

# Instance-Based Ontology Mapping

Karin K. Breitman<sup>1</sup>, Daniela Brauner<sup>1</sup>, Marco Antonio Casanova<sup>1</sup>, Ruy Milidiú<sup>1</sup>, Alexandre Gazola<sup>1</sup> Marcelo Perazolo<sup>2</sup>

<sup>1</sup>Pontifícia Universidade Católica do Rio de Janeiro, <sup>2</sup>IBM Corporation  
(karin, brauner, casanova, ruy,agazola)@inf.puc-rio.br, mperazol@ibm.us.com

## Abstract

*We have explored the use of formal ontology and alignment techniques as a general approach to mediate and reconcile different representations. This approach proved to be very effective when fueled by ontologies rich in detail, (properties, restrictions, attributes and axioms that hold among classes), but performs poorly when available representations are incomplete or lacking, which is often the case in real life settings. In this paper we propose an enhancement to the original approach, based on a, instance-based matching technique.*

## 1. Introduction

Perhaps the greatest challenge in managing today's IT scenario is the diversity of managed resource interfaces that co-exist today in a typical data centers. Many argue that the ability to integrate and share such resources in a rational way will ultimately decide the efficiency of such environments [Sterritt05, Kephart03].

The great majority of such representations derive from the Distributed Management Task Force's common information models standards such as those proposed by the Distributed Management Task Force (DMTF), but this fact does not necessarily guarantee that the original semantics of the standard is retained. A canonical example is the variations over DTMF's Common Information Model (CIM) standards [DTMF07]. Many management applications are available on the market today that provide their own interpretation of the CIM standard, e.g. Microsoft Windows operating systems (WMI), and the IBM Tivoli Monitoring platform (ITM), making it very hard to promote cross platform integration and sharing. It is often the case where different interpretations of the same underlying model coexist, thus providing different solutions to the same management problem.

Our group has been consistently addressing this problem. A general solution was obtained by adapting semantic web ontology integration techniques [Casanova 07, Breitman07, Breitman06, Brauner06]. We created a semantic layer on top of different CIM-

based resource representations using the Web Ontology Language (OWL). An ontology integration engine, CATO, was developed to map between different representations, thus enabling tight integration and avoiding ambiguous/different interpretations of ontological classes.

Although quite accurate, the ontology integration approach implemented by CATO fails when the available resource representations are poor in detail, i.e., little information on properties, restrictions and axioms that hold among ontology classes. This information is the foundation on top of which the ontology alignment engine works. Henceforth, poorly defined ontologies yield insufficient mappings. In this paper we argue in favor of an instance-based technique that combines query results to learning algorithms to identify mappings between ontologies. This new technique complements the CATO approach and improves results when the original resource representations are incomplete or poorly defined. The rest of this paper is divided as follows. In section 2 we summarize the CATO ontology integration engine. In section 3 we introduce the proposed instance based approach. In section 4 we discuss model calibration. In section we 5 discuss the contributions of the proposed approach and present our final remarks.

## 2. Ontology Alignment Approach

The problem of integrating heterogeneous CIM-based IT resource representations can be compared to that of database schema matching. Given two schemas, A and B, one wants to find a mapping  $\mu$  from the concepts in A into the concepts of B in such a way that, if  $a = \mu(b)$ , then b and a have the same meaning. To enable the automation of the process, we coded the resource representations using the web ontology language (OWL). Ontologies are expressive, formal, machine processable representations that fulfill the knowledge requirements of autonomic systems [Stojanovic04], i.e., "(1)capture facts of the world and render them a way that it can be used by autonomic managers and (2) enable new knowledge to be discovered and learned" [Miller05].

Industrial scale IT applications involve dozens of resources to provide an integrated solution. In this scenario we should expect that if each resource is represented by an independent ontology, there must be a mechanism to support ontology negotiation. Often referred to as ontology integration, this negotiation process aims at finding an intermediate representation that can be shared by the involved applications to ensure communication. Semantic interoperability among ontologies has been in the research agenda of knowledge engineers for a while now. A few approaches to help deal with the ontology integration problem have been proposed. The most prominent ones are: merging [Hage05], alignment [Ehrig06 Noy99, Stoilos05, Ehrig05], mapping [Noy03] and integration [Zhang05].

