Integrating Ontologies

Workshop Proceedings

K-Cap Conference

October 2, 2005

Banff, Canada

Table of Contents

The Integrating Ontologies Workshop at K-CAP 2005 **1**
Benjamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt

Reverse Leibniz, and then Bend It Like Beckham: Temporal Ontology Mapping as Problem-Solving Method **2**
Hans Akkermans

Towards Browsing Distant Metadata Using Semantic Signatures **10**
Andrew Choi and Marek Hatala

Semantic Association of Taxonomy-based Standards Using Ontology **18**
Hung-Ju Chu, Randy Y. C. Chow, Su-Shing Chen, Raja R.A. Issa, and Ivan Mutis

Relaxed Precision and Recall for Ontology Matching **25**
Marc Ehrig and Jérôme Euzenat

Searching Web Resources Using Ontology Mappings **33**
Dragan Gasevic and Marek Hatala

GMO: A Graph Matching for Ontologies **41**
Wei Hu, Ningsheng Jian, Yuzhong Qu, and Yanbing Wang

Towards Semantic Based Information Exchange and Integration Standards: the art-E-fact ontology as an extension to the CIDOC CRM (ISO/CD 21127) Standard **49**
Carlos Lamsfus, María Teresa Linaza, and Tim Smithers

An approach to ontology mapping negotiation **54**
Nuno Silva, Paulo Maio, and João Rocha

Introduction to the Ontology Alignment Evaluation 2005 **61**
Jérôme Euzenat, Heiner Stuckenschmidt, and Mikalai Yatskevich

FOAM – Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative **72**
Marc Ehrig and York Sure

CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest **77**
Yannis Kalfoglou and Bo Hu

FalconAO: Aligning Ontologies with Falcon **85**
Ningsheng Jian, Wei Hu, Gong Cheng, and Yuzhong Qu

oMAP: Results of the Ontology Alignment Contest **92**
Umberto Straccia and Raphael Troncy

OLA in the OAEI 2005 alignment contest **97**
Jérôme Euzenat, Philippe Guégan, and Petko Valtchev

The Integrating Ontologies Workshop at K-CAP 2005

Benjamin Ashpole
Lockheed Martin Advanced Technology Laboratories

Marc Ehrig
University of Karlsruhe

Jérôme Euzenat
INRIA Rhône-Alpes

Heiner Stuckenschmidt
Vrije Universiteit Amsterdam

INTRODUCTION

The Integrating Ontologies Workshop is a forum for researchers and application developers from the area of ontology interoperability to exchange knowledge, ideas, approaches, and challenges for handling multiple competing ontologies. The workshop will facilitate methodological and technical discussions.

For many knowledge domains, a variety of ostensibly "standard" ontologies have been engineered, learned, and extended. Each is an interface for a similar purpose yet uses different nomenclatures. To enable collaboration within and across application domains, software agents require transparency between the various formalisms. This requires both semantic alignment and syntactical translation. Purely manual approaches are error-prone, onerous, and insufficient to support dynamic systems interoperability.

However, recent research in ontology alignment exploits "meaning" that is explicit and implicit in these formalisms. If heterogeneity can be mitigated with minimal use of standards by way of partially or fully automated alignment, then the integration of and for commercial, non-profit, military, and government systems will be simplified and improved. This workshop will exhibit new approaches to alignment, mediation, and other methods that promise to help fulfil the vision of the Semantic Web.

K-CAP'05, October 2-5, 2005, Banff, Canada.

Like any software research endeavor, the study of automated ontology alignment will most clearly demonstrate progress through rigorous experimentation. The ontology alignment research community has embraced this challenge, having conducted collaborative experiments to compare their respective alignment tools on a standard set of ontology pairs. The results of these experiments have established clear performance benchmarks and informed new approaches to ontology alignment.

The Integrating Ontology Workshop includes discussion of the third such experiment since 2004, following the Information Interpretation and Integration Conference[1] and the Evaluation of Ontology-based Tools Workshop.[2]

This workshop also features research presentations describing the latest efforts for ontology alignment and mediation.

Further information on the Integrating Ontology Workshop and the ontology alignment experiment can be found on the workshop website.[3]

Thanks to all the members of the program committee, authors, experiment participants and local organizers for their efforts. The workshop represents significant cooperation and progress on many levels.

[1] http://www.atl.external.lmco.com/projects/ontology/i3con.html

[2] http://km.aifb.uni-karlsruhe.de/ws/eon2004/

[3] http://km.aifb.uni-karlsruhe.de/ws/intont2005

Reverse Leibniz, and then Bend It Like Beckham: Temporal Ontology Mapping as Problem-Solving Method

Hans Akkermans
AKMC Knowledge Management BV
and
Free University Amsterdam VUA, Business Informatics Department
Amsterdam, The Netherlands
email: Hans.Akkermans@akmc.nl

ABSTRACT

I discuss and construct ontology mappings between different ontologies of time. I show how you can use them as a new method to solve significant dynamics problems, by exploiting the properties of the ontology mapping. A unique feature of a nonlinear ontology mapping I propose is that it can rigorously treat infinitesimals as strictly finite computational quantities. The approach also suggests some novel, I believe intriguing, insights into the nature of time, particularly regarding "density" and "curvature" of time. The paper provides an in-depth case study in ontology mapping, offering some evidence that ontology building, mapping, and reuse is much a substantive issue, more than a matter of generic representation language and semantic tooling.

Categories and Subject Descriptors

H.4: Information Systems Applications – *Miscellaneous.*

General Terms

Theory, Algorithms.

Keywords

Time, ontology, mapping, signal analysis, dynamic systems

INTRODUCTION

Time is a very generic upper-level ontological concept. There are many different ontologies of time [1], but the two temporal ontologies most widely used [2, 3] in science and engineering are *point-based*: continuous time and discrete time. In continuous-time systems, time is represented by a real-numbered parameter $t \in \Re$. In discrete event-based systems, time is represented by a "step" variable $S \in \aleph$, i.e. an integer.

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

Continuous and discrete approaches represent two very different ontological viewpoints on the same concept of time. They not only differ in appearance, but also come with radically different concepts and methods, witness the mathematical and computational analysis of continuous versus discrete systems, for which there exists a vast literature spanning several centuries (e.g., [4] and [5]).

From the computational perspective, there is the additional problem that continuous analysis is based on the notion of derivatives and infinitesimal quantities (differential calculus dating back to Leibniz's 1684 article [4]). As the computer is an inherently *discrete* machine, computer methods for *continuous* systems invariably introduce approximations that are in fact a kind of systematic error (known as discretization or truncation error [2, 3]). In this paper I pose – and solve – the problem: can we construct an ontology mapping between continuous and discrete ontologies of time, which is both mathematically rigorous and computationally adequate, *and* is able to avoid systematic error in changing from one temporal perspective to the other?

More simply: *is it possible to reformulate any given form of continuous-time dynamics in discrete time, rigorously, without any compromise or computational approximation?*

The answer to this question is yes; the ontology mapping solution outlined in this paper entails a novel method that I call the T transform. Important characteristics of this new transform method are: (1) conceptually, it is a radical departure from the traditional view and techniques regarding the relationship between continuous and discrete time; (2) it succeeds in fundamentally avoiding computer-introduced systematic error in handling differential calculus; (3) it has informational advantages, by generating certain important systems information directly that is not so easy to obtain by conventional methods; (4) it gives rise to several new and elegant discrete algorithms for systems analysis; and (5) it has an extremely wide spectrum of applications and generalizations (even beyond time).

I will go through these aspects below in brief.

"NAIVE DYNAMICS": TEMPORAL ONTOLOGIES AND THEIR MAPPING

Axiomatization of Time Ontologies

Van Benthem [1] gives a tense-logical formalization of a great variety of temporal ontologies. His axiomatization for point ontologies of time is over temporal structures consisting of a non-empty set of time points ordered by a binary precedence relation <. It contains the following *shared* axioms for discrete and continuous time:

- TRANS: time ordering is transitive.
- IRREF: the property of irreflexivity; together TRANS and IRREF model the (asymmetric) notion of the flow (or arrow or "river") of time.
- LIN: linearity, expressing that time structures have a single path (or river flow bed) without branching.
- SUCC: time has no end point (continuing succession towards the future).

The difference between discrete and continuous time comes with the choice of a final temporal axiom, *either one* of the following two options:

- DENS: infinite divisibility of time, i.e. between any two time points there is always another one.
- DISC: discreteness; time is not infinitely divisible, but has the property of "stepwise" succession.

This suffices as axiomatization of the point-based temporal ontologies I consider in this paper. Van Benthem shows that this axiomatization is syntactically complete. He also shows that it admits of several models. Thus, real-numbered time $t \in \Re$ (where the axiom DENS is implied by the stronger continuity axiom CONT) and discrete event-based integer time $S \in \aleph$ (where as we shall see S indeed can be usefully read as "step") are a specific model choice for the above ontological theories. However, these are by far the most common and useful ones in scientific practice, and that's why I stick to them.

Clearly, the above two formal temporal ontologies are rather concise and simple themselves. This turns out not to be the case for discrete-continuous ontology mappings, however. I will now proceed to show that (1) temporal ontology mappings are important general constructs with many practical applications and implications, but (2) they are not unique, as several different useful ontology mappings can be constructed.

Standard Time Ontology Mapping

Let us consider first the traditional approach to continuous-discrete temporal ontology mapping, which is entrenched in today's standard techniques for numerical analysis and simulation of system dynamics and evolution [2]. Typically, one assumes that the discrete time steps or events S = 0, 1, 2, ... are embedded in continuous time $t \in \Re$ by assuming that the *integer* time point "1" (etc.) maps onto the *real* time point "1.000..." (etc.), as depicted in Figure 1. This looks very logical and natural indeed: formally the standard time ontology mapping between continuous and discrete time is (note: both ways) given by the simple *linear* function:

$$t / \tau = S \quad \text{or} \quad t = S\tau, \ \text{so that} \tag{1a}$$

$$X_S = x_{t/\tau} \ . \tag{1b}$$

This equation is the ontological explication of the standard operating procedure in conventional real mathematical-numerical analysis. Here, τ denotes the free (user-selectable) parameter known as the "stepsize" in continuous systems simulation.

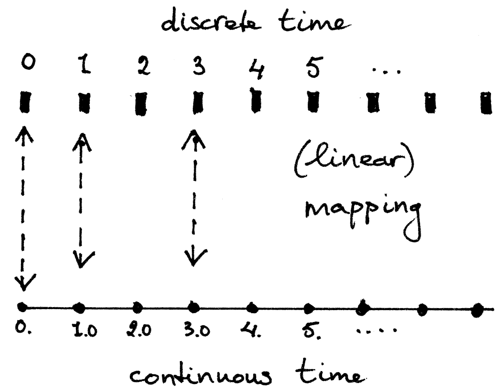


Figure 1. Traditional view on the mapping between continuous and discrete time.

How does this linear ontology mapping work in practice? Let us take a look at *the* prototypical formulation of continuous dynamic systems, viz., the ordinary differential equation[1] (ODE):

$$d/dt \ x_t = f(x_t) \tag{2}$$

As this is an equation in continuous time involving, moreover, infinitesimal calculus, it is not suitable for direct computer treatment. The standard approach then is to discretize the ODE (2) in time. The simplest choice to do so is

---

[1] It is conceptually interesting to reread Leibniz's original article of 1684 [4]. He clearly talks about *dx* and *dt* as *finite* differences, and then proceeds with stating the rules of differential calculus *as if* they are infinitesimals. He does not offer any justification; in actual fact his rules are incorrect for finite quantities (they neglect the higher-order differences that vanish in the infinitesimal case). He is sufficiently self-confident (or arrogant) to simply ignore this fundamental problem, and then saves the day by coming up with an important useful application example (he derives the refraction law of Snellius directly from Fermat's principle). So it seems he was lucky to live in ancient times as his article would be unlikely to survive any modern peer review. No wonder that people like the idealist philosopher Bishop Berkeley (1734) made fun out of the believers in this new calculus, deriding it as strange magic by juggling with different kinds of zeros (he added the wider point, against non-religious rationalists such as Halley, that if you do believe in this weird calculus stuff ("ghosts of departed quantities"), you also rob yourself of the right to criticize matters of theology). Proponents, among them Bernoulli, Euler, Maclaurin, D'Alembert, etc. ultimately defeated him by striking back with proper theoretical foundations of the new calculus. Interestingly, this whole conceptual struggle, involving a long string of the brightest mathematical geniuses of their time, took more than one century and a half.

by dropping the infinitesimal limit in the definition of the derivative and invoking Eq. (1b). Then

$$(x_{t+\tau} - x_t) / \tau = f(x_t) \qquad (3a)$$

or

$$\Delta X_S \equiv X_{S+1} - X_S = \tau\, f(X_S) \qquad (3b)$$

Equation (3) is computationally a one-step forward difference, generally known as the *Euler algorithm*.

The Euler formula is essentially a direct application of the standard continuous-discrete time ontology mapping of Eq. (1) and Figure 1 (where I have assumed that the step size τ = 1; this simplifies the bookkeeping and can be done without loss of generality). It replaces the continuous dynamic system by a discrete-time one that is easy to compute (here, by a one-step forward recursion). It is not really used in practice, precisely because it nicely illustrates a key problem of the digital computation of continuous dynamics: it is inherently approximate (the systematic error mentioned earlier: the higher-order differences ignored by Leibniz can no longer be neglected in a finite computation).

Nevertheless, Euler's formula is rightfully seen as the grandfather of all ODE solving algorithms. Any ODE solver attempts to correct its shortcomings, lack of accuracy and sometimes also of stability, by including higher difference contributions (equivalently, orders of the Taylor expansion) up to a prespecified order. As there are zillions of ways to do this, each with specific advantages and drawbacks, this has become a computing art in itself. The famous Runge-Kutta algorithms, the universal workhorse to simulate continuous dynamic systems, are a case in point. However, this can only be done to a limited extent, as explained in a very accessible and practical way in [2]. In essence, the standard linear ontology mapping of Eq. (1) inherently and unavoidably introduces approximations in the digital computation of what basically are infinitesimal quantities.

"Naive Dynamics"

Although never stated this way, the key problem of the standard algorithms for continuous dynamics transformed to a computationally tractable discrete-time system is therefore the underlying assumption of a linear mapping between time ontologies.

With an allusion to Hayes's "Naive Physics Manifesto" (1978/85), one might say that what makes the standard linear ontology mapping attractive is that it leads to a simply understandable form of "naive dynamics" for complex (nonlinear) systems of the type (2), witness Eqs. (1) and (3). Equations (1) and (3) are both nice to have from the standpoint of naive dynamics. Unfortunately, scientific history has demonstrated that they cannot both be valid simultaneously. The standard approach then makes the choice that the linear time ontology mapping (1) is correct, but precisely this assumption invalidates the basic discrete Euler formula (3) and its descendants such as Runge-Kutta for infinitesimal calculus proper. Correcting for this is what makes the usual algorithms for continuous dynamics so complicated (or non-naive).

Now, my aim is to retain in some form this idea of naive dynamics. I will do this in a novel way, in fact the precise opposite of the standard computational approach. Specifically, I will start from the principle of the correctness of an Euler-type formula as (3). The key reason is that, if you succeed in doing this, computation and prediction of continuous systems is extremely simple, since Eq. (3) is a one-step forward difference in discrete time, and the whole future is predicted (without any approximation in the sense of built-in systematic bias, as in the standard approach) by repeated application of (3).

