An Ontology and Concept Lattice Based Inexact Matching Method for Service Discovery

Tang Shancheng
Communication and Information Institute, Xi'an University of Science and Technology, Xi'an 710054, China
e-mail tangshancheng@21cn.com

Abstract: To overcome shortcomings of Exact Matching Method (EMM) and Substitute Description Method (SDM), an Ontology and Concept Lattice Based Inexact Matching Method (OCLIMM) is introduced. This method, firstly, describes resources information with Ontology languages to get Resource Ontology Descriptions (RODs); then, infers Substantiality Concept Lattices and Property Concept Lattices from ROD; lastly, computes Formal Concept Distance between the required resource and each of existing resources based on Concept Lattices, and selects resources by formal concept distances and thresholds. The results of experiments indicate: firstly, OCLIMM improves EMM at the aspect of matching count when inputs are same, and the more count of input properties is, the more degree of improvement is when their parameters are uniform, i.e., OCLIMM can utilize fully resources discovered than EMM; then, OCLIMM is averagely 4.55 times as large as EMM at the aspect of matching count even if the threshold is zero; at last, OCLIMM can adapt more smartly than SDM to the condition of lots of resources and properties being in pervasive computing environment.

Key word: ontology; concept lattice; inexact matching; service discovery

1 Introduction

Devices in Pervasive Computing Environment have some features such as large-scale, mobility, alterability; these features bring galactic frustration to manual resources configuration, resources discovery and interaction with resources. To users' views devices in Pervasive Computing environment should disappear into the background (be "invisible"), the devices should automatically be configured, managed, discovered, and used by other devices with a minimum of manual effort, not intrude on users' consciousness [1]. Resource Discovery and Interaction (RDI) technologies are developed to remove this frustration and to fulfil the "disappearance". Selecting resources from resources discovered is one important goal of RDI technologies. Many of the existing RDI technologies are based on Exact Matching Method (EMM), such as Intentional Naming System (INS), Salutation, Jini, UPnP; EMM supports an attribute-based discovery as well as a simple name lookup to select resources [2][3]. For example, a user needs "a printer printing A4 paper", and there are three resources in pervasive computing environment: "R1, a printer printing A3 paper", "R2, a printer printing A4 paper", "R3, a plotter printing A1 paper", then, EMM will select R2 for the user; If there are only R1 and R3 in the environment EMM will select nothing for user, although R1 satisfies truly user's need and R3 can satisfy the need at a certain extent.

To overcome the shortcoming of EMM which cannot utilize fully resources discovered, there is an intuitionistic method: Substitute Description Method (SDM). SDM records "Substitute" relations, i.e., describes "a printer printing A3 paper" as a substitute of "a printer printing A4 paper", then, when there are no printers required SDM can find a substitute to answer the need. But this method is not perfect, it requires that we should describe relations of substitute in advance, that's very difficult in pervasive computing environment for the environment is mobile, alterable.

To overcome shortcomings of EMM and SDM, and to utilize fully resources discovered this paper introduces an Ontology and Concept Lattice Based Inexact Matching Method (OCLIMM); This method assumes that there are no semantic conflicts (i.e., Naming conflicts, domain conflicts, structural conflicts, Metadata conflicts [4]) in RDI, though these semantic conflicts are needed to solved this paper don't discuss them.

2 OCLIMM

2.1 Outline

This method, firstly, describes resources information (such as resources, resource properties, relations between resources, relations between resource properties) with Ontology languages to get Resource Ontology Descriptions (RODs); then, infers Substantiality Concept Lattices and Property Concept Lattices from RODs; lastly, computes Formal Concept Distance between the required resource and each of existing resources based on Concept Lattices, and selects resources by Formal Concept Distances and thresholds. Following description follows the above-men-

 978-1-935068-01-3 © 2009 SciRes.

tioned procedure.