Our past experience with semantic interoperability enabled us to provide a solution, CATO, that combines well known algorithmic solutions, e.g., natural language processing, the use of similarity measurements, and tree comparison, to the ontology integration problem. The first version of CATO was fully implemented in JAVA and uses a specific API (Application Programming Interface) that deals with ontologies, JENA. CATO was developed in the Eclipse environment [Breitman07, Breitman06, Brauner06, Felicissimo04]. We propose to incorporate CATO to the knowledge element of autonomic managers. Another possibility is to implement it as an autonomic manager of its own right whose managed elements would be the published resource ontologies of other autonomic managers. This new autonomic manager would contain a single plan, to align two or more ontologies.

The philosophy underlying CATO's strategy mixes syntactical and semantic analysis of ontological components. During the integration process lexical and structural comparisons are performed in order to determine if concepts in different ontologies should be considered semantically compatible. A refinement

approach is used that alternates between lexical and structural comparison between ontological concepts. The process begins when concepts from both ontologies go through a lexical normalization process, in which they are transformed to a canonical format that eliminates the use of plurals and gender flexions. The concepts are then compared, with the aid of a dictionary. The goal is to identify pairs of lexically equivalent concepts.

We assume that lexically equivalent concepts imply the same semantics, if the ontologies in question are in the same domain of discourse. For pairs of ontologies in different domains, lexical equivalence does not guarantee that concepts share the same meaning [Stoilos05, Noy03]. To solve this problem, we adopted a structural comparison strategy. Concepts that were once identified as lexically equivalent are now structurally investigated. Making use of the intrinsic structure of ontologies, a hierarchy of concepts connected by subsumption relationships, we now isolate and compare concept sub-trees. Investigation on the ancestors (super-concepts) and descendants (sub-concepts) will provide the necessary additional information needed to verify whether the pair of lexically equivalent concepts can actually be assumed to be semantically compatible.

Lexical comparison is done during the first and second steps of the strategy. Structural analysis is done in the second and third steps of the strategy. The final result is an OWL document containing *equivalent class* statements (<owl:equivalentClass>) that relate the equivalent concepts from the two input ontologies. This is equivalent to a mapping between conceptual schemas. The proposed strategy is depicted in Figure 1.

We must imagine our universe of discourse to be composed of a myriad of autonomic managers that act autonomously in an open ended environment, driven by their own goals [IBM06, Perazolo06]. In order to fulfill their tasks, collaboration with other

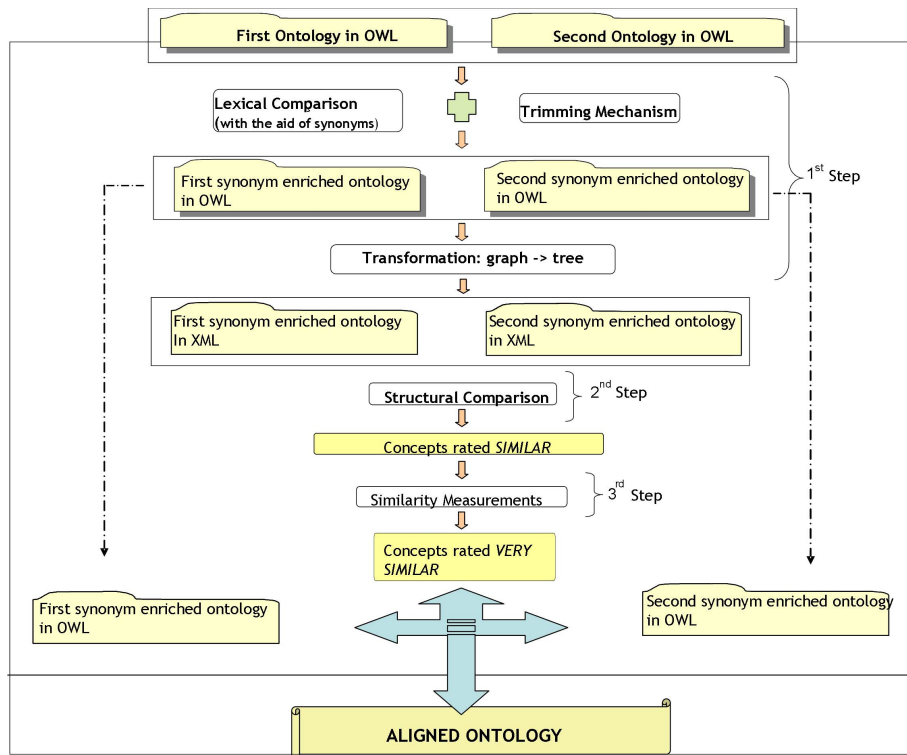


Figure 1. CATO ontology alignment strategy [Breitman07]

