

Em Direção a Descoberta de Recursos Baseada em Matching Semântico para UBICOMP

Fundamentos da IA Contemporânea

Renato Marques Dilli
Prof. Luiz A. M. Palazzo
Prof. Adenauer C. Yamin

Universidade Católica de Pelotas

13 de agosto de 2009

Mestrado em Ciência da Computação

- 1 Objetivos
- 2 Contexto
- 3 Web Semântica
- 4 Modelo Proposto
- 5 Conclusão e Trabalhos Futuros

Objetivo Geral

Modelar as características iniciais do mecanismo de descoberta de recursos para Computação Ubíqua com suporte semântico.

Objetivos Específicos

- Definir a arquitetura inicial do mecanismo de descoberta;
- Construir uma ontologia para o mecanismo;
- Instanciar a ontologia com alguns recursos, através do Protégé;
- Utilizar a API Jena para processar a ontologia;
- Simular consultas por recursos instanciados na ontologia com a utilização de raciocinadores;

- O ambiente ubíquo deste trabalho é disponibilizado pelo *middleware* EXEHDA;
- O modelo de descoberta de recursos fará parte da arquitetura deste *middleware*

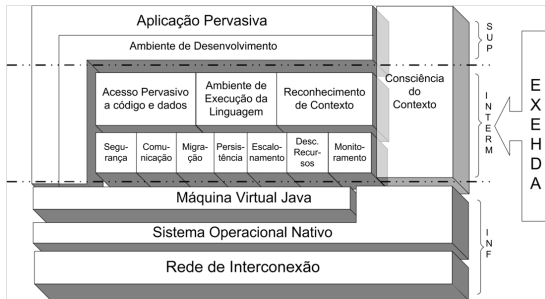


Figura: Middleware EXEHDA

- A topologia do ambiente é formada por células, conforme a figura abaixo:

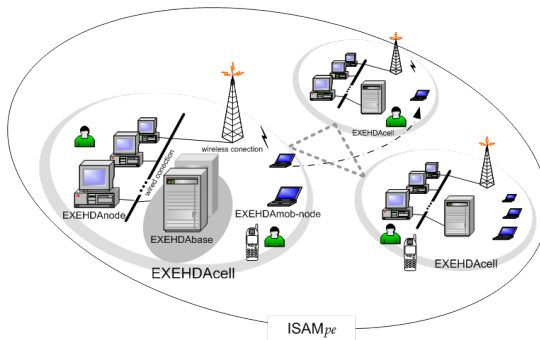


Figura: Ambiente Pervasivo

Características Importantes:

- Linguagem de Consulta e Mensagens
- Escalabilidade
- Suporte a Raciocinadores

- Transitive reasoner:
- RDFS rule reasoner
- OWL, OWL Mini, OWL Micro Reasoners
- DAML micro reasoner
- Generic rule reasoner
- Externos baseados no padrão DIG (*Description Logic Interface*)

Modelo Proposto - ISAMPe

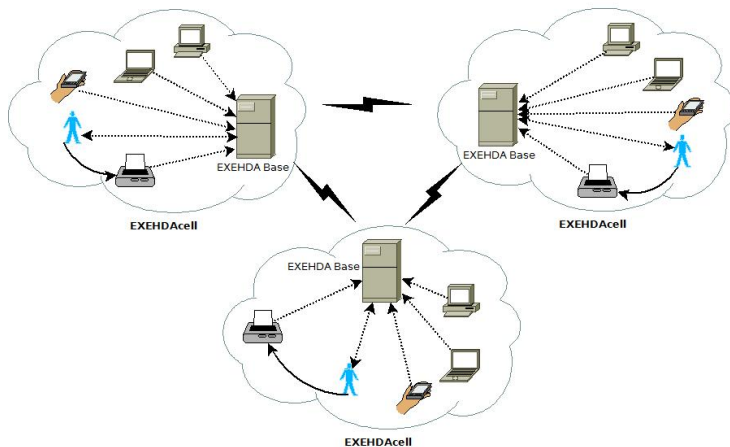


Figura: Ambiente Celular de Descoberta

Modelo Proposto - Ontologia Instanciada

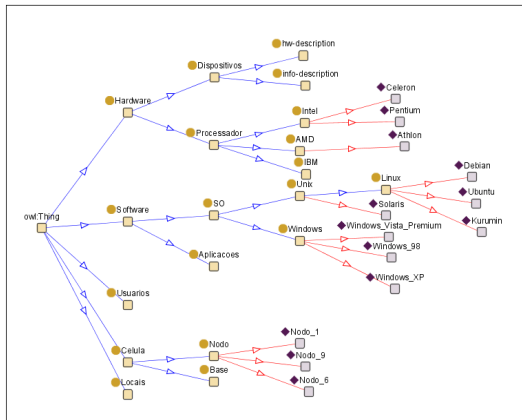


Figura: Ontologia do Mecanismo de Descoberta

- Hardware: qualquer dispositivo que poderá ser utilizado no ambiente;
- Software: qualquer software que pode ser utilizado no ambiente;
- Locais: todos locais onde pode ser usado o modelo;
- Célula: contém informações sobre as células do EXEHDA, inclusive nodos e EXEHDA Base;
- Usuários: todos usuários que podem usar o ambiente.