The necessary consequence of this alternative route is that one has to drop the correctness of the linear ontology mapping (1). However, as I will show, there is no principal reason why there can't be alternative ontology mappings with beautiful and desirable conceptual and computational properties – but consequently they must be nonlinear with respect to time. In other words, you have to "bend" time.

T: THE TRANSFORMATION OF TIME

Probabilistic Embedding of Events in Time

I now construct a new alternative class of temporal ontology mappings by means of the following procedure that embeds discrete events S in continuous time t. Imagine that the time difference (in continuous time) between the occurrence of two subsequent (discrete) events is not fixed and constant, as in the traditional approach (cf. the constant τ in Eq. (1)), but actually is *random*. So, after the start event S=0 that occurs at some given start time t_0 (taken to be t=0 in the remainder), the discrete time events S >0 occur randomly at continuous time points t_S, and the time intervals between two steps $T_1=t_1-t_0$, ..., $T_{S+1}=t_{S+1}-t_S$ are all random variables, governed by some given probability distribution (which I assume to be the same for all events).

Accordingly, let $P(t, S)$ be the probability that in the interval $[0,t]$ precisely S discrete events or steps have occurred. Then the time ontology mapping replacing Eq. (1) reads:

$$x_t = \sum_{S=0}^{\infty} P(t,S) \cdot X_S \qquad (4)$$

Generally, this is a *nonlinear* ontology mapping, with respect to both time variables t and S. Equation (4) actually represents a whole *class* of ontology mappings, because there are many choices for the probability function $P(t, S)$.

For this probability I now take a specific choice, namely:

$$x_t = \sum_{S=0}^{\infty} \frac{1}{S!}\left(\frac{t}{\tau}\right)^S e^{-t/\tau} X_S \qquad (5)$$

This nonlinear ontology mapping ($x_t = T(X_S)$ in short) embeds discrete events in continuous time by means of a stochastic process known as the Poisson process. Although fundamentally different from Eq. (1), it likewise has an elegant conceptual interpretation. The linear ontology mapping (1) essentially says that all discrete events occur *totally correlated* in continuous time: once we know the time instant of the initial event and the (fixed) waiting time constant τ between events, the time occurrence of all events is wholly fixed, carved in stone with military precision as it were (cf. Figure 1).

In contrast, the nonlinear ontology mapping of Eq. (5) essentially represents the opposite situation, in which all events occur *independently* and so are totally *un*correlated. This situation often occurs in reality. For example the arrival of incoming phone calls at a helpdesk is expressed by a Poisson process. Calls arrive not with fixed time intervals between them but irregularly; the probability distribution for the random time T_S between two steps is a negative exponential, and the constant τ in Eq. (5) now represents the *average* waiting time between two subsequent events.

What does this buy us? In essence, the time ontology mapping is a transform expression – the case of Eq. (5) I call the T or $\mathtt{T}$ transform – that transforms a continuous function x_t into a discrete function X_S. Transform methods are well-developed: they already stem from early 19th century mathematics, the Laplace and Fourier transforms probably the best known ones. Although mathematically demanding, their key idea is simple: if you can map the original *problem* (say, the ODE (2)) from the original space (here, continuous time) into a different problem in a new space where it is simple to solve, then you are done by simply back-transforming the found *solution* to the original space. This is what for example the Laplace transform does: it transforms differential equations from continuous time into simple-to-solve algebraic equations in frequency space. But you already find this transform idea in the solution of the mutilated chessboard problem, or in that of the children's game called Nim.

My transform idea expressed in Eq. (4) and in the T or $\mathtt{T}$ transform (5) is new and special in the sense that it transforms a problem formulated in continuous time into one that is formulated in discrete time. Discrete problems are much more suitable for solution by a computer than continuous ones; once the discrete solution is found we simply back-transform it into the continuous solution we are actually looking for by using (4) or (5).[2] That this idea practically works I am going to show now.

Key Properties of the T Transform

Some key properties of my T ($\mathtt{T}$) transform between discrete and continuous time are given in Table 1.

Table 1. Properties of the T transform Eq. (5)

Property No.	Continuous-time function $x_t = T(X_S)$	Discrete-time function $X_S = \mathtt{T}(x_t)$
I.	1 (constant)	1 (constant)
II.	t	S
III.	t^2	S(S-1)
IV.	t^3	S(S-1)(S-2)
V.	t^n	S! / (S-n+1)!
VI.	e^{At}	$(1 + A)^S$
VII.	$A\, x_t + B\, y_t$	$A\, X_S + B\, Y_S$
VIII.	$d/dt\, x_t$	$\Delta X_S \equiv X_{S+1} - X_S$
IX.	$d^n/dt^n\, x_t$	$\Delta^n X_S$
X.	$f_t \equiv y_t \times x_t$	$F_S = \sum_{n=0}^{n=S} [S!/((S-n)!n!)]\, \Delta^{S-n} Y_0 \times X_n$

Proofs. There are several possible derivations of the properties in Table 1, but they require some background in real mathematical analysis. Property I immediately follows from the observation that the sum of *P(t, S)* over all steps equals unity by definition, because it is a probability function. Property VII also follows immediately by direct algebraic manipulation. To prove property VIII, we differentiate both sides of Eq. (5) with respect to *t* and rearrange terms at the right-hand side, with a simple change of discrete-time variable *S*. Property IX then follows (for example) by repeating this procedure and complete induction towards the order of differentiation. Properties II-V all follow from differentiating Eq. (5) and invoking I, VIII and IX. Finally, properties VI and X are discussed in more detail in the next section in the context of various ODE applications. □

The first property (No. I) is interesting in that it conceptually implies that any constant of the motion in continuous time (think of energy, momentum, angular momentum, probability, flux) is also a constant of the motion in discrete time. The second property (No. II) states that linear functions in continuous time transform to linear functions in discrete time. These are properties that the nonlinear ontology mapping (5) shares with the linear one of Eq. (1).

[2] For the real connoisseur, I mention in passing that the linear ontology mapping of Eq. (1) can be interpreted, like my T transform (5), as a special case of the probabilistic transform (4). It is the limiting case in which the waiting-time distribution between events is the Dirac delta function $\delta(t-\tau)$. Reworking Eq. (4) on this basis by using its Laplace transform, one is led to a generating function method known as the *z* transform, which in digital control theory is also sometimes called the discrete Laplace transform. It directly yields the linear ontology mapping of Eq. (1b). Hence – although this is hardly ever explicitly recognized – also the standard numerical approach using the linear ontology mapping Eq. (1) is fundamentally based on a transform idea.

Other properties are unique to my T transform (5). In particular, continuous-time functions map onto similar (e.g. same-order polynomials, cf. properties III-VI) but not identical functions in discrete time. This is in stark contrast to the assumption in the standard received view that employs the *same* function in both continuous and discrete time (cf. Eq. (3a)). Property VII says that the T transform is a linear transform[3].

Properties VIII and IX are the crucial ones: the T transform maps the derivative d/dt onto a *finite* first-order discrete forward difference Δ. Hence, Euler-type formulas similar to (3) will be correct under the T transform. Moreover, this extends to the higher-order derivatives, which are simply found by repeated application of the finite forward difference Δ. As a consequence, a beautiful and important property of the nonlinear time ontology mapping (5) is that it in discrete time produces the higher-order derivatives of continuous time, one by one and exactly. The production of the X_S values in discrete time yields a tableau (see Figure 2), by simple subtraction or addition, that contains all desired information.

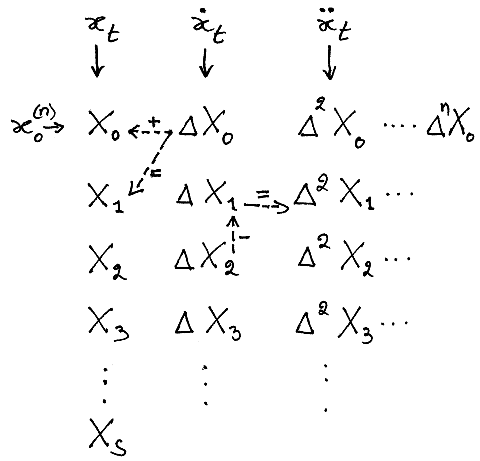


Figure 2. The T transform yields a tableau that contains the discrete solution to derivatives of any order.

If for example x_t denotes the position in a space at a certain time, the tableau not only gives the solution for the location (coefficients X_S) but it *simultaneously* solves the question as to its velocity (ΔX_S), acceleration ($\Delta^2 X_S$), etc. In addition it is able to reconstruct, by using only the T transform equation (5), the values of the continuous variables at *any* desired point in continuous time *t*. These are all major informational and computational advantages that show the power of the T transform temporal ontology mapping.

Leibniz Reversed

Consequently, we have achieved the earlier stated aim of "naive dynamics" by showing the validity of the one-step forward difference formula (3b) as the correct expression for differentiation of a continuous variable. In a sense, we have achieved this by conceptually reversing Leibniz. Leibniz talked about infinitesimals as finite quantities, and subsequently invented the correct rules of differential calculus. We took differential calculus, and subsequently invented a temporal ontology mapping that makes it actually *correct* to treat infinitesimals as finite quantities, just by switching from continuous to discrete time!

A FEW APPLICATIONS AND IMPLICATIONS

Computer Right, Man Wrong (Save Euler)

I first show how you can solve large-scale linear differential systems by temporal ontology mapping. This turns out to have an interesting side implication on the conceptual interpretation of what algorithms do.

A widely used special case of the ODE (2) is the linear system:

$$d/dt\, x_t = A\, x_t \qquad (6)$$

This equation also describes dynamic systems in many dimensions; then, *A* is not to be interpreted as a one-dimensional constant (scalar) but as a matrix. The derivation below is then generally valid for any number of dimensions.

To solve this by ontology mapping, we first transform the problem from continuous time to discrete time. Using property VIII of Table 1, the discrete version of Eq. (6) is:

$$\Delta X_S = A\, X_S \qquad (7)$$

Next, we construct the solution in discrete time starting from the known initial condition $x_0 = X_0$, and repeatedly applying the forward difference definition of the operator Δ. In effect, the whole discrete solution is stepwise produced (also in many dimensions) by the Euler algorithm (3b). From Eq. (7) it is easy to see that the discrete solution is:

$$X_{S+1} = (1 + A)\, X_S \;\Rightarrow\; X_S = (1 + A)^S\, X_0 \qquad (8)$$

Finally, this solution is back-transformed to continuous time by using Eq. (5). In the general case this can be done computationally by various methods (e.g. by successive sequences of one-step recursions or, parallelized, by matrix methods), where as a bonus you have a free choice for the time points *t* you are actually interested in. In the present case, the continuous solution is just a matter of simple table look-up, see property VI in Table 1. Hence, the ontology mapping method solves the dynamic problem (6) by first transforming the problem to a new (discrete) space, next solve it there, and then transform this solution back to the original (continuous) space, where it reads:

$$x_t = e^{At}\, x_0 \qquad (9)$$

So, we have solved a problem involving infinitesimal calculus in a strictly discrete fashion, not by directly attacking

[3] Perhaps this sounds a bit confusing, but linearity is only a *relative* notion. As stated earlier, the T transform is nonlinear with respect to the time variables *t* and *S* (see Eq. (5)). With respect to the temporal functions X_S and x_t, however, it is linear, in accordance with property VII.

the computation of derivatives in an approximate fashion (which is the standard way of doing it), but indirectly by changing the problem space first. I mention in passing that the above also yields a proof of property VI in Table 1: just insert Eq. (8) into Eq. (5) and carry out the summation.

The case discussed here has several important general applications. For example, it applies to both random walks and master equations; both have many practical applications in many different disciplines. As a bonus, the T transform proves that they are their mutual discrete and continuous-time equivalents, see also [6].

A possibly even broader application is that it is applicable to the modern state-space approach to control systems theory: adding a control signal term to Eq. (6), i.e. a function explicitly dependent on *t*, yields the fundamental systems formulation underlying control engineering of multi-dimensional continuous systems. The methods developed in this paper open up the opportunity to treat such systems by strictly discrete computer methods.

The above results give some (I believe entertaining) rehabilitation of the Euler algorithm. Let me quote a statement from [2], a remark that is prototypical for any modern textbook treatment of numerical methods: "There are several reasons that Euler's method is not recommended for practical use, among them, (i) the method is not very accurate (...), and (ii) neither is it very stable" ([2], p. 704). In conflict with this statement, the dynamic problem (6) *has* been solved here exactly by the Euler method. However, you should interpret the results of the algorithm not as rough *direct* estimates of the continuous-time point solution (see Eq. (3a), the standard interpretation). Instead, it is to be seen as an *indirect* method producing the exact solution, however, in discrete time (according to Eq. (3b)). In conclusion, (1) evasive maneuvers do solve problems, and (2) the computer got it all right all these years, but man's conceptual interpretation of its outputs has always been wrong (except for Euler, of course).

Shoham's Extended Prediction Problem Does Not Exist

In his book "Reasoning about Change" (1988), Shoham worries that the usual differential dynamics (cf. Eqs. (2) and (6)) only gives a prediction of an infinitesimally small time step forward from the considered current time point *t*. So how is it actually possible at all to make predictions over extended and finite periods of time on this basis? He calls this the extended prediction problem. My ontology mapping gives a direct solution to this: it turns the derivative into a strictly discrete and finite one-step forward difference into the future. Once you have solved this finite and discrete problem, you simply transform its solution back for any desired time *t* using the T transform (5). The extended prediction problem thus seems to satisfy the quoted Bishop Berkeley 1734 characterization concerning "ghosts of departed quantities".

Nonlinearity and the Curvature of Time: Bend It Like Beckham

The next important step is to show that the T transform method also handles *nonlinear* dynamics well. This gives it a major advantage over other transforms such as the Laplace and *z* ones. I will give a basic example of this, by considering a special case of the ODE (2), namely:

$$d/dt\, x_t = A\, x_t\, (1 - x_t) \qquad (10)$$

which is generally known as the logistic equation.

Logistic equation models. Varieties of it are widely used in practice, for example in population models of competing species in ecology. In one dimension, the linear term represents exponential growth, but the nonlinear (quadratic) term models self-limiting effects: lambs eat grass, but if there are too many in a territory (outside paradise), their population growth is ultimately restricted due to resource limitations. In more dimensions, the logistic equation can model interactions between species: lions eat lambs, but if they eat too many, first the number of available lambs will drop, and ultimately their own population numbers will go down. It is easy to imagine that such models often lead to (nonlinear) oscillatory cycles in population growth, with time delays between those of interacting species.

Solving the nonlinear logistic system (10) follows the same transform procedure as discussed above. Now, however, we have to use property X of Table 1 for its time transformation. This property might seem mathematically complex, but it is actually a discrete convolution that is computationally very simple to handle (it's just a sequence of basic additions and multiplications). Property X can be formally proven by (rather tedious) algebraic manipulation, properly rearranging terms at the right-hand side (a much more elegant derivation uses symbolic operator algebra, but this is beyond the space of this article). This results in an analytical solution of the nonlinear ODE (10) *in discrete time*:

$$X_{S+1} = (1+A)X_S - A\sum_{n=0}^{S} \binom{S}{n} \Delta^{S-n} X_0 \cdot X_n \,. \qquad (11)$$

The first term of this solution gives the linear part (as discussed above), and the second term yields the nonlinear effects. Again a variant of the Euler-type algorithm is suited to the task of prediction: from Eq. (11) it is easy to see that the discrete solution X_S obtains by successive single-step forward recursions starting from the known initial condition X_0 and then going forward in time: *S*=1, next *S*=2 etc. The probabilistic T map (5) then delivers the solution in continuous time for any desired time point *t*.