### 3. Instance Based Approach

autonomic managers is often required. Since different knowledge resources are likely to provide separate ontologies, the ability to align the ontologies into a single representation that can be shared by different applications is paramount to ensure communication. To secure true autonomic behavior, decisions taken by autonomic managers must be done as independently of human intervention as possible

The ontology alignment solution thus proposed presents some degree of risk, in the sense that it cannot fully guarantee that the most adequate resource or resource combination will be always identified. Limitations of the algorithms used, time to perform the computations, and possible lack of information coded in the original ontologies may, in some of the cases, prevent the automated solution to identify answers that would be otherwise manually found. Heavy experimentation will be required in order to determine whether the proposed solution is satisfactory, to what cases, and in which situations it may be considered cost effective.

In the previous section we described CATO, an ontology integration engine that is able to automatically detect matches based on information coded in the input ontologies. CATO takes into consideration the structure of the ontologies (which are the parents, siblings and offspring of the classes in question, and do they provide useful information that help identify or rule out a possible match?), their syntax (what are the labels in use in the input ontology? Are there counterparts in the second ontology? Do they provide reliable hints of possible matches?), and semantics (What roles, axioms and restrictions can we identify in relation to a class? How does a given class relate to other classes in the ontology (specially with those whose mappings to the second ontology were already identified).

The success of the CATO approach depends on the volume and quality of the information coded in the input ontologies. The richer and more complete the information, the better the results. Conversely, if the input ontologies are poorly defined, incomplete or lacking, the ontology integration engine has little data to work upon, and thus is not likely to deliver adequate results.

To tackle such situations, we propose the use of an instance based approach. Instead of matching ontology classes, as does CATO, we switch our attention to the ontology instances. Ontologies, as opposed to databases, store the data schema (classes) together with data itself (instances) in a single file. Each implementation of a commercial IT management application, such as the Windows Management Instrumentation service in Microsoft Windows (WMI), or the IBM Tivoli Monitoring platform (ITM), can be thus represented by an ontology, containing a set of classes, restrictions, properties (data schema), that correspond to its interpretation of the DMTF's CIM [DTMF07] standards, and a set of instances, that correspond to the actual resources it manages, i.e., printers, servers, routers, etc.

The goal of the instance based approach is the same, find matching classes across different ontologies. Traditional matching approaches, such as the one implemented by CATO, use the data schemas themselves to find the mappings, i.e., class hierarchy and class descriptions, including axioms, restrictions and properties. Another way to look at the problem is reducing it to detecting how equivalent objects (assuming that there is an automatic way to recognize object equivalence) are stored in different databases (ontologies in our particular case) [Casanova07]. Proceed by analyzing the classes (on either side) under which equivalent objects were found to see if they can be aligned.

The central idea is to process results from a query posed to two different ontologies as a means to devise a reliable estimation model that indicates mapping rates among pairs of aligned concepts. Depending on a threshold value (detailed in section 4) our strategy identifies valid pairings. Fundamental to this approach is the ability to automatically detect instance equivalence. In this particular case, every resource instance possess the a unique identifier provided by either the global trade item number (GTIN) or by the universal product code (UPC) tag.