Para a Classe Nodo foram criados os seguintes relacionamentos:

- SistOper: Um Nodo possui uma instância da Classe SO;
- Proces: Um nodo possui uma instância da Classe Processador.

No primeiro experimento foi realizada a consulta por instâncias de sistemas operacionais UNIX sem o uso de raciocinadores. Esta consulta foi realizada através da linguagem SPARQL.

Listing 1: Consulta SPARQL

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://localhost/Ontologia.owl#>
SELECT *
    WHERE
        {?recurso rdf:type ont:Unix}
```

Esta consulta foi realizada sem auxílio do raciocinador da API Jena e apresentou a seguinte resposta:

Instâncias de Unix sem Raciocinador:

```
=====
<http://localhost/Ontologia.owl#Solaris>
```

No segundo teste foi ativado o raciocinador e realizada a mesma consulta, apresentando o seguinte resultado:

Instâncias de Unix com Raciocinador:

```
=====
<http://localhost/Ontologia.owl#Solaris>
<http://localhost/Ontologia.owl#Kurumin>
<http://localhost/Ontologia.owl#Debian>
<http://localhost/Ontologia.owl#Ubuntu>
```

A segunda consulta realizada realiza uma procura por nodos com Sistema Operacional Unix e Processadores Intel. Sendo que Unix e Intel são classes que contém sub-classes.

Listing 2: Consulta SPARQL 2

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://localhost/Ontologia.owl#>
SELECT *
WHERE
    { ?nodo ont:SistOper ?SO .
      ?SO rdf:type ont:Unix .
      ?nodo ont:Proces ?Processador .
      ?Processador rdf:type ont:Intel }
```

O resultado obtido com a segunda consulta foi o seguinte:

```
<Ontologia.owl#Nodo_1> | <Ontologia.owl#Debian> |
<Ontologia.owl#Pentium>
```

Listing 3: Jena - Definição de Regras (Ontologia.rules)

```
@prefix ont: <http://localhost/Ontologia.owl#>.
@include <RDFS>.

[dualcore: (?n ?p ont:Nodo) (?n ont:Num_Proc 2) -> (?n rdf:type
    ont:dualcore)]
[quadcore: (?n ?p ont:Nodo) (?n ont:Num_Proc 4) -> (?n rdf:type
    ont:quadcore)]
[manycore: (?n ?p ont:Nodo) (?n ont:Num_Proc ?x) greaterThan(?x
    ,4) -> (?n rdf:type ont:manycore)]
```

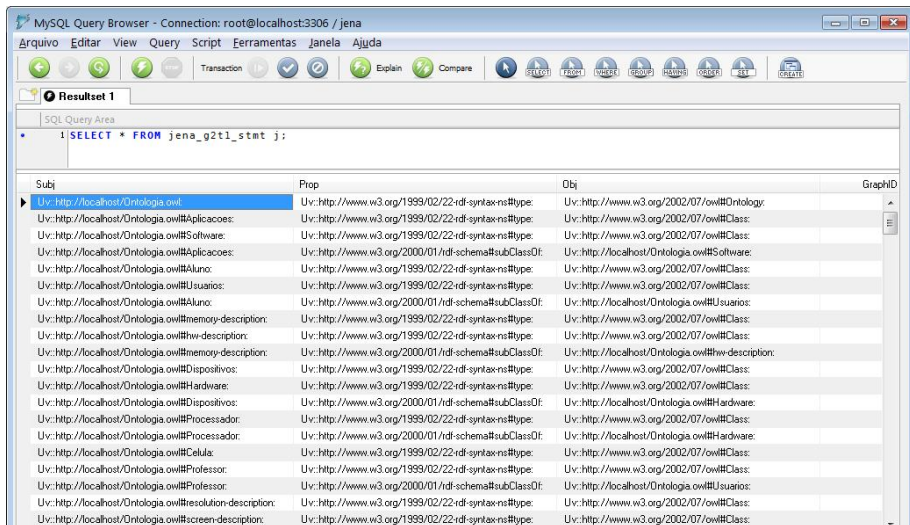
Listing 4: Jena - Resultado da Consulta por dualcore

run :

```
| nodo |
=====
| <http://localhost/Ontologia.owl#Nodo_6> |
| <http://localhost/Ontologia.owl#Nodo_2> |
```

Modelo Proposto - Persistência de Dados

Através do suporte a persistência de dados da API Jena foi possível armazenar a ontologia no banco de dados MySQL