It is instructive to compare the solution (11) of the continuous ODE (10) with (i) the discrete solution (8) of the linear system (6), and with (ii) the nonlinear discrete dynamic system that is usually seen as its discrete analog (and therefore is known as the logistic map):

$$X_{S+1} = A\, X_S\,(1 - X_S) \qquad (12)$$

This logistic map is famous because it is more or less the simplest system that exhibits *chaotic* dynamic behaviour (in contrast to the logistic ODE). Again, there is a linear term and a quadratic nonlinearity, now in discrete time. But there is an essential structural difference between the discrete solution (11) to the logistic ODE on the one hand, and the linear system (8) and logistic map (12) on the other hand. The latter are iterated maps, i.e., result from repeated function application; to obtain the value at the next timepoint one only needs the preceding timepoint.

In contrast, Eq. (11) shows that in the solution of the logistic ODE all previous time points are involved. So, this continuous dynamic system has a memory in discrete time, even though this is not at all evident from the ODE formulation (10) that involves a single continuous timepoint. Although they share the name, the logistic ODE and the logistic map are totally different in their dynamic behaviour.

Lorenz chaos. Property X of Table 1 also enables to solve in discrete time the well-known Lorenz model (1963), developed to better understand atmospheric dynamics for long-range weather prediction. It became prominent because it was the first demonstration of the occurrence of chaotic behaviour in deterministic systems, with a so-called strange attractor (the famous "butterfly" shape to which the system tends in phase space). The Lorenz model is a simple 3D system with quadratic-type (in fact, bilinear) nonlinearities. So, property X directly applies, and the discrete solution of the Lorenz model has the same structure as Eq. (11).

In general, nonlinearity in continuous dynamics has the effect that it "bends like Beckham" the solution in discrete time: in contrast to the linear system Eq. (8), *all* values at time points before *S* play an explicit role in the full solution at time *S*, although the solution itself can always be computed by a one-step forward algorithm that also maintains the normal causal order of events, both in continuous and discrete time.

Preview of Coming Attractions

It is probably most interesting here to briefly investigate the *impact on the structure of time* resulting from nonlinear dynamics. Namely, the above methods and results suggest some intriguing conceptual (or if you wish, philosophical) insights into the nature of time, particularly regarding "density" and "curvature" of time.

Consider the nonlinear differential equation (2) in general and how it changes under the T temporal ontology mapping (5). The left-hand side T(lhs) is easy: according to property VIII it always maps onto a simple one-step forward difference. The right-hand side involves a composite function f(x(t)) that is generally nonlinear. Taking the transform T(rhs) changes our ontological view on the dynamic world in two stages:

- First, it changes the function f, seen as a function of x only, into a similar but not identical function F (witness for example the properties III-VI and X). This is already an important difference with the standard approach depicted in Figure 1.
- Second, it also changes the function x seen as a function of *t* (since the same properties apply again).

If we attempt to visualize this latter effect, we get a picture radically different from Figure 1. What happens is that the average *density* of the occurrence of (discrete) events is not constant but *changes* over the (continuous) time axis.

You might visualize this by imagining that the discrete time axis gets curved, and in continuous time you only see its projection onto the continuous time axis (see Figure 3). It is actually not difficult to find examples where the discrete-time "curvature" becomes so strong that it creates a singularity in (note: finite) continuous time.

Thus, the metaphor of the flow of time as a river [1] gets strangely bent due to nonlinearities: it's possible to create something like a black hole in the river bed of the timeline!

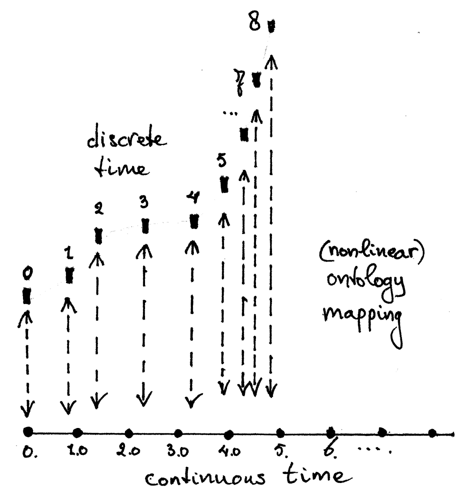


Figure 3. The T temporal ontology mapping may lead to flows of time that are "curved".

UPPER-LEVEL GENERALIZATIONS

This paper only outlines a small fraction of the results I have developed concerning nonlinear ontology mappings between time, and could only hint at the underlying mathematical proofs and algorithms. A few final general remarks are in order.

The uses of "old" science. First, the whole theory of temporal ontology mapping can be founded upon various treasures stemming from rather ancient mathematics. Much of it has more or less become extinct and superseded by modern computer (in fact, number crunching) approaches, and as a result is not treated anymore in modern textbooks on numerical methods. Specifically, this theory can be set up in a very elegant and concise way by means of symbolic operator algebra [3] as you find it in the textbook by Boole

[5] (the first edition was from 1860, by the way). What I actually find gratifying is that these methods are not just ancient, but if you develop them further as I tried to do in this paper, they can be actually made ready for today's intelligent system-style computing, and made valuable beyond popular number crunching styles of computing. A few examples have been given in this paper.

Computational complexity. An issue not discussed in this paper is the computational complexity of the algorithms related to the T transform. Generally, they are of low-degree polynomial complexity. The calculation of the T transform (5) itself is of order $O(N^2)$, where N is the number of timepoints considered (same order as matrix multiplication; this also applies to the inverse transform, as T turns out to be a so-called orthogonal transformation). The computational complexity of the solution of ODEs is commonly measured in terms of the number of function evaluations (the right-hand side of Eq. (2)). Then, the T transform method of solution is of *linear* complexity in time, cf. Eq. (11); I note that this is also true in the general case.

AI temporal reasoning and android epistemology. The present work has important differences with respect to much of the work in AI temporal reasoning (see e.g. [7]) in terms of focus and assumptions. The present work uses standard point algebra from mathematics, and therefore interval algebras and axiomatizations (such as Allen's, see also [1] and [7]) are not really relevant here. Important in my approach is that time is a *metric space*, i.e. a distance measure can be defined (in AI temporal reasoning usually called duration information). Another important difference is the type of tasks considered. AI temporal reasoning has spent much effort on constraint-based algorithms for establishing (partial) temporal ordering, possibly under incomplete or uncertain information. In contrast, this paper assumes full linear ordering in time (this is precisely what the variable S expresses), and focuses on tasks of prediction and control (in line with physics and mathematics). This paper shows that also in the point approach to temporal reasoning a lot of interesting progress can still be made.

If one refrains from delving into the mathematics, it yields some conceptual consequences for "android epistemology". Androids (computers, robots and other discrete machines such as presumably StarTrek's Mr. Data) live in a discrete spacetime. Humanoids seem to live in a continuous spacetime. So they inhabit ontologically speaking fundamentally different worlds. One might think that continuous beings can do all kinds of things in their spacetime that discrete beings cannot do in theirs – since continuous spacetime has many more points one can do something in or at than discrete spacetime. This paper shows this is not true: if they are sufficiently intelligent, discrete beings can do anything continuous beings can. Being "intelligent" can even be mathematically expressed here: the reasoning assumption that spacetime has characteristics of randomness, and still is causally ordered, according to Eq. (5).
Above I discussed things from the perspective of time. Surely this is a key top-level ontology concept. However, my approach and methods also work for types of independent variables other than time. Other continuous variables can be formally discretized in this way as well. For example, one can in this manner also treat the concept of space.

Ontology: content vs. representation. Finally, the paper has provided an in-depth case study in ontology mapping. I submit that this provides some evidence that ontology building, mapping, and reuse is much a substantive issue, more than a matter of generic representation language and semantic tooling. I note that this is already the case for such a high-level, generic, common, and commonsensical concept as time that does not depend on a specific domain. Substantive or content issues will be even more strongly present in task and domain specific ontologies. But in the end this is where the real semantic and web intelligence applications will be. This is perhaps a sign that the semantic research community at some point cannot avoid significant substantive issues in Web ontology, and has to be careful about (over)emphasis of generic representation and tooling issues without adequate domain grounding. Or, be sufficiently moderate in its expectations of the size of its own role in building the Semantic Web.

Acknowledgment. This work has been supported in part by a travel grant from the KnowledgeWeb Network of Excellence (EU-IST-2004-507482).

REFERENCES

[1] Van Benthem, J.F.A.K., *The Logic of Time*, 2nd ed., Kluwer, Dordrecht, NL (1991).

[2] Press, W. H., et al., *Numerical Recipes – The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, UK (1992).

[3] Dahlquist, G., and Björck, Å., *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ (1974).

[4] Leibniz, G.W., *Nova Methodus Pro Maximis et Minimis*, Acta Eruditorum, Vol. 3, pp. 467-473 (1684); English transl. Struik, D.J., Ed., *A Source Book in Mathematics*, pp. 272-280, Princeton, NJ (1986).

[5] Boole, G., *A Treatise on the Calculus of Finite Differences*, 2nd ed., Macmillan and Company, UK, (1872); reprint Dover, New York, NY (1960).

[6] Akkermans, J.M., and Běták, E., *Annals of Physics*, Vol. 194, pp. 148-172 (1989).

[7] Hayes, P., *A Catalog of Temporal Theories*, Technical Report (1995), http://www.ihmc.us/users/phayes/.

Towards Browsing Distant Metadata Using Semantic Signatures

Andrew Choi
Simon Fraser University
School of Interactive Arts and Technology
Surrey, BC, Canada
aschoi@sfu.ca

Marek Hatala
Simon Fraser University
School of Interactive Arts and Technology
Surrey, BC, Canada
mhatala@sfu.ca

ABSTRACT
In this document, we describe a light-weighted ontology mediation method that allows users to send semantic queries to distant data repositories to browse for learning object metadata. In a collaborative E-learning community, member data repositories might use different ontologies to control a set of vocabularies describing topics in learning resources. This could hinder the search of learning resources based on local ontological concepts. With the use of WordNet, we develop a toolkit that indexes ontological concepts with WordNet senses for semantic browsing in order to integrate information in a distributed learning community. The effectiveness of the toolkit was validated with real-world data in a specific domain, namely E-learning metadata.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – information integration, retrieval models, search process

General Terms
Algorithms, Management, Experimentation, Verification

Keywords
Semantic Retrieval, Data Integration, Ontology Mediation

INTRODUCTION
As the advance of the Internet and rapid development in E-learning, more and more institutions are joining to form a distributed learning network to allow users to access resources from different learning repositories. This creates pressure for institutions to provide an efficient way to organize a huge volume of materials located in different repositories, according to a consistent concept classification, in order to answer distributed retrieval requests. Currently, the use of metadata and ontologies to formalize semantics of concepts in the E-learning domain does not completely resolve the problem of interoperability in a federated environment. This is because metadata in different repositories are very often annotated with concepts defined by different ontologies specific to their organizations or communities. That makes finding information based on a local conceptual framework difficult. Different organizations with different backgrounds and target audience may use different terms with similar semantics to define and describe two similar learning resources. In addition to ontological differences, linguistic variations in metadata values and lack of use of metadata standard across learning network makes direct querying with keywords sometimes ineffective to discover a conceptually similar metadata.

PROBLEM DESCRIPTION
The primary objective of this research is to explore the use of semantic signatures expressed in WordNet senses to provide mediation between different ontologies in order to enhance concept retrieval. Consider the scenario when the learner L_1 associated with the repository R_1 looking for learning resources related to the topic of how to find a good bass musical instrument, L_1 sends out a request "*search for bass*" to remote repositories R_2 and R_3 respectively in an E-learning network. However, the returned results from them are mixed with many irrelevant resources related to catching a bass (e.g. fish). Such a problem occurs frequently when the concepts are defined by different domain ontologies with different sets of vocabularies carrying different intended meanings. Imagine another case when the same learner L_1 sends out a distributed request for learning resources on the topic "*advance databases*". Since the topic is annotated by the concept "*database systems II*" in remote repositories, that is to say it is labelled differently. Therefore, in a concept-based label matching search, learning resources defined by the concept "*database systems II*" will not be returned for the request of "*advance databases*" even though the two concepts are actually semantically equivalent.

From these simple scenarios, one can easily see that without a proper semantic mapping between ontologies in heterogeneous data sources, even with the ontology to

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

define vocabulary used to describe metadata on learning resources, it is still challenging to find learning resources based on the local conceptual definition.

OVERVIEW OF ONTOLOGY MAPPING

Semantic or ontology mapping can be described as a mapping task that identifies common concepts and establishes semantic relationships between heterogeneous data models in the same domain of discourse [1]. Since semantics is mostly defined by ontological constructs in modern knowledge systems, we will use the term semantic mapping interchangeably with ontology mapping in this discussion. According to [10], ontology mapping between two ontologies O_1 and O_2, can be expressed as a mathematical structure: $O_1 = (C_1, A_1)$ to $O_2 = (C_2, A_2)$ by a function f: $C_1 \rightarrow C_2$ to semantically related concept C_1 to concept C_2 such that $A_2 \models f(A_1)$ whose all interpretations that satisfy axioms in O_2 also satisfy axioms in O_1. For example, if the concept *agent* (C_1) is defined in O_1 by a set of properties such as <*broker*, *travel agent* and *officer*> with axioms such as <*part-of agency*, *is-a individual*, *is-a organization* and *type-of communicator*> (ignoring other attributes and cardinality for the sake of simplicity), it is possible to map it to a concept *representative* (C_2) defined in O_2 with a set of properties such as <*government agent*, *client*, *spokesperson* and *advisor*> and having axioms such as <*part-of government*, *is-a person*, and *is-a expert*>. This assumes that all the semantic interpretations of C_1 will be respected by C_2 in the domain of discourse when executing logical inference operation on C_2.

REVIEW OF OTHER APPROACHES

This section presents a brief overview of two approaches on semantic mapping. The two selected approaches are GLUE and MAFRA. The former is a system that employs machine-learning techniques to find ontology mappings with the use of probabilistic multiple learners while the latter uses a declarative representation of mappings as instances in a mapping ontology defining bridging axioms to encode transformation rules. With two domain ontologies, for each concept in an ontology GLUE claims to find the most similar concept in another ontology [7]. A number of features distinct GLUE from other similar mapping systems. First, unlike many mapping systems that only incorporate single similarity function to determine if two concepts are semantically related, GLUE utilizes multiple similarity functions to measure the closeness of two concepts based on the purpose of the mapping. The intuition behind the multiple similarity functions is to take advantage of the mapping requirement to relax or limit the choice of corresponding concepts. For instance, based on the requirement of the application the task of mapping the concept "*associate professor*" can be satisfied by similarity criteria "exact", "most-specific-parent" or "most-general-child" similarity criteria to find "*senior lecturer*", "*academic staff*" or "*John Cunningham*" respectively. This gives GLUE flexibility to find semantic mappings between ontologies. Second, GLUE applies a multi-strategy learning approach to use certain information discovered by different classifiers during the training process. This approach divides the classification process into two phases. First, a set of base classifiers is developed to classify instances of concepts on different attributes with different algorithms. Then, the prediction of these base classifiers, assigned with different weights representing their importance on overall accuracy, is combined to form a meta-learner. Finally, the classification is determined by the result from the meta-learner. As an instance, one base learner can exploits the frequency of words in the name property using a Naïve Bayes learning technique while another base learner can use pattern matching on another property using a Decision Tree Induction technique. At the end, the meta-learner will gather all the results to form the final prediction. Using multiple classifiers, GLUE intends to increase the accuracy of the overall prediction. Third, GLUE incorporates label relaxation techniques into the matching process to boost the matching opportunity based on features of the neighbouring nodes. Generally, the relaxation labelling iteratively makes use of neighbouring features, domain constraints and heuristic knowledge to assign the label of the target node.