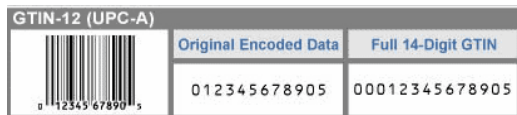
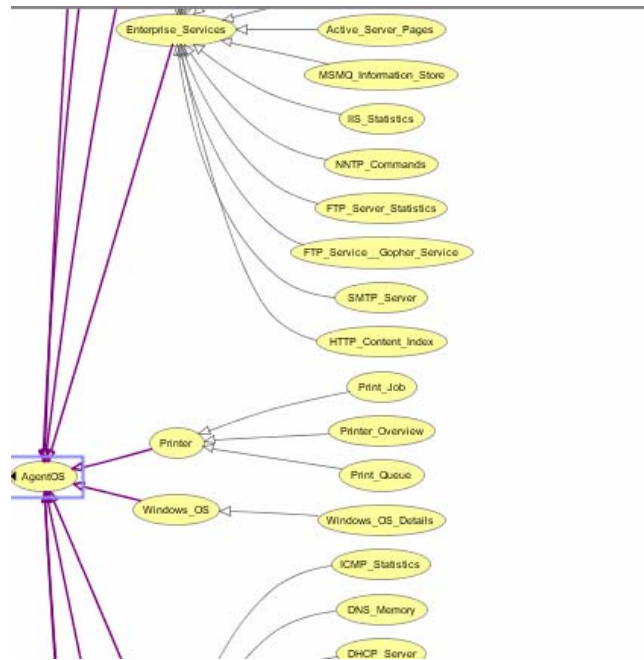


Figure 2. Example of a GTIN tag

In the following sections we describe the instance based approach. To facilitate understanding we adopted as working example the integration of a WMI ontology (746 classes, current implementation with 78

instances of managed resources) and ITM ontology (345 classes, 67 instances of managed resources). Both correspond to real life implementations. Figures 3 and 4 depict the ontologies and a partial list of their instances.



Class Name		GTIN	
NT_Physical Disk	HP - Disk drive - floppy disk ( 1.44 MB ) - 3.5" Slim Line	Machine96876	390164-B21
NT_Physical Disk	Wyse CD-ROM - internal	Machine96876	920223-03
NT_Physical Disk	CD Rom Drive	Machine9987	920263-04
NT_Physical Disk	Wyse CD-ROM - internal	Machine96654	920223-03
NT_Physical Disk	Wyse CD-ROM - internal	Machine2356	920223-03
NT_Printer	LPT1	room12	
NT_Printer	LPT2	room23	<a href="#">0734646028677</a>
NT_Printer	obaluae	room34	<a href="#">0829160867885</a>
NT_Printer	LPT2	room12	<a href="#">0013803025569</a>
NT_Printer	andorinha	room21	<a href="#">0042215441469</a>
NT_Printer	Printer239	room34	<a href="#">0829160410852</a>
NT_Printer	Impressora267	room67	<a href="#">0829160410852</a>
NT_Printer	Impressora_99_9	room34	<a href="#">0734646123273</a>

Figure 3. (a) ITM ontology classes (b) Partial list of ITM ontology instances

Class Name	GTIN
------------	------

<a href="#">Win32_FloppyDrive</a>	Manages the capabilities of a floppy disk drive.	Wyse_Disk_5676920223-12		
<a href="#">Win32_FloppyDrive</a>	Manages the capabilities of a floppy disk drive.	Wyse_Disk_2376920223-12		
<a href="#">Win32_FloppyDrive</a>	Manages the capabilities of a floppy disk drive.	Wyse_Disk78676920223-12		
<a href="#">Win32_PhysicalMedia</a>	Represents any type of documentation storage medium.			
<a href="#">Win32_PhysicalMedia</a>	Represents any type of documentation storage medium.	Wyse_Disk_876	9202233-01	
<a href="#">Win32_PhysicalMedia</a>	Represents any type of documentation storage medium.	Wyse_Disk_376	920223-12	
<a href="#">Win32_PhysicalMedia</a>	Represents any type of documentation storage medium.	Wyse_Disk_876	920223-12	

Figure 4. Partial list of WMI ontology instances

### 3.1 Querying

Let  $O$  be an ontology that has a class hierarchy  $C$ .  $C$  can be expressed by the set  $[c_1 \dots c_n]$  where  $c_1 \dots c_n$  are classes in  $O$ . Every class in  $O$  may be instantiated by one or more objects, denoted by an expression of the form  $[r_{k1} \dots r_{kn}]$  where  $r_{k1} \dots r_{kn}$  are instances of class  $c_k$  that belongs to class hierarchy  $C$ .

Assume that we have the ontologies depicted in Figures 3 and 4 that we call  $O_A$  and  $O_B$ , storing objects from the same domain, classified using the class hierarchies shown in Figures 3 and 4  $C_A$  and  $C_B$ , respectively. Also, we will be interested in mapping classes from the class hierarchy  $C_A$  to  $C_B$ . The generalization of this discussion to more than two ontologies are outside the scope of this paper.

