

Explorando Processamento Semântico na Descoberta de Recursos para UBICOMP

Renato M. Dilli¹, Mateus Santin², Adenauer C. Yamin^{1 2}, Luiz A. M. Palazzo¹

¹PPGINF – Centro Politécnico – Universidade Católica de Pelotas (UCPEL)

²CI – Centro de Informática – Universidade Federal de Pelotas (UFPEL)

{dilli, adenauer, lpalazzo}@ucpel.tche.br

{mateus, adenauer}@ufpel.edu.br

Abstract. *This article proposes a mechanism for resource discovery for ubiquitous computing technologies with semantic processing aggregated to its architecture. The EXEHDA-SD contributes to increased expression in the representation and consultation for resources in ubiquitous environment. The mechanism considers the dynamics in which the resources enter and leave the environment and pursue issues such as scalability and User Preferences. The semantic processor and other components of the architecture of the mechanism are described as central contributions of work in progress.*

Resumo. *Este artigo propõe um mecanismo para descoberta de recursos para computação ubíqua com tecnologias de processamento semântico agregadas à sua arquitetura. O EXEHDA-SD contribui com aumento da expressividade na representação e consulta por recursos no ambiente ubíquo. O mecanismo considera a dinamicidade em que os recursos entram e saem do ambiente e persegue aspectos como escalabilidade e preferências do usuário. O processador semântico e os demais componentes da arquitetura do mecanismo são descritos enquanto contribuições centrais do trabalho em desenvolvimento.*

1. Introdução

Um ambiente de computação ubíqua pode ser composto por uma grande quantidade de recursos heterogêneos e dispersos. Cabe ao mecanismo de descoberta de recursos gerenciar os recursos para atender as demandas dos clientes de forma transparente. Ele também deve prever a dinamicidade em que os recursos usualmente entram e saem do ambiente [Soldatos et al. 2006]. A utilização de tecnologias para processamento semântico permite qualificar a descrição e consulta por recursos através de linguagens de descrição expressivas e inferir características que estavam implícitas.

Este trabalho está sendo desenvolvido no G3PD/PPGINF/UCPel, tendo como objetivo a modelagem e concepção do EXEHDA-*Semantic Discovery* (EXEHDA-SD), um mecanismo para descoberta de recursos com suporte semântico. O mecanismo proposto visa qualificar a descoberta de recursos do *middleware* EXEHDA [Yamin et al. 2005], no ambiente ubíquo disponibilizado pelo mesmo. Este ambiente é organizado em células compostas por uma Base, responsável pelo seu gerenciamento, e nodos fixos e móveis. A abrangência da célula é delimitada pelo administrador do ambiente, levando em conta o poder computacional de sua Base.

O modelo proposto também persegue suprir as demandas inerentes do ambiente ubíquo dentre as quais destacam-se: escalabilidade através de redes *super peer*, preferências do usuário, notificação do cliente quando o recurso desejado está disponível, controle de acesso a recursos de acordo com o perfil do usuário e agenda de períodos de disponibilidade dos recursos.

2. Trabalhos Relacionados

Esta seção apresenta uma comparação resumida dos mecanismos de descoberta de recursos quanto a linguagem utilizada para descrição e consulta por recursos, abrangência de uso, arquitetura utilizada na descoberta e também se considera as preferências do usuário. A tabela 1 apresenta uma visão geral desta comparação.

Tabela 1. Comparativo dos Trabalhos Relacionados

Modelos	Descrição	Consulta	Abrangência	Arquitetura	Pref.
Allemmand	OWL	RDQL	Grade	3*	-
Allia	-	-	Ad-hoc	3*	Sim
Condor	ClassAd	A/V	LAN	-	-
DReggie	DAML	Prolog	<i>m-commerce</i>	1*	Sim
GLOBUS	LDAP	LDAP	Grade	3*	-
Ludwig, Santen	DAML-S	LARKS	Grade	3*	-
INS/TWINE	XML	-	LAN	1,2*	-
Jini	LP A/V	LP	LAN	1,2*	Sim
OMM	TRIPLE	LP	Grade	3*	Sim
PerDis	XML	A/V	Ubíquo	3*	Sim
Pernas, Dantas	OWL	-	Grade	3*	-
Salutation	<i>templates</i> A/V	A/V	LAN	1,2,3*	-
SLP	STRING	A/V	LAN	1,3*	Sim
UPnP	XML	A/V	LAN	3*	-