2.2 Resources Ontology

2.2.1 Ontology and Semantic Web

In the domain of knowledge management, ontology is referred as the shared understanding of some domains, which is often conceived as a set of entities, relations, axioms and instances. It has four significations: conceptualization, explicit, formal, share. There are several reasons for developing RODs based on ontologies: ① Knowledge Sharing. RODs enable computational entities in pervasive computing environments to have a common set of concepts about condition and to avoid misconceiver; ② Logic Inference. Based on ontology, RDIs can deduce high-level, conceptual knowledge from low-level, raw resource descriptions; ③ Knowledge Reuse. We can build new Ontologies based on reusing well-defined Ontologies without starting from scratch.

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It is designed for use by applications that need to process the content of information instead of just presenting information to humans [5]. It has a set of standards, among these standards, RDF is a datamodel for objects and relations between them, provides a simple semantics for this datamodel; RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization hierarchies of such properties and classes; OWL adds more vocabulary for describing properties and classes. OWL is based on description logic (DL), which allows OWL to exploit DL reasoning. We will describe resource ontology with RDF/RDFS/OWL.

2.2.2 ROD and Ontology Reasoning

Resource Ontology is structured around a set of abstract entities, which are physical or conceptual objects. Each entity (rdfs:Class) is associated with its attributes (rdf: Property) and associated with other entities. The built-in rdfs: subClassOf and rdfs:subPropertyOf allows to hierarchically structure sub-class and sub-property respectively (Figure 1).

The equivalence of OWL and description logic allows OWL to exploit DL reasoning to meet important logical requirements, which include concept satisfiability, class subsumption, class consistency, and instance checking) to carry out experiments. We use Jena2 Semantic Web Framework [6] to reason RODs and to get Substantiality Concept Lattices and Property Concept Lattices; Jena2 supports rule-based inference over OWL/RDF graphs.

2.3. Concept Lattices

To describe this paper expediently, we firstly get following definitions:

```
<owl:Class rdf:ID="Resource"/>
<owl:Class rdf:ID="Output">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</owl:Class>
…
<owl:Class rdf:ID="Hardcopy">
  <rdfs:subClassOf rdf:resource="#Output"/>
</owl:Class>
…
<rdfs:Class rdf:ID="Printer">
  <rdfs:subClassOf rdf:resource="#Hardcopy"/>
</rdfs:Class>
…
<rdf:Property rdf:ID="HardcopySize">
  <rdfs:domain rdf:resource="#Hardcopy"/>
</rdf:Property>
<rdf:Property rdf:ID="HardcopySizeA1">
  <rdfs:subPropertyOf rdf:resource="#HardcopySize"/>
</rdf:Property>
<rdf:Property rdf:ID="HardcopySizeA3">
  <rdfs:subPropertyOf rdf:resource="#HardcopySizeA1"/>
</rdf:Property>
<rdf:Property rdf:ID="HardcopySizeA4">
  <rdfs:subPropertyOf rdf:resource="#HardcopySizeA3"/>
</rdf:Property>
…
```

Figure 1. Part of RDF/RDFS/OWL description of resources ontology

Definition 1. Formal Context

A **formal context** (context) is a triple [7,8]:

$$K = (G, M, I)$$

G is a finite set of objects; M is a finite set of properties. I is a binary relation between G and M: $I \subseteq G \times M$.

Let $X \subseteq G$ and $Y \subseteq M$. The mappings:

$$\sigma(X) = \{m \in M \mid \forall g \in X: (m, g) \in I\}$$

the common properties of X, and

$$\tau(Y) = \{g \in G \mid \forall m \in Y: (m, g) \in I\}$$

the common objects of Y, form a Galois connection.