We say that instances  $r_a$  in  $O_A$  and  $r_b$  in  $O_B$ , respectively, are *equivalent*, denoted  $r_a \equiv r_b$ , when they represent the same (real-world) object; in this case, we also say that  $c_a$  and  $c_b$  *map to each other*, where  $c_a$  and  $c_b$  are the classes under which instances  $r_a$  and  $r_b$  are classified under, respectively. The exact procedure that computes instance equivalence depends on the application, as previously discussed. In this case we use the GTIN – UPC tag, as illustrated in the rightmost columns in Figures 3 and 4.

We proceed by posing several similar pair of queries to  $O_A$  and  $O_B$ . The idea is to retrieve objects from both ontologies that are similarly classified, albeit using different class hierarchies.

For each pair of classes  $c_a \in C_A$  and  $c_b \in C_B$ , we store the following information:

- $n(c_a, c_b)$ , the sum of the all occurrences of pairs of instances  $r_a$  and  $r_b$  such that:

- o  $r_a$  and  $r_b$  are in the result sets of  $Q_A$  and  $Q_B$
- o  $r_a \equiv r_b$
- o  $c_a$  and  $c_b$  are the classes of  $r_a$  and  $r_b$ , respectively

$P(c_a, c_b)$ , an estimation for the frequency that the class  $c_a$  maps to the class  $c_b$ , for each pair of classes  $c_a \in C_A$  and  $c_b \in C_B$ . We call  $P(c_a, c_b)$  the *mapping rate estimator* for  $c_a$  and  $c_b$ , calculated as follows:

$$P(c_a, c_b) = \frac{n(c_a, c_b)}{n(c_a)}$$

For each  $c_a \in C_A$  we also store  $n(c_a)$ , the sum of all occurrences of  $c_a$  in submitted queries.

### 3.2 Analysis

This section outlines the statistical model we adopt to analyze queries and their results. Assume that we have already processed a number of queries and that estimations for  $n(c_a)$ ,  $n(c_a, c_b)$  and  $P(c_a, c_b)$ , for

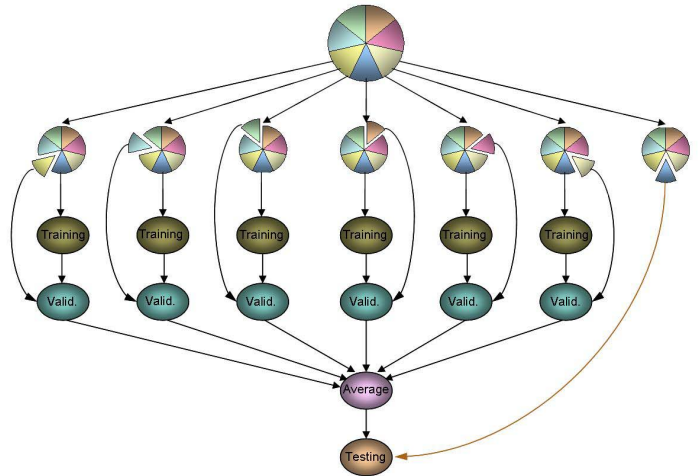


Figure 5 6-fold cross validation technique

each  $c_a \in C_A$  and  $c_b \in C_B$  are already computed. In a real life setting, the queries will be posed by either humans (data center support staff), or by software (autonomic managers using a library of preprogrammed queries).

Let  $\mathcal{R}_A$  and  $\mathcal{R}_B$  be the result sets of queries  $\mathcal{Q}_A$  over  $O_A$  and  $\mathcal{Q}_B$  over  $O_B$ , respectively. Given classes  $c_a \in C_A$  and  $c_b \in C_B$ , we use  $\mathcal{R}_A$  and  $\mathcal{R}_B$  to re-estimate  $n(c_a)$ ,  $n(c_a, c_b)$  and  $P(c_a, c_b)$  as follows:

