no *Media Server* [34]. As ações do *Content Directory* são listadas na Tabela 4.4.

O *Connection Manager* permite o fluxo de dados de mídia entre dispositivos UPnP [35]. Ele permite ao *Media Server* enviar seu conteúdo de mídia para outros dispositivos UPnP

Dispositivo	Serviço	Obrigatoriedade
MediaServer	ContentDirectory	Obrigatório
	ConnectionManager	Obrigatório
	AVTransport	Opcional

Tabela 4.3: Serviços do UPnP *Media Server*

Ação	Obrigatoriedade
GetSearchCapabilities	Obrigatório
GetSortCapabilities	Obrigatório
GetSystemUpdateID	Obrigatório
Browse	Obrigatório
Search	Opcional
CreateObject	Opcional
DestroyObject	Opcional
UpdateObject	Opcional
ImportResource	Opcional
ExportResource	Opcional
StopTransferResource	Opcional
GetTransferProgress	Opcional
DeleteResource	Opcional
CreateReference	Opcional

Tabela 4.4: Ações do *Content Directory*

que desejem consumi-lo, como um *Media Render*, por exemplo. Na Tabela 4.5 são listadas as Ações do *Connection Manager*.

### 4.5.2 *Control Point*

O acesso às funcionalidades do *Control Point* necessárias para acessar os *Media Servers* disponíveis na rede são fornecidas pelo BRisa UPnP AV *framework* [36, 37]. O BRisa é escrito em Python, e fornece uma infra-estrutura para a implementação de *Media Servers*, *Media Renderers* e *Control Points* de acordo com a UPnP AV *Architecture*.

O módulo da arquitetura *Control Point* API faz o acesso ao serviço de rede UPnP através do Através do BRisa UPnP AV *framework*, permitindo a busca de dispositivos e o acesso aos serviços *Content Directory* e *Connection Manager*.

O *Device Descriptor* obtém a descrição dos *Media Servers* através da Ação *browse* do Serviço *Content Directory*. Através desta Ação é possível navegar através de uma

Ação	Obrigatoriedade
GetProtocolInfo	Obrigatório
PrepareForConnection	Opcional
ConnectionComplete	Opcional
GetCurrentConnectionIDs	Obrigatório
GetCurrentConnectionInfo	Obrigatório

Tabela 4.5: Ações do *Connection Manager*

estrutura, semelhante a uma estrutura de diretórios em um sistema de arquivos, e obter a lista de itens de mídia disponibilizados pelo *Media Server*, assim como informações sobre os itens, como Artista, Álbum ou Estilo de Filme. Estas informações são utilizadas como características descritoras do dispositivo e a relevância de dada características é dada pelo número de itens que possuem a característica.

O *Usage Register* é responsável monitora as atividades do usuário através das chamadas ao Serviço *Connection Manager*, sempre que algum item de mídia for reproduzido é feita uma chamada ao *Connection Manager*, cada vez que o usuário reproduz um item de mídia o *Usage Register* informa ao *Recommender Server*.

Toda a comunicação com o *Recommender Server* é feita através da *Server Access Layer* que é implementada através de um cliente SOAP, utilizando a biblioteca Python SOAPy [38] que faz acesso aos serviços remotos disponibilizados pelo *Recommender Server*.

### 4.5.3 *Recommender Server*

O acesso ao *Recommender Server* é feito através da *Recommender Facade* que é implementada como um servidor SOAP e disponibiliza os serviços descritos na subseção 4.2.3 através de *webservices*. O servidor SOAP é baseado na biblioteca Python twisted [39].

O elemento central do *Recommender Server* é o *Hybrid Recommender*. A implementação do *Hybrid Recommender* foi feita através da criação de uma biblioteca de recomendação escrita em Python. Esta biblioteca implementa os algoritmos de recomendação descritos no capítulo 3 conforme a subseção 4.2.1.

Cada vez que um *Media Server* é usado para reproduzir alguma mídia o *Recommender Server* é informado e o *Profiler*, de posse da quantidade de itens de mídia associados a cada característica, calcula o peso de cada uma das características do dispositivo e atualiza o perfil do usuário ativo nos *Users Databases*.

Os *Users Databases* utilizam duas Tabelas separadas, armazenadas em arquivos e acessadas através de uma API Python. Esta abordagem foi utilizada para fins de protótipo tendo em vista que em um ambiente de produção se faz necessário o uso de um sistema



gerenciador de banco de dados.

# Capítulo 5

## Conclusão

Em ambientes pervasivos, vários dispositivos possuem alguma capacidade de processamento e conectividade, e destes dispositivos uma grande parte possui mobilidade, podendo se conectar na rede sem fio. Nestes ambientes os diversos dispositivos podem disponibilizar serviços aos usuários ou a outros dispositivos. Desta forma sempre que um usuário ou dispositivo se move entre estes ambientes é necessário que os serviços disponíveis sejam facilmente encontrados e configurados, levando a necessidade de mecanismo para a disponibilização localização e configuração destes serviços. Com a grande quantidade de dispositivos na rede o usuário também pode ter o problema de como localizar o serviço que realmente ele deseja, dentre tantos disponíveis. Criando a necessidade de filtrar os serviços mais relevantes ao usuário. A utilização de Sistemas de Recomendação em conjunto com a tecnologia UPnP, permite ao usuário localizar os serviços disponíveis na rede e dentre estes filtrar automaticamente os serviços mais relevantes.