Definition 2. Formal Concepts

A **formal concept** (concept) is a pair of sets: a set of objects (the extent) and a set of properties (the intent) (X, Y) such that:

$$Y = \sigma(X) \text{ and } X = \tau(Y)$$

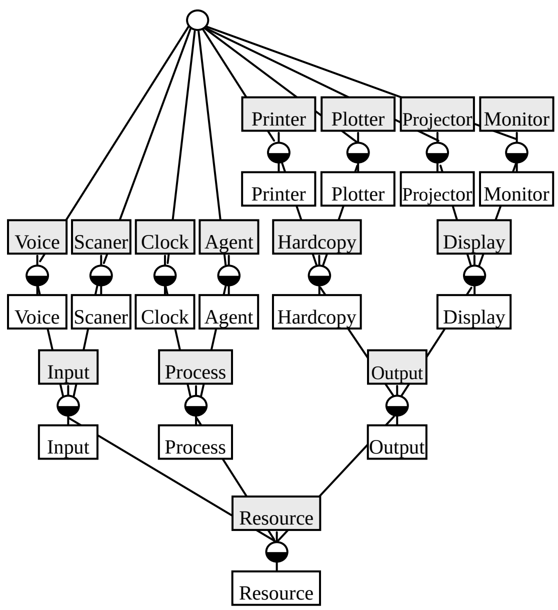
Therefore a concept is a maximal collection of objects sharing common properties.

Definition 3. Concept Lattices

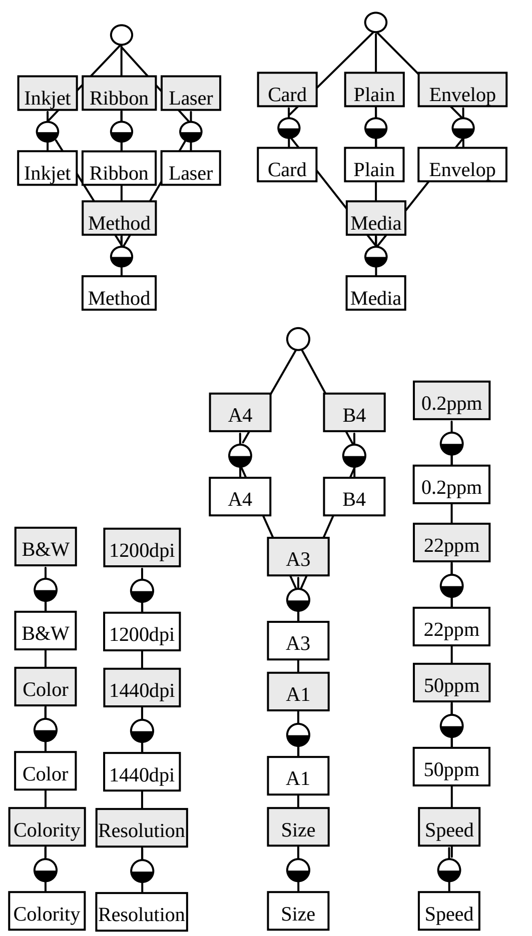
In the following we denote bβ(K)y or β(G, M, I) the set of all concepts of the context K. We now define a binary relation ≤ (to be read as "is a subconcept of") on the set β(K) of all concepts of K as follows:

If (X_0, Y_0) and (X_1, Y_1) are concepts of K, then $(X_0, Y_0) \le (X_1, Y_1)$ (i.e. the concept (X_0, Y_0) is a **subconcept** of the concept (X_1, Y_1)) if and only if, $X_0 \subseteq X_1$ (or $Y_1 \subseteq Y_0$), and (X_1, Y_1) is then also called **superconcept** of the concept (X_0, Y_0).

With the subconcept-superconcept relation "≤" on the set β(K), we get ordered set (β(K), ≤), and the ordered set is then called the "**concept lattice**" of K.



(a) Substantiality Concept Lattice



(b) Six Property Concept Lattices of Substantiality Hardcopy

Figure 2. Concept Lattices infered from Resource Ontology

Definition 4. Substantiality Formal Concept Distance

In the following, let a context S = (E, D, R), E and D are a finite set of substantialities respectively, E = D, R is a binary relation of parent class between E and D: R = {(e, d) | e ∈ E, d ∈ D and e is a parent class of d}, S is then called Substantiality Formal Context. For example, E = D = {Resource, Input, Output, Voice, Scaner, …}, R= {(Resource, Resource), (Resource, Input), (Resource, Output), Resource, Voice), (Resource, Scaner), (Input, Input), (Iinput, Voice), (Input, Scaner), …}.