1. Compare the instances in  $\mathcal{R}_A$  with those in  $\mathcal{R}_B$  to
2. discover pairs of objects  $r_a \in \mathcal{R}_A$  and  $r_b \in \mathcal{R}_B$  such that  $r_a \equiv r_b$ .
3. Compute  $\delta(c_a, c_b)$ , the number of occurrences of pairs of objects  $r_a \in \mathcal{R}_A$  and  $r_b \in \mathcal{R}_B$  such that  $r_a \equiv r_b$  and  $c_a$  and  $c_b$  are the classes under which  $r_a$  and  $r_b$  are classified, respectively. ←
4. Compute  $\delta(c_a)$ , the number of occurrences of  $c_a$  in  $\mathcal{R}_A$ .
5. Recompute  $P(c_a, c_b)$  using Equation (2):

$$P(c_a | c_b) = \frac{\Delta(c_a, c_b) + \alpha [n(c_a, c_b) + \Psi]}{\Delta(c_a) + \alpha [n(c_a) + 1]} \quad (2)$$

Where:

$\alpha$  is a coefficient that takes values from the set  $\{0.01, 0.1, 0, 1, 10, 100\}$ , calibrated empirically

$\Delta(c_a, c_b)$  = current number of occurrences of  $c_a$  and  $c_b$   
 $\Delta(c_a)$  = current number of occurrences of  $c_a$   
 $n(c_a, c_b)$  = total number of occurrences of  $c_a$  and  $c_b$   
 $n(c_a)$  = total number of occurrences of  $c_a$

←

and  $\Psi$  a smoothing coefficient, calculated as follows:

$$\Psi = \frac{1}{|C_B|}$$

6. Add  $\delta(c_a, c_b)$  to  $n(c_a, c_b)$  and add  $\delta(c_a)$  to  $n(c_a)$ .

Note that the above procedure is symmetric in  $c_a$  and  $c_b$ . Hence, the entire process can be easily adapted to compute estimations for the frequency in which classes in  $C_B$  map into terms in  $C_A$ .

The  $\alpha$  coefficient is used to compensate the existence of duplicated entries between queries.

The next step is calibrating the parameters of the mapping rate estimation model, i.e., determining a threshold value above which we can assume mappings between pairs of classes to be reliable.

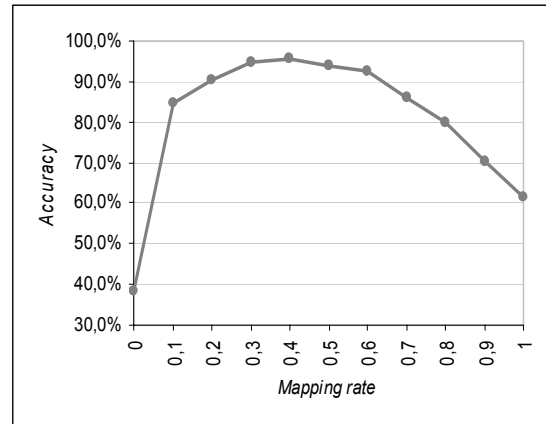
#### 4. Model Calibration

To calibrate the parameters of the mapping rate estimation model, the data collected was partitioned into 7 datasets as depicted in Figure 5.

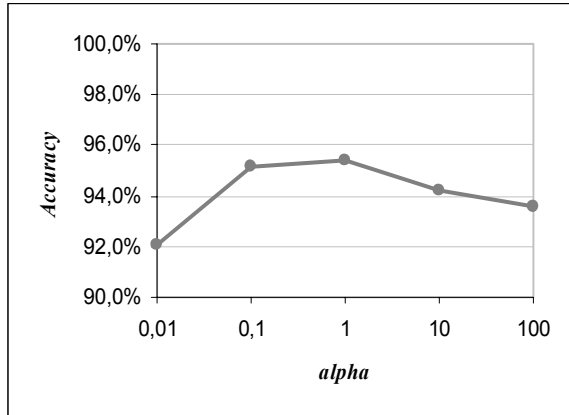
In order to apply the 6-fold cross-validation technique we partitioned the tune set into a validation and a training set. The 6-fold technique consists in alternating the validation sets with the training sets to guarantee an

averaged precision to the model [Geisser06] To accomplish this, the datasets (Ex) from 1 to 6 were replicated and manually labeled with true or false for each occurrence of the terms pairs, creating the validation sets (Val). The labeling was made by comparing thesauri descriptions, with the help of a domain expert.