MAFRA (Mapping FRAmework) is another ontology mapping methodology that prescribes "all phases of the ontology mapping process, including analysis, specification, representation, execution and evolution" [14]. It uses the declarative representation approach in ontology mapping by creating a Semantic Bridging Ontology (SBO) that contains all concept mappings and associated transformation rule information. In this model, given two ontologies (source and target), it requires domain experts to examine and analyze the class definitions, properties, relations and attributes to determine the corresponding mapping and transformation method. Then, all accumulated information will be encoded into concepts in SBO. Therefore, SBO serves as an upper ontology to govern the mapping and transformation between two ontologies. Each concept in SBO consists of five dimensions: they are *Entity*, *Cardinality*, *Structural*, *Constraint* and *Transformation*. During the process of ontology mapping, software agent will inspect the values from two given ontologies under these dimensions and execute the transformation process when constraints are satisfied.

Some recent approaches like INRIA[1] make use of OWL API to build a set of alignment APIs with built-in WordNet function for the purpose of ontology alignment or axioms generation and transformations. However, the details on the use of WordNet to generate the alignments are not well documented in the published literatures.

[1] http://co4.inrialpes.fr/align/index.html

WORDNET

WordNet is a widely recognized online lexical reference system, developed at Princeton University, whose design is inspired by "current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synsets (synonym sets), each representing one underlying lexical concept that is semantically identical to each other" [2]. Synsets are interlinked via relationships such as synonymy and antonymy, hypernymy and hyponymy (*Subclass-Of* and *Superclass-Of*), meronymy and holonymy (*Part-Of* and *Has-a*) [3]. Each synset has a unique identifier (ID) and a specific definition. A synset may consist of only a single element, or it may have many elements all describing the same concept. Each element in a particular synset's list is synonymous with all other elements in that synset. For example, the synset {World Wide Web, WWW, Web} represents the concept of computer network consisting of a collection of internet sites. In this context, 'World Wide Web', 'WWW' and 'Web' are all semantically equivalent. For cases where a single word has multiple meanings (polysemy), multiple separate and potentially unrelated synsets will contain the same word. For instance, the word 'Web' can have 7 multiple meanings defined in WordNet as computer network, entanglement, simply spider web and etc.

OUR APPROACH

To help distributed learning repositories to organize and manage their metadata in compliance with a global semantic view, we create a semantic mapping strategy using WordNet as a mediator to provide word sense disambiguation and to generate semantic signature each representing learning resource category.

Semantic signature in the categorical browsing context can be defined as a logical grouping of representational word senses for a class of metadata. In essence, it is a semantic representation of a class label with important WordNet senses regarding context. To formalize the concept of semantic signature, it can be written as follows:

$$Sig(c) = \bigcup_{j=1}^{n} DS_j = \bigcup_{i=1}^{n} BS_{di} \quad BS_{di} = Max\{Fav(d_j, s), \{t \in T \mid s \in WS(t)\}\}$$

where $Sig(c)$ = semantic signature for class c

DS_j = set of document senses for class c

BS_{di} = set of best sense in document d_j

T = all keywords in document d_j

Fav = selection function to find best sense

$WS(t)$ = set of WordNet sense for term t_i

To briefly explain, semantic signature of a class of metadata is built from a set of important document senses from all documents (metadata records) belonging to a particular class. In turn, document senses are generated

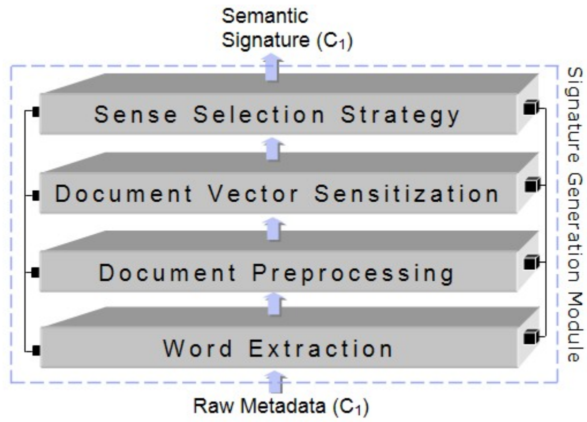


Figure 1. Semantic Signature Generation Framework

from a collection of the best WordNet senses for all representational keywords for a particular document.

The generation of a semantic signature for a class of metadata is divided into three distinct phases. In the rest of this section, the general architecture of the methodology is described while each phase is discussed in detail and as well as illustrated with examples.

System Design and Architecture

The methodology for creating semantic signature relies heavily on the assumptions that the aggregates of all semantic information from metadata records of a particular class are a good representation of the concept for that class. In fact, the metadata record is an instance of a concept in the ontological framework. Moreover, the methodology assumes that semantic information of a class can be approximated by a set of important word senses from all metadata records. Besides, semantic word senses specific to the context can be found based on important terms extracted from metadata through WordNet. Finally yet importantly, it assumes that the local semantic signature for a class of metadata is similar to signatures for metadata of semantically equivalent concepts in distant repositories. The methodology uses k-Nearest Neighbour (kNN) search algorithm to classify semantically relevant concepts in distant repositories based on local semantic signatures [11]. The instances (metadata) of concepts in local repository serve as the training dataset. Based on semantic features of the local metadata, semantic signatures for each class of concepts are formed. To find semantically relevant concepts in distant repositories, a distance function is defined and used to measure closeness between the query signature and semantic signatures for concepts in distant repositories. Eventually, k most similar concepts to the query signature will be retrieved from remote repositories.

Figure 1 shows the four phases of the semantic signature generation framework. In the Word Extraction phase representative features are extracted from each metadata document. The Document Preprocessing phase eliminates all irrelevant information as well as all non-noun words. In the Document Vector Sensitization phase all the representative keywords are used as seeds to find the corresponding word senses from WordNet. Finally, in the Sense Selection phase several strategies are applied to select the best word sense is selected among all senses to represent each word term.

Signature Generation in Action

Phase I: Word Extraction

First, the input metadata are transformed to comply with the IEEE LOM standard[2] using XML transformer. Then, adapted from Edmundsonian paradigm [4], content from <*Title*> and <*Description*> elements is extracted to represent the whole metadata document. That presumes that the content from these two elements carry important weight as cue phrase to be able to represent the whole document [4]. This view seems reasonable in the case of learning object metadata because other elements like *publication date*, *ISBN* or *format* do not bear good semantic information to signify the category of the metadata.

Phase II: Document Preprocessing

The condensed metadata with only the <Title> and <Description> elements are subjected to cleaning to remove all stopwords, punctuation information, numerical values and irregular symbols. Next, all non-noun words are removed using part-of-speech tagger except some commonly used phrasal words which carry specific meaning. For example, the word "artificial" in the phrase "artificial intelligence" will be preserved to retain the special meaning of the binary phrase in the branch of computer science. The reason why this approach only uses nouns as the base keyword is explanined in [5] where it is said that long phrases are not easily disambiguated comparing to a single word term or a binary word term. The accuracy to use a phrase as a distinguishing feature for a document classification in effect will be lower through previous experiments demonstrated in [6]. On the other hand, it has been shown that the use of noun word terms carry the most salient expression to serve as distinguishing feature for doing text classification [7].

Phase III: Document Vector Sensitization

Supposing that all irrelevant information has been eliminated, the physical metadata documents are projected onto the vector space model. The document vector becomes a logical representation of the physical metadata record. Then, using TFIDF weighting scheme we select most significant terms across all document vectors to represent a category of metadata [12]. After that, each word term with the TFIDF score higher than the threshold is sent to WordNet to retrieve the corresponding word senses and its definition. The threshold is determined by trial and error approach with a test run. A single word term can have multiple word senses retrieved. For example, the word "search" can be mapped to WordNet senses as <hunting, hunt>, <lookup> and <investigation>. Because of this, the mapping information of a single noun word term can be denoted by a triple construct in the form <T, S, D> where T is the original word term, S is the synset of T and D is the definition of T. When a noun term can be mapped to multiple senses, there will be multiple triples. Take the word term "search" as an example. After the sensitization, it becomes <search – {hunting. hunt} = "the activity of looking thoroughly in order to find something or someone" (TFIDF 0.623101)> in triple construct. The triple construct format is used to substitute the original word term in the master document vector. Then again, recall that since a single word term could be mapped to possible different word senses through WordNet. Each word sense is represented in synset which may have multiple synonymous terms. Because of this, the length of the document vector in word sense will grow considerably. This problem is addressed in the next phase.

Phase IV: Sense Selection Strategy (S^3)

This is the last, and the most crucial phase in the method. It chooses the best word sense among all retrieved word senses from WordNet to represent the word term. As stated, a word term can be mapped to multiple WordNet senses. In such a case, the dimensionality of the vector grows significantly after the sensitization procedure. Imagine that a word term "light" can be mapped to 15 WordNet noun senses "visible light", "light source", "luminosity", "lighting", etc. The growth ratio is 15 times in this case. Such a high dimension not only negatively affects the efficiency of the similarity computation, but more seriously, the many senses are noise which does not carry actual meaning of the word in the *context* of the document. Included irrelevant senses will distort the semantic representation of the signature and lower the accuracy in similarity calculation when finding similar classes of metadata using signature matching. On the other hand, from the semantic knowledge standpoint, WordNet senses only provide the lexical information of the word term, but not the contextual information to determine how the meaning is clarified in a specified context [8]. Without that, the semantic signature is just a bigger collection of keywords and would have small use in identifying the classes of metadata based on the semantic relevance of the signature. Therefore, it is necessary to find a way to reduce the dimension and only select the sense that conveys the main idea of the word in the current context. To select the best sense representing a word term, a contextual-based Senses Selection Strategy (S^3) is applied to retrieved word senses. The strategy is based on the assumption that the local contextual information of a document serves as a good hint to tell which sense represents the actual meaning of the word term best. The S^3 approach can be summarized in the following algorithm:

[2] http://ieeeltsc.org/wg12LOM/lomDescription

Steps of algorithm (Calculate the best senses for class C1):

For each metadata document $D \in C_1$

Get the list of synsets for each word term $T_1 \in D$

For each synset Syn_1 of the word term T_1

For each sense term $S_i \in Syn_1$

1. Compute associative frequency *af* for S_i to other senses $S_k \in Syn_k$, $Syn_k \subseteq T_k$ and $T_1 \neq T_k$

   1.1 Find the sense Sl with highest score Max(*af*)

   1.2 If (Max(*af*) < 1) then go to 2 otherwise stop and return S_l

2. Compute associative frequency *af* for S_i to k-order parent senses $PS_k \in P(Syn_k)$, $P(Syn_k) \subseteq T_k$ and $T_1 \neq T_k$

   2.1 Find the sense S_p with highest score Max(*af*)

   2.2 If (Max(*af*) < 1) then go to 3 otherwise stop and return S_p

3. Return the most popular sense S_w offered by WordNet

Return the Best Sense to represent word term T_1

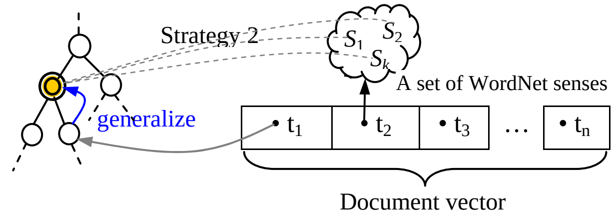
Aggregate all sense from all important word terms to represent signature of the document *D*

The algorithm works in the following way. For each word sense of a word term, it first computes the associative frequency (*af*) of each sense term in a synset to other sense terms in other synset of other word terms in the same document. From this, the most occurred word sense will be used to substitute the semantic representation of the word term.

Next, if the word sense of a word term cannot be discriminated by Strategy 1, the algorithm generalizes the word term to the k-order parent senses. In this approach, the value of k is 1. Hence, it generalizes to its immediate parent word sense. Referring to Figure 2, Strategy 2 will use the immediate parent sense to compute the associative frequency against other senses from other word terms in the document vector. As such, in this example the word term t_1 will be rolled up to its immediate parent through hypernym (is-a) relation in the WordNet hierarchy. Then, the parent's synset is used to calculate the associative frequency to other word senses for other word terms. Unlike other generalization approaches [7, 13], we generalize the sense to its most-specific parent only. The reason why it uses immediate parent senses (k=1) to compute the associative frequency is given in [9] where the most specific parent in a hierarchical terminology has a higher distinctive power to classify the topic. Essentially following the intuition that if a word sense is generalized to higher order parent sense than k=1, the generalized sense may be too general and becomes incoherent to local context, and would become noise when used to classify metadata.

Finally, as arranged by WordNet, the word senses retrieved from WordNet for a particular word are a partial order set ranked by popularity in English usage. If the previous two strategies can not find the best sense to represent the word term, then the most popular sense offered by WordNet will be adopted in Strategy 3.

Figure 2. Compute associative frequency between immediate parent with other word sense



The rationale behind sequencing three strategies is based on observations and hypothesis that the local context is the most specific and relevant candidate to provide contextual meaning for the word term sense. Therefore, a word sense for a particular term can most likely be disambiguated by other local senses (Strategy 1). If it could not be resolved by step 1, then it compares the immediate parent sense to the other word senses to check if the parent sense is a frequently occurring sense for the underlying word term. At last, the most popular sense is adopted to represent the semantic meaning for a word term when the two strategies above could not resolve the ambiguity of the word term.

Following the above procedures, a set of senses becomes a semantic signature of a document. In order to generate the final semantic signature for a class of documents referring to particular concept, TFIDF scheme is applied again to each word sense in all document signatures for a particular class. Based on the score, the most relevant senses for characterizing the class of metadata are aggregated to form the final signature for the class.

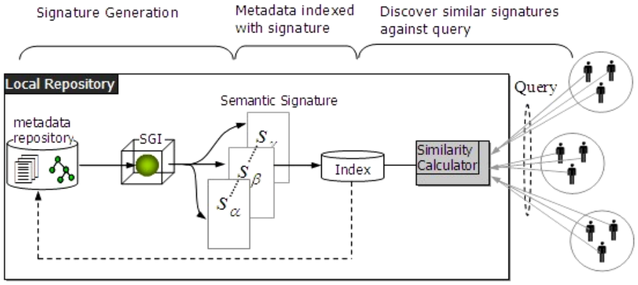
Concept browsing in heterogeneous ontologies

In our application, the generated semantic signatures are used to index the actual classes of metadata for fast distributed browsing. We developed a tool called Signature Generation Indexer (SGI) that supports the methodology described in the previous section. Focusing on the efficiency, the design of SGI is to allow repository operators to produce semantic signatures for classes of learning object metadata easily without tedious human interaction, or complicated implementation.

The ultimate goal is to achieve semantic search based on E-learning topics defined by heterogeneous ontologies in a federated network. In a collaborative learning environment, users expect to be able to access all the learning resources within the learning network. To fulfill this anticipation, it is important to assume that all participant repositories in the collaborative network employ the same strategy to index learning resources metadata with WordNet semantic signature.