On the concept lattice (β(S), ≤) (Figure 2 (a)), set two substantialities o ∈ E and p∈ D, the **Substantiality Formal Concept Distance** between o and p is defined by:

FCDS (o, p) = max (|A - B|, |B – A|) / max (|A|, |B|)

A = τ(σ(o)), B= τ(σ(p)); |A|, |B|, |A - B|, |B – A| are the cardinalities of sets A, B, A – B, B – A respectively; max (x, y) returns the larger one from x and y.

Explanation:

(1) 0 ≤ FCDS (o, p) ≤ 1;
(2) FCDS (o, p) = 0, o is p, the substantiality formal concept distance between o and p is smallest;
(3) FCDS (o, p) = 1, there are no relation of parent class between o and p, the substantiality formal concept distance between o and p is largest;
(4) FCDS(o, p) = FCDS(p, o);
(5) FCDS (o, p) > FCDS (m, n), the substantiality formal concept distance between o and p is larger than it between m and n.

Definition 5. Property Formal Concept Distance

In the following, let a context P = (E, D, R), E and D are a finite set of properties respectively, E = D, R is a binary relation of compatibility between E and D: R = {(e, d) | e ∈ E, d ∈ D and e is compatible with d}, P is then called Property Formal Context. For example, E = D = {Size, A1, A3, A4, B4}, R={(Size, Size), (Size, A1), (Size, A3), (Size, A4), (Size, B4), (A1, A1), (A1, A3), (A1, A4) , (A1, B4), (A3, A3), (A3, A4), (A3, B4), (A4, A4), (B4, B4)}.

On the concept lattice (β(P), ≤) (Figure 2 (b)), set two properties o ∈ E and p∈ D, the **Property Formal Concept Distance** between o and p is defined by:

$$FCDP(o,p)=\begin{cases}1 & |A\cap B|\le 1\\ |B-A|/|B| & |A\cap B|>1\end{cases}$$

A = τ(σ(o)), B= τ(σ(p)); |B|, |B – A|, |A ∩ B| are the cardinalities of sets B, B – A, A ∩ B respectively.

Explanation:

(1) 0 ≤ FCDP (o, p) ≤ 1;
(2) FCDP (o, p) = 0, o is p, or p is compatible with o, the property formal concept distance between o and p is smallest;
(3) FCDP (o, p) = 1, there are no relation of compatibility between o and p, the property formal concept distance between o and p is largest;
(4) FCDP(o, p) is not always equal to FCDP(p, o);

978-1-935068-01-3 © 2009 SciRes.

(5) FCDP (o, p) > FCDP (m, n), the property formal concept distance between o and p is larger than it between m and n.

Definition 6 . Property Tuple Formal Concept Distance

Assume that one property tuple is $T_i = (i_1, i_2, i_3, \ldots, i_n)$, another property tuple is $T_j = (j_1, j_2, j_3, \ldots, j_n)$; $i_k \in E_k$, $j_k \in D_k$; $P_k = (E_k, D_k, R_k)$, $E_k = D_k$; Property Tuple Formal Concept Distance between T_i and T_j is defined by:

$$FCDPT(T_i, T_j) = \left[\sum_{k=1}^{n} w_k (FCDP(i_k, j_k))^2\right]^{1/2}$$

W_k is the weight of property formal context P_k, and $\sum_{k=1}^{n} w_k = 1$.