To validate the model, the frequencies of the pairs of terms of the training sets were estimated many times, varying the mapping rates from  $\{0, 0.1, 0.2, \dots, 1\}$  and the  $\alpha$  coefficient from  $\{0.01, 0.1, 0, 1, 10, 100\}$ . The cross-validation compares these results with the labelled pairs from each validation set. During the validation, we count the number of successful estimated mapping rates in the training sets pairs of terms. As showed in Figure 6, better results were obtained with  $\alpha = 1$  and mapping rate 0.4 to the cross validation from  $C_A$  to  $C_B$ .



(a)  $\alpha = 1$



(b) mapping rate = 0.4

**Figure 6. Model calibration results**

Finally to test the mapping rate estimation model we use the test set (dataset 7) and the parameters calibrated in model calibration step:  $\alpha = 1$  and mapping rate 0.4

Initial results, obtained for the ontologies depicted in Figures 7 a and b are encouraging. We have obtained 89.7% of accuracy and 81.3% of recall when mapping from  $O_A$  and  $O_B$ . We illustrate some mapping rates as follows, please note that pairs with values under 0.4 were not considered reliable mappings.

```
P(NT_Physical Disk, Win32_CDROMDrive) = 0.2258
P(NT_Physical Disk, Win32_FloppyDrive) = 0.3226
=====
P(NT_Physical Disk, Win32_PhysicalMedia) = 0.4516
P(NT_Printer, Win32_Printer) = 1
P(NT_Processor, Win32_Processor) = 1
```

## 5. Conclusion

In this paper we proposed an instance based matching technique that complements the ontology integration approach proposed by the CATO engine. This approach improves matching results in situations where the original ontologies are either poorly specified or incomplete. The proposed approach uses a query probing technique that consists of exhaustively sending keyword queries to original ontologies [Wang04]. Further analysis of the results using learning algorithms and statistical analysis provides indication of good matches. A limitation of the approach is its dependency on large numbers of instances. Scarcely populated ontologies will not provide the necessary information for the statistical analysis and may result in bad matches.

Our goal is to allow for the design of an autonomic manager component that is capable of serving as a mediator among applications that implement different “flavors” of DTMF’s CIM standard. This component will allow users to choose a vocabulary of their preference, whether being ITM’s

version of CIM, HPs, CAs and so on and so forth, rather than forcing applications to adopt a canonical standard. To solve semantic conflicts the proposed instance-based approach focuses on estimating weighted relationships between classes of distinct ontologies. To achieve this goal we propose the instance-based matching approach, in which we collect statistics about the instances returned by queries to the ontologies.

The proposed approach could be generalized to any domain that provides a reliable substitute for an unique instance identifier. In the geographic information systems domain, for example, there are various geo-referencing schemes that associate geographic object with a description of its location on the Earth’s surface. This location acts as a universal identifier for the object, or at least an approximation thereof. We have successfully applied this approach to build mediators for Geographic Data Catalogs [Brauner 06, Brauner07, Gazola07].

The contribution to autonomic computing resides in providing associated management functions to aid the task of consulting, searching, retrieving and integrating heterogeneous knowledge resources. The use of formal ontology to represent the resources, allied to the CATO engine and instance-based matching strategy, provide support to the monitoring and analyzing tasks of the autonomic control loop. It supports monitoring by filtering and reporting to other autonomic managers that are compatible with the current request or need. Our solution also supports the analysis task by the correlation and integration of distributed autonomic managers. Finally, the proposed solution enables resource integration in the real world, where information is often incomplete or lacking.

## Acknowledgments

This work is partially supported by CNPq under grants 550250/2005-0, 551241/2005-5 and by the IBM Corporation.

## 6. References

[Brauner06] – Brauner, Daniela F ; Casanova, M. A. ; Milidiu, Ruy . Mediation as Recommendation: An Approach to Design Mediators for Object Catalogs. In: 5th International Conference on Ontologies, DataBases, and