*1-Diretório Centralizado Plano, 2-Centralizado Hierárquico, 3-Descentralizado

Dos 14 trabalhos representados na tabela 1 apenas 5 utilizam uma linguagem com maior expressividade na descrição dos recursos. A grande maioria dos mecanismos utilizam comparações de Atributo/Valor no processo de *matching*. A maioria dos mecanismos foram projetados para descoberta em pequenas redes ou *grids*. A maioria possui uma arquitetura descentralizada, sendo que alguns possibilitam o seu uso em arquiteturas centralizadas. Apenas 6 trabalhos consideram as preferências dos usuário.

As referências bibliográficas e o resumo detalhado de cada trabalho estudado podem ser encontradas em [Dilli 2010].

3. Visão geral do EXEHDA-SD

Com o intuito de potencializar a descoberta de recursos no ambiente ubíquo, o EXEHDA-SD promove o gerenciamento dos recursos através do processamento de ontologias [Jepsen 2009] valendo-se da API Jena, banco de dados MySQL e as linguagens de programação Java e SPARQL. As consultas também podem acontecer em células diferentes daquela que originou a pesquisa.

A arquitetura de software do EXEHDA-SD está sendo concebida considerando o estudo dos trabalhos relacionados, as demandas do *middleware* EXEHDA quanto à descoberta de recursos e as etapas no processo de descoberta definidas em [Zhu et al. 2005]. Sua arquitetura é baseada em três componentes distintos: CC - Componente Cliente, CR - Componente Recurso e CD - Componente Diretório, conforme a figura 1.

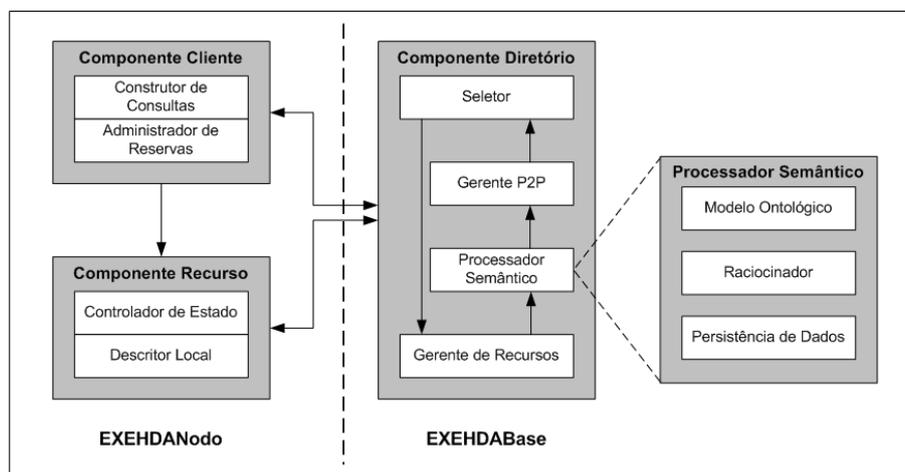


Figura 1. Modelo Proposto: Arquitetura EXEHDA-SD

3.1. CC - Componente Cliente

O CC é responsável pela seleção dos recursos desejados, através de critérios, e um perfil de descoberta que define características pertinentes no processamento da consulta, e ainda, dependendo da natureza da consulta são utilizadas as preferências do usuário. O CC é composto dos módulos Construtor de Consultas e Administrador de Reservas.

O Construtor de Consultas é responsável por interpretar os recursos necessários para atender uma aplicação e gerar o arquivo XML com a pesquisa desejada. O arquivo XML é enviado ao CD para geração da consulta em formato SPARQL.

O Administrador de Reservas é um módulo responsável por notificar o cliente quando o diretório verificar que o recurso desejado tornou-se disponível.

3.2. CR - Componente Recurso

O CR é responsável por notificar o estado atual do recurso. Isto é feito por troca de mensagens com o CD, dentro de um intervalo de tempo. Outra função do CR é anunciar seus recursos em células que estão sendo visitadas. Quando o usuário ingressa em uma nova célula o CR envia a descrição dos seus recursos para o CD da célula que está sendo visitada. O CD adiciona as instâncias referentes a descrição do recurso em sua ontologia. O CR é composto pelos módulos Controlador de Estado e Descriptor local.