In this way, when users launches a query by selecting a specific topic (concept) from the local ontology (e.g. via user interface), the corresponding semantic signature representing the topic is retrieved from local database. The signature is then sent across the network to participating

Figure 3. Integrated process of semantic-based browsing of metadata



learning repositories. The query in the form of semantic signature is the input of the Similarity Calculator in distant repositories. The Similarity Calculator is used to compute the similarity of signatures in each of the learning repositories. The similarity calculator uses the cosine similarity function, thereby the more matched elements in the signature, the higher the score is. In calculating the similarity score, different weights are assigned to senses from <Title> and <Description> in which the match in the title sense gets higher contribution to overall score than the one from the description tag.

In order to ensure the global accuracy of the result, results from participating remote repositories are merged and sorted in the descending order based on the cosine similarity score. Then, the top k (k=5) topics of the metadata are offered as the answer to the local query. The overall operation of the semantic-based browsing of learning resources metadata is shown in Figure 3.

IMPLEMENTATION

The SGI is implemented in the C# programming language. The current version is a desktop application, but it can be easily extended to a web service. The goal of SGI is to integrate signature generation, document indexing and browsing capability. The signature indexes are stored in an inverted index database (e.g. MS Access). The similarity calculator is a separated module implemented in C# as well and connected to the index database. Figure 4 shows the browsing interface of SGI to illustrate how to search distant concept semantically.

Figure 4. Browsing interface of SGI

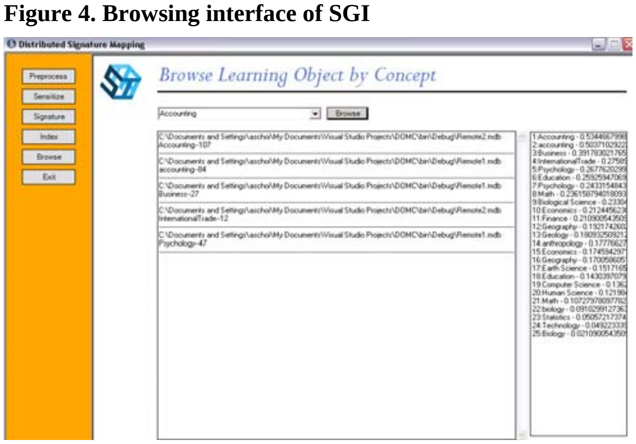
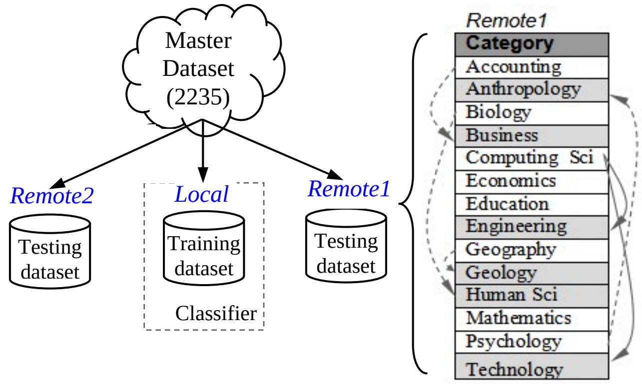


Figure 5. Dataset distributions into training and testing data



EVALUATION

In order to test the hypothesis of using semantic signatures to enable distributed semantic browsing and to improve relevance we have simulated the distributed concept retrieval and compared the results with the traditional keyword-based and label-matching method. To replicate the distributed repositories in a collaborative E-learning network, the three independent databases are set up. As shown in Figure 5, they are called "local", "remote1" and "remote2" where the local, of course, denotes a local data source and both remote1 and remote2 simulate distant data sources. A single master set of metadata in 8 different categories is distributed evenly in number and randomly into the three simulated repositories.

The metadata have been transformed to conform to the IEEE LOM format. After the distribution, the local database contains the metadata that represents the set of the training data for the classifier. During the training phase, the kNN classifier uses the instance of the local metadata to learn the features to identify the class of the metadata. It starts by extracting keyword terms from each category of metadata and projecting them into the vector space model. Next, after running through the signature generation module, each category of metadata is represented and indexed by a semantic signature in the database.

The dataset in both remote1 and remote2 is controlled to model the situation of potentially different ontological classification in a distributed environment. To simulate the effect of varied concept labelling, the original 8 categories of metadata are expanded to 14 categories in remote1. The 6 derived categories are labelled with different class names from their respective sources and described with the metadata taken out from source categories. Each newly derived category contains metadata belonging to the same class. To illustrate, a part of the metadata from the category "computing science" is distributed to the derived categories "technology" and "engineering" in remote1. Thereby, the metadata for concept "computing science" is now grouped into “computing science”, “technology” and “engineering”. Essentially, this simulates the situation when a concept “computing science” could be categorized differently into concepts like “technology” and “engineering” in different ontology. The same distribution principle is applied to remote2 database which includes 13 categories of which 7 are derived categories.

Similar to the local database, each category of the metadata in remote1 and remote2 is mapped to a semantic signature in WordNet senses and stored in the local database as an index. To test semantic-based search, semantic signature representing a local concept is sent to query the remote repositories. The semantic similarity is compared between the query signature and the distant signature based on the similarity function. Finally, the result of the k most similar concept signatures from the remote databases are studied based on the relevance metric.

Dataset

Since there is no publicly available dataset of learning resources metadata, the experiment metadata were acquired through a number of different sources. Table 1 shows the category of metadata acquired and their respective sources. In total, 2235 metadata subdivided into the 8 different categories are acquired. The dataset is partitioned into training and testing groups. As mentioned, the local database stores the training dataset while remote1 and remote2 store the testing dataset. All metadata are known with their class label. Metadata are distributed randomly, using Microsoft Excel random generator, to train and test the group. After distribution, the local database contains 667 training records while remote1 and remote2 contain 1568 testing records.

Results

In order to gauge the effectiveness of the proposed mediation method between different E-learning ontologies, three standard metrics for information retrieval are used in the evaluation of the system performance: they are Recall, Precision and F-measure. Table 2 shows that the use of semantic signature can consistently improve retrieval relevance in terms of recall and precision. In all categories, the semantic based retrieval out perform both keywords-based retrieval and label-matching retrieval.

Table 2. Comparison on precision, recall and F-measure on concept retrieval

Category	Precision			Recall			F-Measure		
	S	*K*	*L*	*S*	*K*	*L*	*S*	*K*	*L*
Acc	1	0.6	0.5	1	0.75	0.5	1	0.6	0.5
Bio	0.6	0.6	0.5	0.75	0.6	0.5	0.6	0.6	0.5
CS	1	0.5	0.3	1	0.5	0.3	1	0.5	0.3
Econ	1	1	0.6	1	0.75	0.6	1	0.6	0.6
Educ	0.6	0.5	0.5	0.75	0.75	0.5	0.6	0.45	0.5
Geo	0.6	0.5	0.5	0.75	0.5	0.5	0.6	0.5	0.5
Math	1	0.3	0.6	0.6	0.5	0.6	0.7	0.36	0.6
Psy	1	0.3	0.3	0.6	0.6	0.3	0.7	0.4	0.3

S = Signature-based retrieval, *K* = Keywords-based, *L* = Label-matching

Table 1. Source and Category of Metadata

Category	Source	No. of records
Accounting	Business Source Premier Publications	382
Biology	Biological and Agricultural Index, BioMed Central Online Journals	315
Computing Science	Citeseer	320
Economics	American Economic Association’s electronic database	353
Education	Educational Resource Information Center	307
Geography	Geobase	237
Mathematics	arXiv.org, MathSciNet	157
Psychology	PsycINFO, ERIC	164

As oppose to the classic or traditional keywords-based representation, semantic-based indexing with WordNet senses can include more lexicon information than simple syntactic approach. This implies that more features will be added to the class signature representation. Since more features are added, that may also mean that more noise is included as well.

Intuitively, the increased relevance of retrieval can be attributed to the expansion of features in class representation. However, different from what we expected, the precision does not decreased. It is suspected that due to the relatively small size of the dataset and 1-k hypernym generalization, the senses included in the signature are ‘good’ in terms of classification. Therefore, combined with a good contextual-based sense selection strategy, WordNet as a mediatory can provide source for ambiguity resolution and semantic information for the process of semantic browsing. Coupled with that, the selection of kNN algorithm as the classifier also contributes to the performance of the system.

kNN is an instance-based classifier. The performance of instance-based classifiers is more dependent on the sufficiency of the training set rather than other machine learning classification algorithms. Thus, it is a disadvantage for kNN to have a small dataset for training and testing. A smaller training set implies more terms or term combinations important for content identification may be missing from the training sample documents. This negatively affects the performance of a classifier. Nevertheless, the ontology (e.g. WordNet) guided approach seems to somewhat reduce the negative influence of this problem. The replacement of child concepts with parent concept through hypernym relationship appears to be able to discover an optimum concept set without adversely affecting performance. Therefore, an important term, which resides low in the concept hierarchy may be mapped to a parent concept and included in the signature for class comparison, even if this term is not included in the training set.

DISCUSSION

The improvement on concept retrieval by using semantic signature is not uniform across different categories. For example, the improvement on retrieval of "*Psychology*" and "*Accounting*" metadata is more than improvement on "*Biology*" and "*Geology*". We believe that for some classes of metadata like "*Biology*", which are characterised by a set of specific keywords, the use of semantic signatures does not add extra useful information into the representation model to help in classifying metadata. On the other hand, using 1-k hypernym generalization on such a highly specialized domain may in fact introduce more noise to reduce the matching possibility in similarity calculations. In addition, with a small size of dataset, over-fitting on classification model may also result. Therefore, further experimentation and analysis are needed to fully understand the impact of WordNet signature with sense generalization in classification of metadata.

CONCLUSION

This project offers two important contributions. First, it gives a new light-weighted semantic (ontology) mapping approach to enable cross platform concept browsing in a federated network. Unlike many current practices in semantic mapping that either require intensive user involvement to provide mapping information, or resort to complicated heuristic or rule-based machine learning approach, this work shows an effective automatic mapping protocol that can allow federated concept browsing with semantic signature. It is evident for the experimental results that establish the merit of using WordNet to provide semantic knowledge for metadata classification in the domain of E-learning. The merits include the provision of semantic representation of categorical data and increased semantic relevance in categorical browsing.

By using immediate parent sense generalization during sense selection process, it does not only successfully reduce the dimension in semantic signature, but more importantlly introduces flexibility in the sense selection and increases the opportunity to find a better sense without compromising the relevance in the search result. This creates incentive to explore the use of other sense selection strategy.

REFERENCES

[1] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, Pedro Domingos, "*iMap: Discovering Complex Semantic Matches between Database Schemas*", Proceedings of the ACM SIGMOD Conference on Management of Data. (2004)

[2] George A. Miller, *Wordnet: An Online Lexical Database*, International Journal of Lexicography (1993).

[3] Asuncion Gomez-Perez, *Ontological Engineering with Examples from the areas of Knowledge Mangement, e-Commerce and the Semantic Web*, Springer-Verlag London (2004).

[4] H. P. Edmundson, *New Methods in Automatic Extracting*, Journal of the ACM (1969).

[5] Ching Kang Cheng, Xiaoshan Pan and Franz Kurfess, "*Ontology-based Semantic Classification of Unstructured Documents*".

[6] Khaled M. Hammouda and Mohamed S. Kamel, "*Phrase-based Document Similarity Based on an Index Graph Model*", Proceedings of IEEE International Conference on Data Mining (2002).

[7] AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy "*Learning to map between ontologies on the semantic web*", Proceedings of WWW2002 conference (2002).

[8] Ching Kang Cheng, Xiaoshan Pan and Franz Kurfess, "*Ontology-based Semantic Classification of Unstructured Documents*", Proceedings of 1st International Workshop on Adaptive Multimedia Retrieval (2003).

[9] Martin Ester, Hans-Peter Kriegel and Matthias Schubert, "*Web Site Mining : A new way to spot Competitors, Customers and Suppliers in the World Wide Web*", Proceedings of 4th International Conference on Knowledge Discovery and Data Mining (2002).

[10] Yannis Kalfoglou and Marco Schorlemmer, *Ontology mapping: the state of the art*, The Knowledge Engineering Review (2003).

[11] Mineau, G.W, "*A simple KNN algorithm for text categorization*", Proceedings of IEEE International Conference on Data Mining (2001).

[12] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill (1983).

[13] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, *Semantic matching*, In 1st European semantic web symposium (ESWS'04) (2004).

[14] A. Maedche, B. Motik, N. Silva and R. Volz, "MAFRA - A MApping FRAmework for Distributed Ontologies", in EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, pp. 235-250, 2002.

Semantic Association of Taxonomy-based Standards Using Ontology

Hung-Ju Chu, Randy Y. C. Chow, Su-Shing Chen
Computer and Information Science and Engineering, University of Florida
Gainesville, FL, U.S.A.
{hchu, chow, suchen}@cise.ufl.edu
Raja R.A. Issa, Ivan Mutis
Rinker School of Building Construction, University of Florida.
Gainesville, FL, U.S.A.
{raymond-issa, imutis}@ufl.edu

ABSTRACT

The vision of *semantic interoperability*, the fluid sharing of digitalized knowledge, has led much research on *ontology/schema mapping/aligning*. Although this line of research is fundamental and has brought valuable contributions to this endeavor, it does not represent a solution to the challenge, *semantic heterogeneity*, since the performance of proposed approaches significantly relies on the degree of uniformity, formalization and sufficiency of data representations but most of today's independently developed information systems seldom have common knowledge modeling frameworks and their data are often not formally and adequately specified. Consequently, a workable solution usually requires interventions of domain experts.

In human society, *hierarchically structured standards* (or *taxonomies*) for characterizing complex application processes and objects used in the processes are often used as a common and effective way to achieve some *semantic agreements* among stakeholders within a domain. This research hypothesizes that the establishment and the use of such standards can serve as a framework that can effectively facilitate the reconciliation of *semantic heterogeneity* in complex application domains. However, the reality shows that a comprehensive priori consensus is extremely difficult, if not impossible, to reach. Consequently, various complementary and competing standards are often created and their constant-changing nature yields another level of challenge in achieving the hypothesis.

This paper focuses on the development of methodology for bridging *complementary* standards within an application domain. It exemplifies such standards in building construction industry where interoperability problems are prevalent and human interactions are commonplace. It proposes a semi-automatic approach for *semantically associating* the standards to reduce costly human intervention in a workflow. The approach formalizes standards by using ontology and discovers their affinity (to what degree they are related with respect to their usage) from automated project document processing and semi-automatic domain expert inputs. A high-level architecture of an integration framework in web environment is suggested for depicting the role of the semantic association approach in the system.

K-CAP'05, October 2-5, 2005, Banff, Canada.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing - *Indexing methods, Linguistic process*; I.2.4 [Artificial Intelligence]: I.2.1 Applications and Expert Systems - *Industrial automation*; I.2.4 Knowledge Representation Formalisms and Methods; I.2.6 [Artificial Intelligence]: Learning - Knowledge Acquisition

Keywords
taxonomy and standards, semantic interoperability, ontology-based knowledge extraction, semantic mapping.