Explanation:

(1) $0 \le$ FCDP (o, p) ≤ 1;

(2) FCDPT (T_i, T_i) = 0;

(3) FCDPT (T_i, T_j) is not always equal to FCDPT (T_j, T_i).

```
Source[] OCLIMM(Ontology Ont,Source REQ,Source[] SOU,float WeightOfS,float[] WeightOfPT,float T){
  float FCDS=1,FCDPT=1,FCDW=1;
  Souce[] selected;
  Ont.Inference()->S,P;//Infering Ontology
  for(int k=0;k<SOU.length;k++){
    FCDS=getFCDS(S,REQ.REQS,SOU[k].SOUS);
    FCDPT=getFCDPT(P,REQ.REQPT,SOU[k].SOUPT,WeightOfPT);
    FCDW=WeightOfP*FCDS+(1-WeightOfP)*FCDPT;
    if(FCDW<=T){selected.add(SOU[k]); }
  } return selected;
}
float getFCDS(Context S,Substantiality o,Substantiality p){
  if(o==p) return 0;
  HashSet A= S .getCommonObj(S .getCommonPer(o));
  HashSet B= S .getCommonObj(S .getCommonPer(p));
  int tempn=Max(getBase(Difference(A,B)),getBase(Difference(B,A)));
  int tempd=Max(getBase(A),getBase(B));
  return tempn/tempd;
}
float getFCDPT(Context[] P,Property[] ,Property[] SOUPT, float[] WeightOfPT){
  float temp=0;
  for(int i=0;i<P.length;i++){temp = temp + WeightOfPT[i] * Math.pow (getFCDP(P[i], REQPT[i], SOUPT[i]), 2); }
  return Math.sqrt(temp);
}
float getFCDP(Context Pi,Property o,Property p){
  if(o==p) return 0;
  HashSet A= Pi .getCommonObj(Pi .getCommonPer(o));
  HashSet B= Pi .getCommonObj(Pi .getCommonPer(p));
  if(getBase(Intersection(A,B))<=1) return 1;
  return getBase(Difference(B,A))/getBase(B);
}
```

Figure 3. Pseudocode of OCLIMM Arithmetic

2.4. Arithmetic of OCLIMM

We assume that Ont is a ROD, then, we can get a substantiality concept context S and a corresponding concept lattice (β(S), $\le$), and a group of property formal context P = (P_1, P_2, P_3, …, P_x) and relevant concept lattices; REQ is the description of a resource which a user required and it comprises substantiality REQS, Property Tuple REQPT = (i_1, i_2, i_3, …, i_x); The resources in Pervasive Computing environment are SOU = (S_1, S_2, S_3, …, S_n) and SOU_k comprises Concept $SOUS_k$, Property Tuple $SOUPT_k$; We assume that weight of substantiality formal concept distance is WeithtOfS and property formal concept distance is Weight Of PT = (W_1, W_2, W_3, …, W_x), and threshold of holistic formal concept distance is T. The pseudocode of OCLIMM is shown in Figure 3. Asymptotic time complexity of computing holistic formal concept distance in OCLIMM is O(n).

3. Experiments and analysis of results

3.1 Experiments design

Experiments settings: there are three resources in the experiment computing environment: ① R1, Printer, Laser, Black and white (B & W), plain paper, A3, 1200dpi, 50ppm; ② R2, Printer, Inkjet, Color, plain paper, A4, 1440dpi, 22ppm; ③ R3, Plotter, Inkjet, Color, plain paper, A1, 1200dpi, 0.2ppm.

Experiments procedure: firstly, we get all possible printers combination according to substantiality "Hardcopy" and all its properties; Then, we take each of these printers combination as a resource which a user requires to match resources (three) in computing environment by OCLIMM and EMM; Last, we can select resources according to threshold.

Experiments parameters: We have following groups: ① P = 1, T = 0; ② P = 1, T = 0.4; ③ P = 0.5, T = 0; ④ P = 0.5, T = 0.2; ⑤ P = 0.5, T = 0.4.