3.3. CD - Componente Diretório

O CD é formado por quatro módulos: Gerente de Recursos, Processador Semântico, *Gerente P2P* e *Seleção*. O CD está localizado no EXEHDA Base de cada célula do ambiente celular.

3.3.1. Gerente de Recursos

Quando um CC (Componente Cliente) solicita que seja notificado sobre a disponibilidade de algum recurso, primeiramente é realizada a consulta sobre todos recursos “Ativos” da célula, caso a consulta não retorne resultados que atendam a requisição, a consulta será refeita fazendo uso dos recursos “Inativos”, mas existentes na ontologia. Caso a consulta retorne recursos que satisfaçam a requisição, o Gerente de Recursos notificará o CC quando o CR do recurso desejado renovar o *lease*, alterando o estado do recurso para “Ativo”. No estado “Ativo” o recurso estará disponível para ser alocado pelo *Resource Broker* do *middleware* EXEHDA [Yamin et al. 2005].

O CRA (Controle de Recursos Ativos) tem por objetivo gerenciar os recursos que entram e saem do ambiente celular. Quando um recurso entra no ambiente o CR envia uma mensagem para o Gerente de Recursos e este adiciona o *Nodo_ID* onde está localizado o recurso e o *lease* padrão definido pelo mecanismo. O *lease* é um intervalo de tempo que é gerenciado pelo Gerente de Recursos. O CR precisa renovar o *lease* do recurso localizado no CD periodicamente, caso contrário o Gerente de Recursos irá remover o recurso do CRA e alterar o estado para “Inativo”.

Quando uma pesquisa é recebida pelo Gerente de Recursos, e no seu perfil está declarada a solicitação de notificação de recursos disponíveis, o Gerente de Recursos armazenará no repositório do CNR (Controle de Notificação de Recursos Disponíveis) o *Nodo_ID* do recurso desejado, mas que não está disponível no momento, o ID do Cliente que solicitou a pesquisa, a consulta, bem como a data e a hora que foi feita a pesquisa e a data e a hora que irá expirar a espera pelo recurso desejado.

Quando o CR anuncia um recurso ao Gerente de Recursos, este verifica se o recurso está no CRA e renova seu *lease* ou adiciona o recurso com o *lease* padrão caso não esteja presente do CRA. Após, o Gerente de Recursos consulta o CNR para verificar se o recurso não está sendo esperado por algum cliente. Se afirmativo, o cliente será notificado que o recurso está disponível.

Este módulo também tem por objetivo controlar o agendamento de disponibilidade do recurso. Pode-se tomar como exemplo um recurso que deverá estar disponível ao ambiente somente nos dias da semana: segundas-feiras, quartas-feiras e sextas-feiras e no seguinte horário: 15h às 21h.

3.3.2. Processador Semântico

O Processador Semântico é implementado em Java com API Jena e é formado por três componentes: Modelo Ontológico, Raciocinador e Persistência de Dados.

O Modelo Ontológico é responsável pela manutenção da ontologia do mecanismo de descoberta. Esta ontologia é desenvolvida na linguagem OWL.

O raciocinador é responsável por aplicar regras e processar as consultas realizadas em SPARQL. O raciocinador irá procurar por recursos idênticos e semelhantes ao solicitado.

O módulo de Persistência de Dados é responsável pelo armazenamento das triplas

da ontologia no banco de dados MySQL.

O EXEHDA-SD utiliza as ontologias para a representação dos recursos no ambiente ubíquo e pesquisa dos recursos instanciados. O Processador Semântico é responsável pela instanciação de novos recursos e processamento de regras de inferência, possibilitando ao mecanismo de descoberta o reconhecimento de conceitos implícitos na ontologia.

3.3.3. Gerente P2P

O Gerente P2P é o módulo responsável pela comunicação com as células vizinhas. A comunicação entre as células será realizada utilizando tecnologias P2P entre os EXEH-DABase de cada célula. Será utilizada uma variação do P2P puro, chamada *super peer*. Neste modelo a comunicação cliente/servidor ocorrerá apenas entre os diretórios localizados nos EXEH-DABase de cada célula. Os CC e CR acessarão apenas o diretório da célula local. O Gerente P2P será responsável por repassar a pesquisa no formato da linguagem SPARQL para as células vizinhas de acordo com o número de saltos definidos pelo CC.