1. INTRODUCTION

The vision of *semantic interoperability*, the fluid sharing of digitalized knowledge, has led much research on *ontology* (formal specification of conceptualization) and its languages, such as Web Ontology Language (OWL) [8]. The language provides primitives for specifying concepts, properties, explicit semantic relationships, and logical constraints on those objects. However, it does not address the issue of *semantic heterogeneity* between two independently developed ontologies. For example, a program that reads an ontology in OWL does not understand another ontology in the same language unless there is an explicit mapping between them. This difficulty has led much research on *ontology/schema mapping/alignment* [4], [5], [6], [11], [12], [13], and [14] and various *matching* technologies have been developed based on the attributes of objects and their associated data. Although this line of research is fundamental and has brought valuable contributions to this endeavor, it does not represent a solution to the challenge as we see. The performance of proposed approaches significantly relies on the degree of uniformity, formalization and sufficiency of data representations. Unfortunately, the concept of unified, formal, and sufficient specification is often an after-thought and most of today's independently developed information systems seldom have common knowledge modeling frameworks and their data are often not formally and adequately specified. Consequently a workable solution usually requires interventions of domain experts.

In human society, *hierarchically structured standards* (or *taxonomies*) for characterizing complex application processes and objects used in the processes are often used as a common and effective way to achieve some *semantic agreements* among stakeholders within a domain. This research hypothesizes that the establishment and the use of such standards can serve as a framework that can effectively facilitate the reconciliation of *semantic heterogeneity* in complex application domains. However, the reality shows that a comprehensive priori consensus is extremely difficult, if not impossible, to reach. Consequently, various complementary and competing standards are often created and their constant-changing nature yields another level of challenge in achieving the hypothesis.

This paper focuses on the development of methodology for bridging complementary standards within an application domain. We have chosen a target application in the building construction domain, where interoperability problems are prevalent and human interactions are commonplace. In that domain, a variety of taxonomy-based standards have been established but still lack a uniform and systematic way for supporting efficient collaboration among project participants using different standards. This problem is further compounded by the complexity and the dynamics of business applications, which often require changes of the well-known standards. The interoperability cost in such environment is tremendous. For example, based on a recent National Institute of Standards and Technology (NIST) report [3], a conservative figure of $15.8 billion was determined to be annual costs due to a lack of interoperability in the capital facilities industry in 2002.

Two mainstream complementary standards, MasterFormat and UniformatII, in that domain are considered in our research. MasterFormat [1] is a specification standard established by the Construction Specification Institute (CSI) for most nonresidential building construction projects in North America. UniformatII is a newer American Society of Testing and Materials (ASTM) standard aiming at providing a consistent reference for the description, economic analysis, and management of buildings during all phases of their life cycles [2]. These standards were created by different stakeholders with different perspectives for different purposes. For instance, an architect is interested in the design and structure of a building, a contractor wants to know what materials are used and how much they cost, and a building inspector is concerned about building code compliance issues. MasterFormat classifies items primarily based on the specification of products and materials used in construction, so it is based on a conceptual view of a contractor. Complementarily, the taxonomical classification in Uniformat II is primarily based on the attributes and location of structural building components, such as foundations and exterior walls, which reflects the architect's view of a construction project. Although their views are different but both address the same building object. In other words, the taxonomies of the standards classify the same set of objects but on different attributes. From here one can easily infer that cross-referencing or document conversion between the standards is inevitable for interaction among project participants in applications such as cost estimation and code compliance checking. For example, a wall (interial or exterial) in UniformatII needs to be associated with the material (metal, wood or fiberglass) in MaterFormat and conformed to its intended usage (hurricane or fire proof) according to building code regulations (standards yet to be formalized by the industry). In general, UniformatII by design is more suitable as a participant communication/interaction framework than MasterFormat during the earlier phases of the life cycle. On the other hand, Masterformat has been used for years and has gained the majority of the construction industrial support for specifying detailed project documents. To facilitate more efficient collaboration among project participants, it is a common practice to supplement UniformatII with Preliminary Project Descriptions (PPDs) or schematic design in earlier phases, and convert them to construction documents in Masterformat during later phases. In addition, the conversion is also necessary for cost calculation since most databases of building materials suppliers are based on MasterFormat. It is desirable to transform pre-bid elemental estimates to MasterFormat, and from there to the trade costs of the project [2]. This process is often tedious and requires cross-area knowledge. Currently, it is done manually by domain experts and it is considered a major cause that hampers interoperability in the construction domain. Bridging the two standards is a key enabler for enhancing the interoperability.

Directly matching approaches based on attributes of the entities of the standards are expected to be inefficient due to the heterogeneous nature of complementary standards. This paper proposes a practical compromise by redefining the notion of mapping with a semi-automatic semantic extraction framework to assist domain experts in achieving interoperability. The mapping is termed as semantic association for relating elements between standards, and is dependent on the intended use such as cross-referencing of elements or specification semantic mapping. The semantic relationship can be characterized in two measurements: similarity (how closely objects resemble each other in their representation) and affinity (to what degree they are coupled in their usage). In some sense similarity is more static while affinity is more dynamic and general. For example, a bicycle is similar to a car due to their physical structures and properties. However, gasoline is more affinitive to a car although they do not resemble each other. Exploiting affinity in addition to similarity through semantic association is the focus of this research.

The approach consists of three components: formalization of taxonomies, ontology-based semantic extraction and measurement of affinity. The first component is a simple and yet novel approach for annotating a standard in primitive descriptive statements constructed by a set of necessary and sufficient orthogonal relations. They are then normalized and generalized into ontology. The second component shows how the ontology can be used for the extraction of relevant information from the instances in other standards for semantic association. The third component quantifies the affinity for ranking the extracted metadata to identify optimal association. The following sections detail the three components and outline an overall architecture of an integration framework depicting the relationship between the proposed approach and other related technologies and systems.

2. FORMALIZATION OF TAXONOMY

Taxonomies are initially designed for human consumption therefore some domain knowledge that is obvious and assumed by stakeholders is often omitted in their specifications. Moreover, taxonomies classifying large and complex items usually have the following characteristics:

1. The entities being classified and the attributes upon which the classification is based, are themselves complex concepts.
2. Multiple attributes (different concepts) might be used to classify entities at the same level.
3. Attributes are not orthogonal and might result in overlapping concepts in low-level entities (an object can fit into multiple categories).

There is a need for a systematic approach for annotating assumed semantics, clarifying complex concepts, and transforming them into formal representation before taxonomies can be effectively used for semantic association.

Semantic depends on context and context depends on applications. In other words, the semantic of a standard is open depending on how they are used. To avoid a standard being bound to specific applications, the intrinsic semantic of a standard without context should include the following:

1. the attributes being used for classification under the general perception in the application domain and
2. the entities under the inheritance of the taxonomy and the attributes.

To model the intrinsic semantics, ontology is considered in this research. The following subsection describes a systematic approach for transforming taxonomy into ontology.

Ontology Development from Taxonomy

The term, ontology, has been widely used in several disciplines, such as philosophy, epistemology, and computer science. There is much confusion in its definition. For example, in philosophy it refers to the subject of existence while in epistemology it is about knowledge and knowing. In computer science, many people use Gruber's definition [10] – an explicit specification of a conceptualization. In the context of our research, we interpret it as a description of the concepts/terms and relationships that can exist in an application domain. Centered on terms and relations, the transformation of taxonomy into ontology is described in the following steps.

Step 1: relation set identification

The goal of this step is to identify a sufficient and necessary set of orthogonal relations for a given taxonomy/standard so that assumed domain knowledge and complex concepts can be formally specified. This step should be manually done by standard committees who know best about the original intended use of the standards. The set should be constructed from two types of relations: primitive and derived.

Primitive relations are those that are unambiguously understood by the general public and the relationship between concepts connected by them does not change over time. Moreover, they reflect the intrinsic properties of objects or describe time and space and the intention of users when the objects are used. In addition, their definitions should include set relationship, such as instance-instance, instance-class, and class-class, to avoid ambiguity. For example, *part_of* is ambiguous since it could mean a subcomponent of an object or the membership of an object in a class. Its meaning can be identified as the first explanation if instance-instance is specified.

Derived relations are those that can be composed/modeled from primitive relations.

To elaborate this step, a small portion of the top three levels in MasterFormat taxonomy, Division 5 (D5) Metals and Division 6 (D6) Wood and Plastic rooted from Material, is exemplified as follows:

Division 5- Metals

05100 Structural Metal Framing

05120 Structural steel

05140 Structural aluminum

05160 Metal framing systems

05400 Cold formed metal framing

05410 Load bearing metal studs

05420 Cold formed metal joists

05430 Slotted channel framing

Division 6 - Wood and Plastics

06100 Rough carpentry

06110 Wood framing

06400 Architectural woodwork

06460 Wood frames

The following relations are identified for formalizing the above example:

1. *used_for* (class-class, human intention): purpose
2. *kind_of* (class-class, intrinsic): containment relation of attributes of instances.
3. *instance_of* (instance-class, intrinsic): membership
4. *made_of* (class-class, intrinsic): material component

Table 1 shows the mathematical properties of these relations that are used in the subsequent step for data normalization. They are also used for reasoning in knowledge extraction.

Table 1. Mathematical Properties of the relations

Relations	Transitive	reflexive	antisymmetric
used_for	-	-	-
kind_of	+	+	+
instance_of	+	+	+
made_of	+	-	-

Step 2: relation statements construction

This step is to construct simple statements using the relations defined in step one and all keywords in the taxonomy. The statements are then processed in subsequent steps for constructing ontology. There are two advantages using this bottom-up approach for formalizing taxonomies. One is that it can better address the dynamic nature of standards by enabling incremental updates and modifications of the statements and their resulting ontology. The other advantage is that domain experts who are not familiar with ontology can directly express their knowledge in the simple statements without communication overhead with knowledge modeling experts.

The following are examples of relation statements that partially describe the example shown in previous step.

1. Metals (D5), Wood (D6), Plastics (D6_1) are *instance_of* Material (root) → (D5_root, D6_root, D6_1_root)
2. Metals (D5) are *used_for* framing → 05100_1
3. Structural is a *kind_of* "metal framing" (05100_1) → 05100
4. Cold formed is a *kind_of* "metal framing" (05100_1) → 05400
5. Studes are *made_of* Metals (D5) → (05410_1)
6. "Load bearing metal studs" are *kind_of* Metal studs (05410_1) → 05410
7. 05410 is *used_for* 05400 → (05400_05410)

Note that each statement is given a unique identifier (following →) derived from the original identifier of a taxonomy entity.

Step 3: normalization

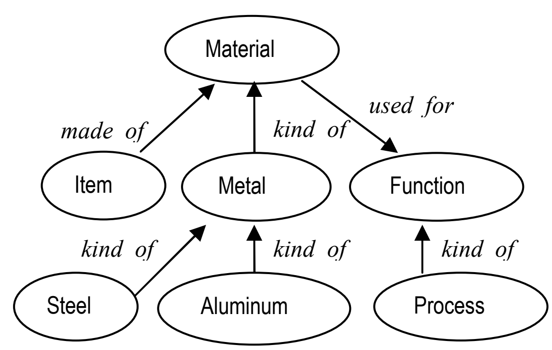
It is likely that redundant or conflict statements are generated along the way when domain experts annotate their taxonomies in the above steps. Based on the mathematical properties of the relations, this step normalizes the statements by:

1. redundancy elimination (removing same or equivalent statements)
2. conflict detection (for example: A-r1-B, and B-r1-A statements are conflict if r1 has asymmetric property)
3. implication detection (for example, A-r1-B, and B-r1 C statements imply A-r1-C through transitive property).

Step 4: semi-automatic generalization

This step is to generalize the resulting statements from step 3 into higher-level concepts connected by the same set of relations. Human being intervention is required in this step due to the complexity of the process. For example, if there exist A-r1-C, A-r1-D, B-r1-C, and B-r1-D, they can be generalized to concept1{A,B}-r1-concept2{C,D} by union. However, it becomes difficult when the above example is extended to include concept1{A,B}-r1-E and concept2{C,D}-r2-F. One cannot conclude concept1{A,B}-r1-concept2{C,D,E} unless an exception indicating no E-r2-F is added. Alternatively, it can be generalized to concept1{A,B}-r1-concept3{E,concept2{C,D}}. The system interacts with users by prompting the dilemmas for resolutions along the process of a whole taxonomy.

Figure 1 shown below depicts the generalized view or ontology of the relation statements shown in previous steps.



{metals, wood, plastics ..} are *instance_of* Material

{stud, joist ..}are *instance_of* Item

{framing, ..}are *instance_of* Function

{cold formed, structural ..} are *instance_of* Process

Figure 1. Ontology Example

4. ONTOLOGY-BASED SEMANTIC EXTRACTION

The task of the previous module, standard formalization, is usually a one-time effort (though it is an iterative process) and it needs significant domain experts' involvement. This module is different in that it is used in every work-flow/task and extracted semantics can be accumulated in repository and used for improving future semantic association performance. Also, it can be relatively automated by using general linguistic processing technologies.

Standards, such as UniformatII and MasterFormat, addressed in this paper are functionally complementary to each other in an application domain and they are costly cross-referenced by domain experts in workflows due to their complexity (vast many-to-many mappings). This module basically is to automat the process by mimicking a domain expert doing cross-referencing from the context of a standard-compliant project specification, a script representation indexed of the standard, which defines intentionality. For example, the following text is quoted from a PPD [7] under entity B2010 in UniformatII taxonomy:

B SHELL

B20 EXTERIOR CLOSURE

B2010 EXTERIOR WALLS

1. *Exterior Wall **Framing**: **Cold-formed**, light gage **steel studs**, C-shape, galvanized finish, 6" metal thickness as designed by manufacturer according to American Iron and Steel Institute (AISI) Specification for the Design of Cold Formed Steel Structural Members, for L/240 deflection. Downside: specifications often contain note-style sentences.*

Supposedly, the PPD is written by an architect and a contractor wants to estimate cost for exterior walls. He might comprehend that the wall framing will be made of cold-formed steel studs (semantic). Based on his expertise, he identifies that its corresponding entity in Masterformat is 05410 Load bearing metal studs (association). The following paragraph shows how the ontology/relation statements being used for discovering the semantic under the context of entity B2010 that links the entity to MasterFormat entity 05410 (semantic association):

B2010 Exterior Wall:

*1. Exterior Wall **Framing**: **Cold-formed**, light gage **steel studs**, C-shape, galvanized finish, 6" metal thickness*

05100_1 → 05400 → **05410**

In the diagram, "steel" and "framing" match the statement 05100_1 (one of the identifiers of the relation statements exemplified in previous subsection) which is Metals (D5) *used_for* framing. The "steel" matches "Metals" through the transitive property of the relation, *kind_of*. The match is extended to statement 05400, which includes "cold-formed". Finally "studs" is added to the match of statement 05410, through statement (05400_05410). Indeed the entity B2010 Exterior Wall in UniformatII has a semantic relationship with 05410 Load bearing metal studs in MasterFormat and the semantic can be described by the relation *made_of*.

One characteristic worthy of mentioning is that the entity B2010 Exterior Wall in the taxonomy provides a good context for helping refining the association. For instance, the above matching, even without the "framing" keyword, is still possible since the inherited semantic of the hierarchy, shell, closure, and exterior walls, has very close meaning as framing.

As shown in the above example, the documents or specifications that this research addresses have following characteristics:

1. Content has limited scope. It often details what, where, how, and when objects and activities being involved in a domain application. It usually contains rich semantics (author's intention for communicating with other stakeholders) related to standards (due to the agreement among stakeholders) that coordinate objects and activities in the domain.
2. Content are categorized according to taxonomy. In other words, text in a document has some assumption or context, which is inherited along the taxonomy hierarchy.
3. Terminologies are relatively unified and unambiguous.
4. Sentences are relatively free styled, such as note-styled or template-styled due to writing convention or standards.