The screenshot displays the MySQL Query Browser interface. The title bar indicates the connection is 'root@localhost:3306 / jena'. The menu bar includes 'Arquivo', 'Editar', 'View', 'Query', 'Script', 'Ferramentas', 'Janela', and 'Ajuda'. The toolbar contains icons for navigation and execution. The 'Resultset 1' tab is active, showing a table with four columns: 'Subj', 'Prop', 'Obj', and 'GraphID'. The 'Subj' column lists various ontology classes and properties, such as 'Uv: http://localhost/Ontologia.owl', 'Uv: http://localhost/Ontologia.owl#Aplicacoes', and 'Uv: http://localhost/Ontologia.owl#Software'. The 'Prop' column contains URIs for the properties, and the 'Obj' column contains URIs for the objects. The 'GraphID' column is empty. The table is sorted by 'Subj' in ascending order.

Subj	Prop	Obj	GraphID
Uv: http://localhost/Ontologia.owl	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Ontology	
Uv: http://localhost/Ontologia.owl#Aplicacoes	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Software	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Aplicacoes	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#Software	
Uv: http://localhost/Ontologia.owl#Aluno	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Usuarios	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Aluno	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#Usuarios	
Uv: http://localhost/Ontologia.owl#memory-description	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#hw-description	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#hw-description	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#memory-description	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#hw-description	
Uv: http://localhost/Ontologia.owl#Dispositivos	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Hardware	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Dispositivos	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#Hardware	
Uv: http://localhost/Ontologia.owl#Processador	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Processador	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#Hardware	
Uv: http://localhost/Ontologia.owl#Celula	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Professor	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#Professor	Uv: http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv: http://localhost/Ontologia.owl#Usuarios	
Uv: http://localhost/Ontologia.owl#resolution-description	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	
Uv: http://localhost/Ontologia.owl#screen-description	Uv: http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv: http://www.w3.org/2002/07/owl#Class	

Modelo Proposto - Persistência de Dados

Após armazenar as triplas da ontologia no banco de dados é possível abrí-lo e realizar consultas SPARQL diretamente no banco, conforme código Java abaixo:

Listing 5: Consultas com Banco de Dados

```
public static void Consulta_BD(String Ontology) {
    String url = "jdbc:mysql://localhost/jena";
    String driver = "com.mysql.jdbc.Driver";
    DBConnection conn = new DBConnection ( url, "root", "
        mysql", "mysql" );
    ModelRDB model = ModelRDB.open(conn,"DR");
    String ocQuery = "PREFIX rdf: <http://www.w3.org
        /1999/02/22-rdf-syntax-ns#>" +
        "PREFIX ont: <http://localhost/
        Ontologia.owl#>" +
        "SELECT * " +
        "WHERE " +
        "{?recurso rdf:type ont:Unix} ";
    QueryExecution ocQe = QueryExecutionFactory.create(
        ocQuery , model);
```

Listing 6: Validação da Ontologia

```
public static void Validar(String Ontology) {  
    OntModel model = ModelFactory.createOntologyModel(  
        OntModelSpec.OWL_MEM);  
    model.read(Ontology);  
    Reasoner reasoner = ReasonerRegistry.getOWLReasoner();  
    reasoner = reasoner.bindSchema(model);  
    InfModel owlInfModel = ModelFactory.createInfModel(  
        reasoner, model);  
    ValidityReport vrp1 = owlInfModel.validate();  
    if (vrp1.isValid()) {  
        System.out.println("Ontologia OK");  
    } else {  
        System.out.println("Ontologia com problemas...");  
        for (Iterator i = vrp1.getReports(); i.hasNext(); )  
        {  
            System.out.println(" - " + i.next());  
        }  
    }  
}
```

Para teste da validação foi alterado o valor da “Veloc_Proc” que está definido como “float” de 3.0 para a string “tres”

O resultado apresentado foi a inconsistência da ontologia gerada:

Ontologia com problemas...

- Error ("range check"): "Incorrectly typed literal due to range (prop, value)"

Culprit = http://localhost/Ontologia.owl#Nodo_1

Implicated node: http://localhost/Ontologia.owl#Veloc_Proc

Implicated node: 'tres' <http://www.w3.org/2001/XMLSchema#float>

Conclusão

O trabalho encontra-se em fase inicial, mas acredita-se que pelos testes realizados o suporte semântico do mecanismo de descoberta garantirá respostas mais completas e abrangentes.

Trabalhos Futuros

- ampliar a ontologia, definindo relacionamentos e regras entre as classes;
- expandir os componentes da arquitetura proposta;
- definir o funcionamento do *matching* entre células do ISAMpe.

Perguntas?

Em Direção a Descoberta de Recursos Baseada em Matching Semântico para UBICOMP

Renato Dilli - dilli@ucpel.tche.br