3.2 Analysis of results

The results of experiments (Table 1) indicate: ① If the inputs are same OCLIMM improves EMM at aspect of matching count, and the more count of properties of input is, the more degree of improvement is when their parameters are uniform; The reason of this result is: OCLIMM utilizes the relation of subconcept – superconcept described in each property concpet lattice, and the subconcept's corresponding property can satisfy the superconcept's corresponding property affirmatively, for example (Figure 2 (b)), there are superconcept ({A3, A1, Size}, {A3, A4, B4}) and subconcept({A1, Size}, {A1, A3, A4, B4}),and their corresponding property are A3 and A1, then, the Size A1 can satisfy A3 predicatively; ② Even if T is zero (i.e. resources selected can fulfill entirely the requirements), OCLIMM is averagely 4.55 times as large as EMM at the aspect of matching count; ③ If we adopt SDM we need to store many relations, the count of relations will increase multiplicatively according to the increase of properties count, the reason is: "Degree of improvement about matching count from OCLIMM to EMM" × "matching count of EMM" = "count of relations needed to store". This shows that SDM does not adapt to the condition of lots of resources and properties being in pervasive computing environment, and OCLIMM will do contrarily.

Table 1. Experiment results of EMM and OCLIMM

Input	Para1	Resu1	P	Para2	Resu2	Resu3
Prn + 0 Per	1	2	1	0	2	0.00
			1	0.4	3	0.50
		12	0.5	0	19	0.58
Prn + 1 Per	17		0.5	0.2	35	1.92
			0.5	0.4	39	2.25
		30	0.5	0	72	1.40
Prn + 2 Per	119		0.5	0.2	171	4.70
			0.5	0.4	297	8.90
		40	0.5	0	139	2.48
Prn + 3 Per	439		0.5	0.2	451	10.28
			0.5	0.4	965	23.13
		30	0.5	0	145	3.83
Prn + 4 Per	900		0.5	0.2	604	19.13
			0.5	0.4	2412	79.40
		12	0.5	0	78	5.50
Prn + 5 Per	972		0.5	0.2	442	35.83
			0.5	0.4	2612	216.67
		2	0.5	0	17	7.50
Prn + 6 Per	432		0.5	0.2	120	59.00
			0.5	0.4	1104	551.00

Note:
Para1: Count of input combination
Para2: threshold T
Resu1: Matching count of EMM
Resu2: Matching count of OCLIMM
Resu3: Degree of improvement about matching count from OCLIMM to EMM
Prn: Printer
Per: Properties

4. Conclusion

To overcome the shortcomings of EMM and SDM OCLIMM is presented. The method firstly, describes resources information with Ontology languages to get RODs, then, infers Substantiality Concept Lattices and Property Concept Lattices from RODs; lastly, computes Formal Concept Distance between the required resource and each of existing resources based on Concept Lattices, and selects resources by formal concept distances and thresholds. The results of experiments indicate: firstly, OCLIMM improves EMM at the aspect of matching count when inputs are same, and the more count of properties of input is, the more degree of improvement is when their parameters are uniform, i.e., OCLIMM can utilize fully resources discovered than EMM; then, OCLIMM is averagely 4.55 times as large as EMM at the aspect of matching count even if the threshold is zero; last, OCLIMM can adapt more smartly than SDM to the condition of lots of resources and properties being in pervasive computing environment.

References

[1] Mark Weiser. The future of ubiquitous computing on campus[J]. COMMUNICATIONS OF THE ACM, 1998, 41(1): 41−42.
[2] F Zhu, M Mutka, L Ni. Classification of Service Discovery in Pervasive Computing Environments[R]. MSU-CSE-02-24. East Lansing: Michigan State University, 2002: 1−17.
[3] Simone A. Ludwig, Peter van Santen. A Grid Service Discovery Matchmaker based on Ontology Description[C]. EuroWeb 2002 Conference St Anne's College, Oxford, UK, 2002: 17−18.
[4] Declan O'Sullivan, David Lewis. Semantically driven service interoperability for pervasive computing[C]. Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access, 2003: 17−24.
[5] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web[J], Scientific American, May 2001.
[6] Jena2 Semantic Web Framework: http://www.hpl.hp.com/semweb/ [W], 2006.
[7] Rudolf Wille. Restructuring lattice theory: an approach based on hierarchies of concepts[M]. Ivan Rival, editor, Ordered sets, Reidel, Dordrecht, 1982: 445−470.
[8] ZHANG Dong-xiao. Decision of the attribute reductivity about concept lattices[J]. Computer Engineering and Applications. 2007, 43(22): 165−169.