Neste trabalho é apresentada uma arquitetura para a habilitação de recomendação de serviços em redes baseadas na tecnologia UPnP. O uso da tecnologia UPnP permite que os diversos dispositivos conectados a rede disponibilizem seus serviços. A adição ou remoção de um *Control Points* com suporte a recomendação de dispositivos UPnP não influencia no funcionamento de uma rede UPnP permitindo estes coexistam com *Control Points* convencionais em uma mesma rede UPnP. A arquitetura UPnP pode ser implementada em uma grande quantidade de tipos de dispositivos, sejam eles definidos pelo fórum UPnP ou pelos próprios fabricantes de dispositivos. O uso de Sistemas de Recomendação ajuda ao usuário lidar com a grande quantidade de dispositivos disponíveis em uma rede UPnP. Desta forma a arquitetura apresentada neste trabalho permite ao usuário localizar e utilizar os serviços que lhe sejam mais relevantes dentre os serviços disponíveis em uma rede, mesmo em ambientes que o usuário nunca tenha visitado.

Como forma de validação desta arquitetura, foi feita uma implementação desta utilizando a linguagem Python. Para a implementação do Servidor de Recomendação foi

criada uma biblioteca de recomendação, que pode ser usada em sistemas de recomendação baseados na linguagem Python. A infra-estrutura UPnP necessária para a implementação do protótipo é provida pelo *BRisa UPnP A/V Framework*, sendo contribuições a este projeto de código livre uma atividade desempenhada em paralelo às atividades do Mestrado. O protótipo implementado permite a recomendação de dispositivos que sigam a especificação UPnP *Media Server*.

Como trabalhos futuros, espera-se estender a implementação existente a outras categorias de dispositivos UPnP, como *Media Renderers* e *Print Service*. Outra atividade a ser desempenhada no futuro será estender a biblioteca de recomendação desenvolvida, procurando adicionar outras técnicas de recomendação, e eventualmente propor uma nova técnica que se mostre mais eficiente para a recomendação de serviços em ambientes pervasivos.

# Referências Bibliográficas

- [1] Contributing Members of the UPnP Forum, *UPnP Device Architecture*. [http://www.upnp.org/download/UPnPDA10\\_20000613.htm](http://www.upnp.org/download/UPnPDA10_20000613.htm), version 1.0 ed., Jun 2000. Acessado em Janeiro de 2008.
- [2] “UPnP Forum WebSite.” <http://www.upnp.org>. Acessado em Janeiro de 2008.
- [3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 734–749, June 2005.
- [4] D. W. McDonald, “Ubiquitous recommendation systems,” *Computer*, vol. 36, no. 10, p. 111, 2003.
- [5] J. L. Herlocker, *Understanding and improving automated collaborative filtering systems*. PhD thesis, University of Minnesota, 2000. Orientador: Joseph A. Konstan.
- [6] S. C. Cazella, *Aplicando a Relevância da Opinião de Usuários em Sistema de Recomendação para Pesquisadores*. PhD thesis, Universidade Federal do Rio Grande do Sul, 2006.
- [7] F. Zhu, M. Mutka, and L. Ni, “Service discovery in pervasive computing environments,” *Pervasive Computing, IEEE*, vol. 4, no. 4, pp. 81–90, Oct.-Dec. 2005.
- [8] B. Miller, T. Nixon, C. Tai, and M. Wood, “Home networking with universal plug and play,” *Communications Magazine, IEEE*, vol. 39, pp. 104–109, Dec 2001.
- [9] Y. Poltavets, J.-H. Park, and D. Kim, “Ieee1394 to upnp software bridge structure,” *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, pp. 375–376, Jan. 2005.
- [10] D. Kim, J. H. Park, P. Yevgen, K. Moon, and Y. Lee, “Ieee 1394/upnp software bridge,” *Consumer Electronics, IEEE Transactions on*, vol. 51, pp. 319–323, Feb. 2005.