3.3.4. Seletor

Quando uma pesquisa por recursos retornar mais de um recurso, o módulo Seletor fará a classificação e ordenará os recursos posicionando os que melhor satisfazem a requisição no topo da lista, enquanto aqueles que menos satisfazem ficarão ao final.

3.4. Especificação da consulta por recursos

O mecanismo de descoberta de recursos proposto está sendo modelado para processar a consulta dos Componentes Cliente através de arquivos XML. O arquivo XML possuirá o recurso desejado, atributos e valores e o perfil a ser aplicado no processo de descoberta.

Exemplo de uma consulta com especificação de critérios:

Listagem 1. Consulta em XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<CONSULTA>
  <PERFIL>SD_1</PERFIL>
  <TIPO>Nodo</TIPO>
  <CRITERIO nome='SO' valor='Unix' op='equ' />
  <CRITERIO nome='Processador' valor='Intel' op='equ' />
</CONSULTA>
```

Na listagem 1 é especificada uma consulta em XML contendo as seguintes *tags*:

- <PERFIL> Indica o perfil de execução a ser utilizada na consulta.
- <TIPO> Tipo de recurso desejado.
- <CRITERIO> O critério de pesquisa é composto por “nome”, “valor” e op (operador). É possível utilizar vários critérios para refinar a pesquisa.

O Gerente de Recursos do Diretório receberá o arquivo XML e converterá em SPARQL para enviar ao Processador Semântico.

Listagem 2. Consulta em SPARQL

```
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#" +
"PREFIX ont: <http://www.owl-ontologies.com/Ontology1251223167.owl#" +
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#" +
"SELECT ?nodo " +
"WHERE " +
" { ?nodo rdf:type ont:Nodo ." +
" ?nodo ont:Nodo_Status 'Ativo' ." +
" ?nodo ont:Nodo_Base false ." +
" ?nodo ont:Nodo_Celula ?celula ." +
" ?celula ont:Celula_ID 'UCPel' ." +
" ?nodo ont:Nodo_Software ?SO ." +
" ?SO rdf:type ont:Unix ." +
" ?nodo ont:Nodo_Dispositivos ?processador ." +
" ?processador rdf:type ont:Intel }";
```

A consulta representada na listagem 2 levou em consideração o perfil SD_1 especificado no arquivo XML. Esse perfil define a profundidade da consulta em célula local. Partindo que a consulta foi originada da célula chamada “UCPel” a consulta em SPARQL irá pesquisar na ontologia por nodos que possuam instâncias de Sistema Operacional “Unix” e instâncias de Processador “Intel”. O processamento semântico da consulta poderá retornar, além da resposta exata, recursos similares ao solicitado. Essa funcionalidade permite expandir a potencialidade do mecanismo retornando recursos que talvez possam atender a demanda do cliente quando o desejado não estiver disponível.

4. Considerações Finais

A principal contribuição do EXEHDA-SD quando confrontado aos modelos existentes é a expressividade na representação e consulta de recursos, a qual é aumentada através do processamento semântico empregando o modelo ontológico projetado para o mecanismo. A discussão detalhando os benefícios decorrentes de um modelo baseado em processamento semântico em relação àqueles baseados em processamento sintático está disponível em [Dilli 2010]. O mecanismo está em fase de modelagem e pretende-se realizar implementações e simulações para validar as características pretendidas para o EXEHDA-SD.

Referências

- Dilli, R. M. (2010). Um mecanismo para descoberta de recursos com suporte semântico na computação ubíqua. *Dissertação de Mestrado I, Universidade Católica de Pelotas, Relatório Técnico*.
- Jepsen, T. C. (2009). Just what is an ontology, anyway? *IT Professional*, 11:22–27.
- Soldatos, J., Pandis, I., Stamatis, K., Polymenakos, L., and Crowley, J. L. (2006). Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services. *Journal of Computer Communications*.
- Yamin, A. C., Augustin, I., da Silva, L. C., Real, R. A., and Geyer, C. F. R. (2005). EXEHDA middleware: aspects to manage the ISAM pervasive environment. In *SCCC*, pages 84–92.
- Zhu, F., Mutka, M., and Ni, L. (2005). Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, 4(4):81–90.