- Applications of Semantics (ODBASE 2006), 2006, Montpellier.
- [Brauner07] - Brauner, D. F.; Intrator, C.; C.Freitas, J.; Casanova, M. A. . An Instance-based Approach for Matching Export Schemas of Geographical Database Web Services. In: IX Brazilian Symposium on GeoInformatics, 2007, Campos do Jordão. Proceedings of the IX Brazilian Symposium on GeoInformatics. Porto Alegre : Sociedade Brasileira de Computação - to appear, 2007.
- [Breitman06] - Breitman, K.K.; Truszkowski, W.; Felicissimo, C.H. (2006) The autonomic semantic desktop: Helping users cope with information system complexity. In: Proceedings of the Third IEEE International Workshop on Engineering of Autonomic and Autonomous Systems EASE 2006, pp. 156–162.
- [Breitman07] - Breitman, K. K.; Perazolo, M. . Using Formal Ontology Representation and Alignment Strategies to Enhance Resource Integration in Multi Vendor Autonomic Environments. In: Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems EASE 2007, 2007, Baltimore, USA. Proceedings of the Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems EASE 2007. Los Alamitos : IEEE Computer Society Press, 2007. p. 117-126.
- [Casanova07] - Casanova, M. A. ; Breitman, K.; Brauner, D. F. ; Marins, A. . Database Conceptual Schema Matching. Computer (Long Beach), v. 40, p. 102-104, 2007.
- [DTMF07] - DTMF - Distributed Management Task Force – Common Information Model Standards - <http://www.dmtf.org/standards/cim/>
- [Ehrig05] - Ehrig, M. Staab, S. Sure, Y.: Bootstrapping Ontology Alignment Methods with APFEL - Proceedings of the 4th International Semantic Web Conference (ISWC 2005) - Galway, Ireland - November, 2005. pp. 186-200
- [Ehrig06] – Ehrig, M. - *Ontology Alignment: Bridging the Semantic Gap* Semantic Web and Beyond series – Springer Verlag, 2006.
- [Felicissimo04] - Felicissimo, C.H.; Breitman, K.K. (2004) Taxonomic ontology alignment - An implementation. In: Proceedings of the Seventh Workshop on Requirements Engineering, Tandil, Argentina, Dec. 9–10.
- [Gazola07] - Gazola, A. ; Brauner, D. F ; Casanova, M. A. . A Mediator for Heterogeneous Gazetteers. In: XXII Simposio Brasileiro de Banco de Dados, 2007, João Pessoa. Anais do XXII Simposio Brasileiro de Banco de Dados - SBBD - Sessão de Pôsteres. Porto Alegre : Sociedade Brasileira de Computação, 2007. v. 1. p. 11-14
- [Geisser06] – Geisser, S. - *Modes of Parametric Statistical Inference*, Wiley, 2006
- [Hage05] -Hage,W. et al - A method to combine linguistic ontology mapping techniques - Proceedings of the 4th International Semantic Web Conference (ISWC 2005) - Galway, Ireland - November, 2005. pp. 732-744.
- [IBM06] - An Architectural Blueprint for Autonomic Computing" - made available by theIBM Corporation, October <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>
- [Kephart03] - Kephart, J. and DM Chess, “The vision of autonomic. computing”, IEEE Computer, Jan. 2003, pp. 41-50.
- [Miller05] - Brent Miller - The autonomic Computing Edge: The role of knowledge in autonomic systems - September 2005 - <http://www-128.ibm.com/developerworks/autonomic/library/ac-edge6/index.html>
- [Noy03] - Noy, N. F., Musen, M. A.: The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. International Journal of Human-Computer Studies, 2003.
- [Perazolo2006] Perazolo, M. (2006) – Symptoms Deep Dive 3: Classify your Symptoms – <http://www-128.ibm.com/developerworks/autonomic/library/ac-symptom3/>.
- [Sterritt05] - Sterritt, R; Rouff, C.; Rash, J.; Truszkowski, W.; Hinchey, M. - Self-\* Properties in NASA Missions - 4th International Workshop on System Software Architectures- Las Vegas, NV -2005
- [Stoilos05] - Stoilos, G.; Stamou, G.; Kollias, S. (2005) A string metric for ontology alignment. In: Proceedings of the Fourth International Semantic Web Conference (ISWC 2005), Galway, Ireland, Nov. 2005, pp. 624–637.
- [Stojanovic04] - Stojanovic et al - The role of ontologies in autonomic computing systems - IBM Systems Journal - Vol 43 No 3 2004 - pp.598-612
- [Wang04] - J. Wang, J.-R. Wen, F. Lochovsky, and W.-Y. Ma, “Instance-based schema matching for web databases by domain-specific query probing”. In: Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.
- [Zhang05] - ]Zhang, Dell; Lee, Wee Sun: Learning to Integrate Web Taxonomies - Journal of Web Semantics Volume 2 Issue 2, Elsevier, July 2005.