- [11] A. Delphinanto, A. Koonen, M. Peeters, and F. den Hartog, “Proxying upnp service discovery and access to a non-ip bluetooth network on a mobile phone,” *Communications and Vehicular Technology in the Benelux, 2007 14th IEEE Symposium on*, pp. 1–5, Nov. 2007.
- [12] S. Knauth, D. Kaslin, R. Kistler, and A. Klapproth, “Upnp compression for ip based field devices in building automation,” *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*, pp. 445–448, Sept. 2006.
- [13] S. Knauth, R. Kistler, D. Kaslin, and A. Klapproth, “Upnp compression implementation for building automation devices,” *Industrial Informatics, 2007 5th IEEE International Conference on*, vol. 1, pp. 75–79, June 2007.
- [14] “IETF Zeroconf Working Group.” <http://www.zeroconf.org>. Acessado em Janeiro de 2008.
- [15] Apple Computer Inc, *Bonjour Technology Brief*. [http://images.apple.com/macosx/pdf/MacOSX\\_Bonjour\\_TB.pdf](http://images.apple.com/macosx/pdf/MacOSX_Bonjour_TB.pdf), Abril 2005.
- [16] “Jini Network Technology.” <http://www.sun.com/software/jini>. Acessado em Janeiro de 2008.
- [17] “Jini Communit.” <http://www.jini.org>. Acessado em Janeiro de 2008.
- [18] Contributing Members of Jini Community, *Jini Architecture Specification*. [http://www.jini.org/wiki/Jini\\_Architecture\\_Specification](http://www.jini.org/wiki/Jini_Architecture_Specification), Setembro 2006. Acessado em Janeiro de 2008.
- [19] M. Balabanović and Y. Shoham, “Fab: content-based, collaborative recommendation,” *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.
- [20] R. van Meteren and M. van Someren, *Using Content-Based Filtering for Recommendation*. [http://www.ics.forth.gr/potamias/mlnia/paper\\_6.pdf](http://www.ics.forth.gr/potamias/mlnia/paper_6.pdf). Acessado em Fevereiro de 2008.
- [21] M. Taschuck, “A hybrid knowledge-based/content-based recommender system in the bluejay genome browser,” 2006. Undergraduate Honours Thesis, Faculty of Medicine at the University of Calgary.
- [22] J. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, (San Francisco, CA), pp. 43–52, Morgan Kaufmann, 1998.

- [23] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” Working Paper Series 165, MIT Center for Coordination Science, Mar. 1994.
- [24] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [25] J. B. Schafer, J. A. Konstan, and J. Riedl, “E-commerce recommendation applications,” *Data Min. Knowl. Discov.*, vol. 5, no. 1-2, pp. 115–153, 2001.
- [26] J. B. Schafer, J. Konstan, and J. Riedl, “Recommender systems in e-commerce,” in *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, (New York, NY, USA), pp. 158–166, ACM, 1999.
- [27] D. Frankowski, S. K. Lam, S. Sen, F. M. Harper, S. Yilek, M. Cassano, and J. Riedl, “Recommenders everywhere:: the wikilens community-maintained recommender system,” in *WikiSym '07: Proceedings of the 2007 international symposium on Wikis*, (New York, NY, USA), pp. 47–60, ACM, 2007.
- [28] U. Shardanand and P. Maes, “Social information filtering: algorithms for automating ‘word of mouth’,” in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 210–217, ACM Press/Addison-Wesley Publishing Co., 1995.
- [29] E. Reategui, E. Boff, and J. A. Campbell, “Endowing a virtual character with personalization capabilities,” in *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, (Washington, DC, USA), pp. 220–224, IEEE Computer Society, 2006.
- [30] Contributing Members of the UPnP Forum, *UPnP AV Architecture*. <http://www.upnp.org/>, version 1.0 ed., Jun 2002. Acessado em Janeiro de 2008.
- [31] Contributing Members of the UPnP Forum, *MediaServer:1 Device Template*. <http://www.upnp.org/>, version 1.0 ed., Jun 2002. Acessado em Janeiro de 2008.
- [32] Contributing Members of the UPnP Forum, *MediaRenderer:1 Device Template Version 1.01*. <http://www.upnp.org/>, version 1.0 ed., Jun 2002. Acessado em Janeiro de 2008.
- [33] M. Pilgrim, *Dive Into Python*. Apress, July 2004.

- [34] Contributing Members of the UPnP Forum, *ContentDirectory:1 Service Template Version 1.01*. <http://www.upnp.org/>, version 1.0 ed., Jun 2002. Acessado em Janeiro de 2008.
- [35] Contributing Members of the UPnP Forum, *ConnectionManager:1 Service Template Version 1.01*. <http://www.upnp.org/>, version 1.0 ed., Jun 2002. Acessado em Janeiro de 2008.
- [36] A. Guedes, D. Santos, J. do Nascimento, L. Sales, A. Perkusich, and H. Almeida, “Set your multimedia application free with brisa framework: An open source upnp implementation for resource limited devices,” *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pp. 1257–1258, Jan. 2008.
- [37] A. L. V. Guedes, D. F. S. Santos, J. L. do Nascimento, L. M. Sales, A. Perkusich, and H. O. Almeida, “Brisa upnp a/v framework,” *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, pp. 1–2, Jan. 2008.
- [38] “A SOAP/XML Schema Library for Python.” <http://soapy.sourceforge.net/>. Acessado em Fevereiro de 2008.
- [39] “Twisted WebSite.” <http://twistedmatrix.com/trac/>. Acessado em Fevereiro de 2008.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)



[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)