These characteristics distinguish this research from others, such as [9] and [15] which extract shallow information from general or web documents.

In addition to the intrinsic semantics of standards, this module also explores their application or context semantics in order to achieve more effective semantic extraction. The application semantics depend on the stakeholders' view or interests, such as information they intent for. For example, a cost estimator might look for MasterFormat items and some numerical information so that they can link them to their MasterFormat-based cost databases. On the other hand, an inspector might be interested in the same information but in different view points that yield to different semantics. For example, to a cost estimator, "*6" metal thickness*" in the PPD means how much the studs with such thickness cost. But for an inspector, it means 6" thickness compliance to associated code.

In summary, this module extracts semantics from the instances (specifications) of multiple standards based on three kinds of ontologies: the ontology of the source standard, the ontology of target standard, and the application ontology based on the stakeholders' views. The extracted semantics are evidences of semantic association of entities between source and target standards.

5. MEASUREMENT OF AFFINITY

The ontology-based semantic extraction module can be implemented via a matching process between relation statements and text. The goal is to identify a set of matched relation statements of related entities with respect to their standards. For a given entity, its associated relation statements carry different weights depending on their positions in the taxonomy and the information content [16] of their keywords. The measurement of affinity is to quantify the weights so that the degree of the closeness between matched relation statements and their associated entity can be determined. Based on the measurement, a ranking scheme can be devised to identify optimal semantic associations among all matches. The ranking scheme can be modeled as a function of the following factors:

1. Number of relation statements matched.
2. Number of keywords matched.
3. Quality of the matches. The measurement of the quality is an open question. Basically the more specific the matches are, the higher quality they represent. One effective way to model the quality is by their positions in the taxonomy (higher level means less specific and thus carries less weight) and by the information content of their keywords. The information content can be quantified by their inverse document frequency (IDF) [17] combined with their counts in the taxonomy (appearing more times means less specific and thus carries less weight)

For instance, in the given example, several entities in MasterFormat contain "framing" and "Metals", which are all candidates for semantic association. The entity 05410 is considered as the optimal one because it matches more keywords along its taxonomy hierarchy and some of them, such as studs, are very specific with respect to both position and IDF.

6. ARCHITECTURE

The major thrust of the research is to develop an integration framework that facilitates exploitation of semantics from taxonomy-based standards and instantiations of the standards to achieve higher interoperability between domain participants and their information systems. To demonstrate the applicability of the proposed approach toward the goal, this section shows an overall architecture depicting one possible implementation and its relationship with other related technologies.

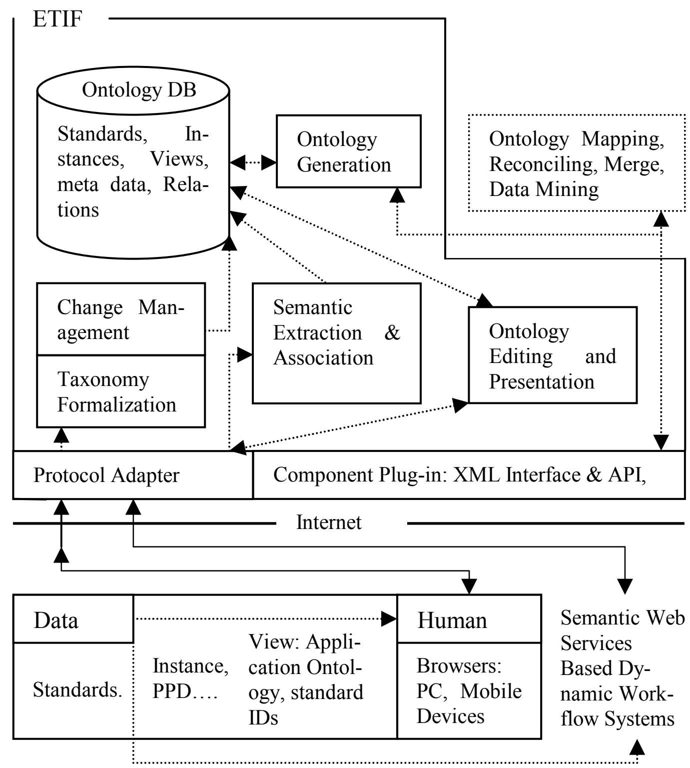


Figure 2. Extensible Taxonomy-based Integration Framework (ETIF)

In the framework shown in Figure 2, relations and relation statements of various versions of standards written in natural languages are developed and uploaded via web-based tools to the system by stakeholders in the application domain. The taxonomy formalization along with the change management modules process them through parsing, normalization, generalization, linguistic processing (such as inflection, derivation, compounds, and synonyms), and indexing for incremental update in the ontology database. For a particular application, the stakeholders upload instances of the source standard (e.g., PPDs), target standard, and its application ontology. After processing the free text of PPD instances through linguistic techniques such as tokenization, chunk parsing, and grammatical function recognition [9], the system applies the semantic extraction and ranking algorithms, and returns/deposits extracted metadata and semantic association to the ontology database and also to the users or clients, if applicable, for feedback.

The integration of competing and complementary standards is a critical step for enhancing interoperability among heterogeneous systems using the standards. The proposed semantic association is only one aspect in this effort. It should be supplemented with other technologies such as ontology mapping, reconciling, and merging to provide a practical and complete solution. The framework includes a plug-in mechanism via XML-based interfaces and API for external software component integration.

The formalized standards, their instances, users' application ontologies, and extracted metadata form a semantic rich ontology repository. Integrating the repository with other ontology techniques through the plug-in mechanism allows the effective construction of application domain ontology.

Web services enriched with the vision of the semantic web have emerged as a mainstream solution to system integration over the Internet. Following the same trend, the implementation of the proposed framework adopts the Web Ontology Language (OWL) [8] with the intention of integrating building construction workflow systems via semantic web services.

7. CONCLUSION AND FUTURE WORKS

This paper demonstrates the effective use of taxonomy for ontology developments and the semantic association of ontology for interoperability in a workflow system with building construction as the target example. It illustrates a systematic approach to semantic association through taxonomy formalization and ontology-based semantic extraction. The overall system implementation in web environment is also proposed. Current activities of the research project include the complete ontological formalization of the MaterFormat and UniformatII standards, refinement of the affinity measure for general taxonomy, and the integration of the algorithms with dynamic workflow systems through semantic web services.

8. ACKNOWLEDGMENTS

This work is partially supported by an NSF research grant ITR-0404113.

REFERENCES

[1] Construction Specifications Institute. MasterFormat 95™ : *Alexandria, VA: The Construction Specifications Institute, 1995 edition.*

[2] Charette, R. P. and Marshall, H. E.: UNIFORMAT II Elemental Classification for Building Specifications, Cost Estimating, and Cost Analysis, *NISTIR 6389, Gaithersburg, MD: National Institute of Standards and Technology, October, 1999*

[3] Gallaher, M. P.; O'Connor, A. C.; Dettbarn, J. L., Jr.; Gilday, L. T.: Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry, *NIST GCR 04-867, Gaithersburg, MD: National Institute of Standards and Technology, August, 2004.*

[4] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm: Generic Schema Matching with Cupid, *at the Twenty Seventh International Conference on Very Large Databases (VLDB'2001), Roma, Italy.*

[5] N.F. Noy and M.A. Musen. The prompt suite: Interactive tools for ontology merging and mapping. *Journal of Human-Computer Studies, 59(6):983--1024, 2003.*

[6] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. *In The First International Semantic Web Conference (ISWC), 2002.*

[7] Rosen, Harold J. : Construction specifications writing : principles and procedures 5th edition, *Hoboken, N.J. : J. Wiley, c2005.*

[8] Mike Dean and Guus Schreiber: Editors OWL Web Ontology Language Reference, *W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210, 10 February 2004.*

[9] Maedche, A., Neumann, G., Staab, S.: Bootstrapping an Ontology-Based Information Extraction System, *Intelligent Exploration of the Web, Springer 2002.*

[10] Gruber, T.R., A Translation Approach to Portable Ontology Specification: *Knowledge Acquisition 5: 199-220, 1993.*

[11] Rahm, E and Bernstein, P. A. "A Survey of Approaches to Automatic Schema Matching." *The VLDB Journal, Vol. 10, pp. 334-350, 2001.*

[12] Do, H., Melnik, S. and Rahm, E. "Comparison of Schema Matching Evaluations." In Proceedings of the 2nd Int. *Workshop on Web Databases (German Informatics Society), 2002.*

[13] Aberber, K., Cudré-Mauroux, P. and Hauswirth, M. "The Chatty Web: Element Semantics through Gossiping." *The Proceedings of the 20th International World Wide Web Conference, pp. 197 – 206, 2003.*

[14] Doan, A., Madhavan, J., Domingos, P. and Halevy, A. "Learning to Map between Ontologies on the Semantic Web." *The VLDB Journal, Vol. 12, pp. 303-319, 2003.*

[15] David W. Embley , Douglas M. Campbell , Randy D. Smith , Stephen W. Liddle.: Ontology-based extraction and structuring of information from data-rich unstructured documents, *Proceedings of the seventh international conference on Information and knowledge management, p.52-59, November 02-07, 1998, Bethesda, Maryland, United States*

[16] Ross, S.: A First Course in Probability. *Macmillan Publishing, 1976.*

[17] Church, K. W. and Gale, W. A. : Inverse document frequency (IDF): A measure of deviations from Poisson. In Yarowsky, D. and Church, K., editors, *Proceedings of the Third Workshop on Very Large Corpora, pages 121--130. Association for Computational Linguistics. 1995.*

Relaxed Precision and Recall for Ontology Matching

Marc Ehrig
Institute AIFB
University of Karlsruhe
Karlsruhe, Germany
ehrig@aifb.uni-karlsruhe.de

Jérôme Euzenat
INRIA Rhône-Alpes
655 avenue de l'Europe.
Monbonnot, France
Jerome.Euzenat@inrialpes.fr

ABSTRACT

In order to evaluate the performance of ontology matching algorithms it is necessary to confront them with test ontologies and to compare the results. The most prominent criteria are precision and recall originating from information retrieval. However, it can happen that an alignment be very close to the expected result and another quite remote from it, and they both share the same precision and recall. This is due to the inability of precision and recall to measure the closeness of the results. To overcome this problem, we present a framework for generalizing precision and recall. This framework is instantiated by three different measures and we show in a motivating example that the proposed measures are prone to solve the problem of rigidity of classical precision and recall.

Categories and Subject Descriptors

D.2.12 [**Software**]: Interoperability; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods; D.2.8 [**Software Engineering**]: Metrics

General Terms

Measurement, Performance, Experimentation

Keywords

Ontology alignment, evaluation measures, precision, recall

1. INTRODUCTION

Ontology matching is an important problem for which many algorithms (e.g., PROMPT[11], GLUE[3], Ontrapro[1], OLA[7], FOAM[4]) have been provided. In order to evaluate the performance of these algorithms it is necessary to confront them with test ontologies and to compare the results. The most prominent criteria are precision and recall originating from information retrieval and adapted to the ontology matching task. Precision and recall are based on the comparison of the resulting alignment with another standard alignment, effectively comparing which correspondences are found and which are not. These criteria are well understood and widely accepted.

However, as we have experienced in last year's Ontology Alignment Contest [13], they have the drawback to be of the all-or-nothing kind. An alignment may be very close to the expected result and another quite remote from it and both return the same precision and recall. The reason for this is that the criteria only compare two sets of correspondences without considering if these are close or remote to each other: if they are not the same exact correspondences, they score zero. They both score identically low, despite their different quality. It may be helpful for users to know whether the found alignments are close to the expected one and easily repairable or not. It is thus necessary to measure the proximity between alignments instead of their strict equality.

In this paper we investigate some measures that generalize precision and recall in order to overcome the problems presented above. We first provide the basic definitions of alignments, precision and recall as well as a motivating example (§2). We then present a framework for generalizing precision and recall (§3). This framework is instantiated by four different measures (including classical precision and recall) (§4) and we show on the motivating example that the proposed measures do not exhibit the rigidity of classical precision and recall (§5).

K-CAP'05, Workshop Integrating Ontologies, October 2, 2005, Banff, Alberta, Canada.

2. FOUNDATIONS

2.1 Alignment

DEFINITION 1 (ALIGNMENT, CORRESPONDENCE). *Given two ontologies O and O', an alignment between O and O' is a set of correspondences (i.e., 4-uples): $\langle e, e', r, n\rangle$ with $e \in O$ and $e' \in O'$ being the two matched entities, r being a relationship holding between e and e', and n expressing the level of confidence* $[0..1]$ *in this correspondence.*

A matching algorithm returns an alignment A which is compared with a reference alignment R.

Let us illustrate this through a simple example. Figure 1 presents two ontologies together with two alignments A_1 and R. In this example, for the sake of simplification, the relation is always '=' and the confidence is always 1.0.

The alignment A_1 is defined as follows:

```
<o1:Vehicle,o2:Thing,=,1.0>
<o1:Car,o2:Porsche,=,1.0>
<o1:hasSpeed,o2:hasProperty,=,1.0>
<o1:MotorKA1,o2:MarcsPorsche,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
```

We present another reasonable alignment A_2:

```
<o1:Car,o2:Thing,=,1.0>
<o1:hasSpeed,o2:hasProperty,=,1.0>
<o1:MotorKA1,o2:MarcsPorsche,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
```

and an obviously wrong alignment A_3:

```
<o1:Object,o2:Thing,=,1.0>
<o1:Owner,o2:Volkswagen,=,1.0>
<o1:Boat,o2:Porsche,=,1.0>
<o1:hasOwner,o2:hasMotor,=,1.0>
<o1:Marc,o2:fast,=,1.0>
```

Further, we have the following reference alignment (R):

```
<o1:Object,o2:Thing,=,1.0>
<o1:Car,o2:Automobile,=,1.0>
<o1:Speed,o2:Characteristic,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
<o1:PorscheKA123,o2:MarcsPorsche,=,1.0>
```

2.2 Precision and Recall

The usual approach for evaluating the returned alignments is to consider them as sets of correspondences and check for the overlap of the two sets. This is naturally obtained by applying the classical measure of precision and recall [14], which are the ratio of the number of true positive ($|R \cap A|$) and retrieved correspondences ($|A|$) or those to be retrieved ($|R|$), respectively.

DEFINITION 2 (PRECISION, RECALL). *Given a reference alignment R, the precision of some alignment A is given by*

$$P(A, R) = \frac{|R \cap A|}{|A|}$$

and recall is given by

$$R(A, R) = \frac{|R \cap A|}{|R|}.$$

2.3 Problems with Current Measures

However, even if the above measurements are easily understandable and widespread, they are often criticized for two reasons: Neither do they discriminate between a totally wrong and an almost correct alignment, nor do they measure user effort to adapt the alignment.

Indeed, it often makes sense to not only have a decision whether a particular correspondence has been found or not, but measure the proximity of the found alignments. This implies that also "near misses" are taken into consideration instead of only the exact matches.

As a matter of example, it will be clear to anybody that among the alignments presented above, A_3 is not a very good alignment and A_1 and A_2 are better alignments. However, they score almost exactly the same in terms of precision (.2) and recall (.2).

Moreover, the alignments will have to go through user scrutiny and correction before being used. It is worth measuring the effort required by the user for correcting the provided alignment instead of only if some correction is needing. This also calls for a relaxation of precision and recall.

3. GENERALIZING PRECISION AND RECALL

Because precision and recall are well-known and easily explained measures, it is good to adhere to them and extend them. It also brings the benefit that measures derived from precision and recall, such as f-measure, can still be computed. For these reasons, we propose to generalize these measures.

If we want to generalize precision and recall, we should be able to measure the proximity of correspondence sets rather than their strict overlap. Instead of the taking the cardinal of the intersection of the two sets ($|R \cap A|$), we measure their proximity (ω).

DEFINITION 3 (GENERALIZED PRECISION AND RECALL). *Given a reference alignment R and an overlap function ω between alignments, the precision of an alignment A is given by*

$$P_\omega(A, R) = \frac{\omega(A, R)}{|A|}$$

and recall is given by

$$R_\omega(A, R) = \frac{\omega(A, R)}{|R|}.$$

3.1 Basic properties

In order, for these new measures to be true generalizations, we would like ω to share some properties with $|R \cap A|$. In particular, the measure should be positive:

$$\forall A, B, \omega(A, B) \geq 0 \qquad \text{(positiveness)}$$

and not exceeding the minimal size of both sets:

$$\forall A, B, \omega(A, B) \leq min(|A|, |B|) \qquad \text{(maximality)}$$

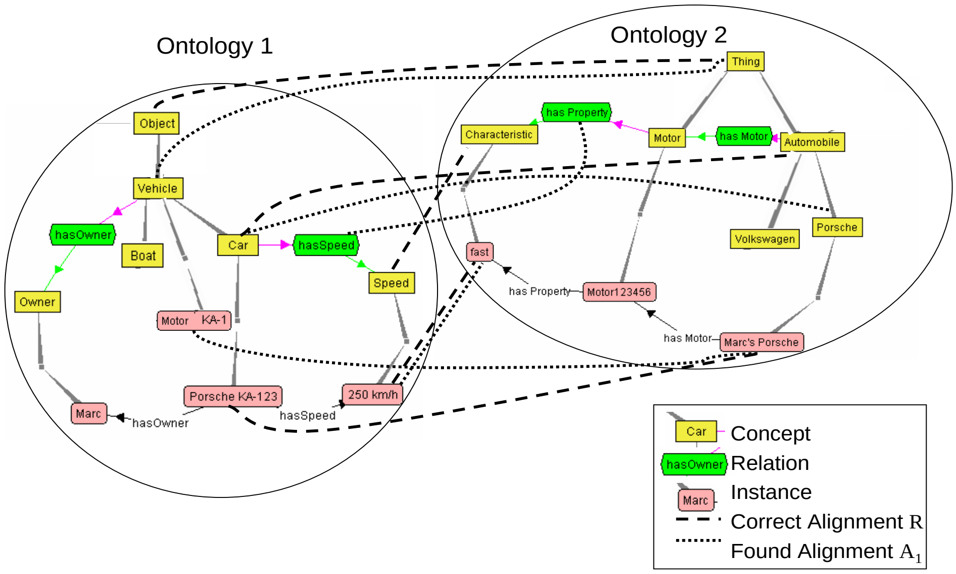


Figure 1: Two Aligned Ontologies

If we want to preserve precision and recall results, ω should only add more flexibility to the usual precision and recall. So their values cannot be worse than the initial evaluation:

$$\forall A, B, \omega(A, B) \geq |A \cap B| \qquad \text{(boundedness)}$$

Hence, the main constraint faced by the proximity is the following:

$$|A \cap R| \leq \omega(A, R) \leq min(|A|, |R|)$$

This is indeed a true generalization because, $|A \cap R|$ satisfies all these properties. One more property satisfied by precision and recall that we will not enforce here is symmetry. This guarantees that the precision and recall measures are true normalized similarities.

$$\forall A, B, \omega(A, B) = \omega(B, A) \qquad \text{(symmetry)}$$

We will not require symmetry, especially since A and R are not in symmetrical positions.

3.2 Designing Overlap Proximity

There are many different ways to design such a proximity given two sets. We retain here the most obvious one which consists of finding correspondences matching each other and computing the sum of their proximity. This can be defined as an overlap proximity:

DEFINITION 4 (OVERLAP PROXIMITY). *A measure that would generalize precision and recall is:*

$$\omega(A, R) = \sum_{\langle a,r\rangle \in M(A,R)} \sigma(a, r)$$

in which $M(A, R)$ is a matching between the correspondences of A and R and $\sigma(a, r)$ a proximity function between two correspondences.

Again, the standard overlap $|A \cap R|$ used in precision and recall is such an overlap proximity.

There are two tasks to fulfill when designing such an overlap proximity function:

- the first one consists of finding the correspondences to be compared M.
- the second one is to define a proximity measure on correspondences σ;

We consider these two issues below.

3.3 Matching Correspondences

A matching between alignments is a set of correspondence pairs, i.e., $M(A, R) \subseteq A \times R$. However, if we want to keep the analogy with precision and recall, it will be necessary to restrict ourselves to the matchings in which an entity from the ontology does not appear twice. This is compatible with precision and recall for two reasons: (i) in these measures, any correspondence is identified only with itself, and (ii) appearing more than once in the matching would not guarantee an overlap proximity below $min(|A|, |R|)$.

There are $\frac{|A|!}{(|A|-|R|)!}$ candidate matches (if $|A| \geq |R|$). The natural choice is to select the best match because this guarantees that the function generalizes precision and recall.

DEFINITION 5 (BEST MATCH). *The best match $M(A, R)$ between two sets of correspondences A and R, is the subset of $A \times R$ which maximizes the overall proximity and in which each element of A (resp. R) belongs to only one pair:*

$$M(A, R) \in Max_{\omega(A,R)}\{M \subseteq A \times R\}$$

As defined here, this best match may not be unique. This is not a problem, because we only want to find the highest value for ω and any of the best matches will yield the same value.

Of course, the definitions M and ω are dependent of each other, but this does not prevent us from computing them. They are usually computed together but it is better to present them separately.

3.4 Correspondence Proximity

In order to compute $\omega(A, R)$, we need to measure the proximity between two matched correspondences (i.e., $\langle a, r\rangle \in M(A, R)$) on the basis of how close the result is from the ideal one. Each element in the tuple $a = \langle e_a, e'_a, r_a, n_a\rangle$ will be compared with its counterpart in $r = \langle e_r, e'_r, r_r, n_r\rangle$. For any two correspondences (the found a and the reference r) we compute three similarities σ_{pair}, σ_{rel}, and σ_{conf}. If elements are identical, proximity has to be one (maximality). If they differ, proximity is lower, always according to the chosen strategy. In contrast to the standard definition of similarity, the mentioned proximity measures do not necessarily have to be symmetric. We will only consider normalized proximities, i.e., measures whose values are within the unit interval [0..1], because this guarantees that

$$\omega(A, R) \leq min(|A|, |R|)$$

The component proximity measure is defined in the following way:

$\sigma_{pair}(\langle e_a, e_r\rangle, \langle e'_a, e'_r\rangle)$: How is one entity pair similar to another entity pair? In ontologies we can in principal follow any relation which exists (e.g., subsumption, instantiation), or which can be derived in a meaningful way. The most important parameters are the relations to follow and their effect on the proximity.

$\sigma_{rel}(r_a, r_r)$: Often the alignment relations are more complex, e.g., represent subsumption, instantiation, or compositions. Again, one has to assess the similarity between these relations. The two relations of the alignment cell can be compared based on their distance in a conceptual neighborhood structure [6, 8].

$\sigma_{conf}(n_a, n_r)$: Finally, one has to decide, what to do with different levels of confidence. The similarity could simply be the difference. Unfortunately, none of the current alignment approaches have an explicit meaning attached to confidence values, which makes it rather difficult in defining an adequate proximity.

Once these proximities are established, they have to be aggregated. The constraints on the aggregation function ($Aggr$) are:

- normalization preservation (if $\forall i, 0 \leq c_i \leq 1$ then $0 \leq Aggr_i c_i \leq 1$);
- maximality (if $\forall i, c_i = 1$ then $Aggr_i c_i = 1$);
- local monotonicity (if $\forall i \neq j, c_i = c'_i = c''_j$ and $c_j \leq c'_j \leq c''_j$ then $Aggr_i c_i \leq Aggr_i c'_i \leq Aggr_i c''_i$).

Here, we consider aggregating them through multiplication without further justification. Other aggregations (e.g., weighted sum) are also possible.

DEFINITION 6 (CORRESPONDENCE PROXIMITY). *Given two correspondences $\langle e_a, e'_a, r_a, n_a\rangle$ and $\langle e_r, e'_r, r_r, n_r\rangle$, their proximity is:*

$$\sigma(\langle e_a, e'_a, r_a, n_a\rangle, \langle e_r, e'_r, r_r, n_r\rangle) =$$

$$\sigma_{pair}(\langle e_a, e_r\rangle, \langle e'_a, e'_r\rangle) \times \sigma_{rel}(r_a, r_r) \times \sigma_{conf}(n_a, n_r)$$

We have provided constraints and definitions for M, ω, and σ. We now turn to concrete measures.

4. CONCRETE MEASURES

We consider four cases of relaxed precision and recall measures based on the above definitions. We first give the definition of usual precision and recall within this framework.

4.1 Standard Precision and Recall

For standard precision and recall, the value of ω is $|A \cap R|$. This is indeed an instance of this framework, if the proximity used is based on the strict equality of the components of correspondences.

DEFINITION 7 (EQUALITY PROXIMITY). *The equality* *proximity is charaterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) = \begin{cases} 1 & \textit{if } \langle e_a, e'_a\rangle = \langle e_r, e'_r\rangle \\ 0 & \textit{otherwise} \end{cases}$$

$$\sigma_{rel}(r_a, r_r) = \begin{cases} 1 & \textit{if } r_a = r_r \\ 0 & \textit{otherwise} \end{cases}$$

$$\sigma_{conf}(n_a, n_r) = \begin{cases} 1 & \textit{if } n_a = n_r \\ 0 & \textit{otherwise} \end{cases}$$

4.2 Symmetric Proximity

The easiest way to relax precision and recall is to have some distance δ on the elements in ontologies and to weight the proximity with the help of this distance: the higher the distance between two entities in the matched correspondences, the lower their proximity. This can be defined as:

$$\left.\begin{array}{rl} & \delta(e_a, e_r) \leq \delta(e_b, e_r) \\ \text{and} & \delta(e'_a, e'_r) \leq \delta(e'_b, e'_r) \end{array}\right\}$$

$$\implies \sigma(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \geq \sigma(\langle e_b, e'_b\rangle, \langle e_r, e'_r\rangle)$$

As a simple example of such a symmetric similarity, we use a distance in which a class is at distance 0 of itself, at distance 0.5 of its direct sub- and superclasses, and at a distance 1 of any other class. This could be further refined by having a similarity inversely proportional to the distance in the subsumption tree. Likewise, this similarity may also be applied to properties and instances (through part-of relationships in the latter case). The similarity between pairs is the complement of these similarities The result is displayed in Table 1. We always mention the assumed alignment and the actual correct alignment.

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	0.5	returns more specialized instances
c,c'	$sup(c)$,c'	0.5	returns more general instances
c,c'	c,$sub(c')$	0.5	returns more general instances
c,c'	$sub(c)$,c'	0.5	returns more specialized instances
r,r'	r,$sup(r')$	0.5	returns more spec. relation instances
r,r'	$sup(r)$,r'	0.5	returns more gen. relation instances
r,r'	r,$sub(r')$	0.5	returns more gen. relation instances
r,r'	$sub(r)$,r'	0.5	returns more spec. relation instances
i,i'	i,$super(i')$	0.5	returns a more restricted instance
i,i'	$super(i)$,i'	0.5	returns a too broad instance
i,i'	i,$sub(i')$	0.5	returns a too broad instance
i,i'	$sub(i)$,i'	0.5	returns a more restricted instance

Table 1: Similarities based on Entity Pairs

Table 2 consider the proximity between relations. It only presents the similarity between equality (=) and other relations.

For the confidence distance we simply take the complement of the difference. The final precision is calculated according to the formula presented in the previous section:

found relation	correct relation	similarity σ_{rel}	comment
$e = e'$	$e = e'$	1	correct relation
$c = c'$	$c \subset c'$	0.5	returns more instances than correct
$c = c'$	$c \supset c'$	0.5	returns less instances than possible, but these are correct
$r = r'$	$r \subset r'$	0.5	
$r = r'$	$r \supset r'$	0.5	
$i = i'$	i partOf i'	0.5	
$i = i'$	i consistsOf i'	0.5	

Table 2: Similarities based on Relations

DEFINITION 8 (SYMMETRIC PROXIMITY). *The symmetric proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \textit{ as defined in Table 1}$$
$$\sigma_{rel}(r_a, r_r) \textit{ as defined in Table 2}$$
$$\sigma_{conf}(n_a, n_r) = 1 - |n_a - n_r|.$$

4.3 Measuring Correction Effort

If users have to check and correct alignments, the quality of alignment algorithms can be measured through the effort required for transforming the obtained alignment into the (correct) reference one [2].

This measure can be implemented as an edit distance [10]: an edit distance defines a number of operations by which an object can be corrected (here the the operations on correspondences authorized) and assigns a cost to each of these operations (here the effort required to identify and repair some mistake). The cost of a sequence of operations is the sum of their cost and the distance between two objects is the cost of the less costly sequence of operations that transform one object into the other one. The result can always be normalized in function of the size of the largest object. Such a distance can be turned into a proximity by taking its complement with regard to 1.

Table 3 provides such plausible weights. Usually classes are organized in a taxonomy in which they have less direct super- than subclasses. It is thus easier to correct a class to (one of) its superclass than to one of its subclasses. As a consequence, the proximity is dissymmetric. Such a measure should also add some effort when classes are not directly related, but this has not been considered here.

The edit distance between relations is relatively easy to design since, generally, changing from one relation to another can be done with just one click. Thus, the relational similarity equals 1 if the relations are the same and 0.5 otherwise.

In this correction effort measure, the confidence factor does not play an important role: ordering the correspondences can only help the user to know that after some point she will have to discard many correspondences. We thus decided to not take confidence into account and thus, their proximity will always be 1.

found e,e'	closest correct e,e'	effort	similarity σ_{pair}	comment
e,e'	e,e'	0	1	correct alignment
c,c'	c,$sup(c')$	0.4	0.6	returns more spec. instances
c,c'	$sup(c)$,c'	0.4	0.6	returns more gen. instances
c,c'	c,$sub(c')$	0.6	0.4	returns more gen. instances
c,c'	$sub(c)$,c'	0.6	0.4	returns more spec. instances
r,r'	r,$sup(r')$	0.4	0.6	
r,r'	$sup(r)$,r'	0.4	0.6	
r,r'	r,$sub(r')$	0.6	0.4	
r,r'	$sub(r)$,r'	0.6	0.4	
i,i'	i,$super(i')$	0.4	0.6	returns a more restricted inst.
i,i'	$super(i)$,i'	0.4	0.6	returns a too broad inst.
i,i'	i,$sub(i')$	0.6	0.4	returns a too broad inst.
i,i'	$sub(i)$,i'	0.6	0.4	returns a more restricted inst.

Table 3: Effort-based proximity between Entity Pairs

DEFINITION 9 (EFFORT-BASED PROXIMITY). *The effort-based proximity is charaterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \textit{ as defined in Table 3}$$
$$\sigma_{rel}(r_a, r_r) = \begin{cases} 1 & \textit{if } r_a = r_r \\ 0.5 & \textit{otherwise} \end{cases}$$
$$\sigma_{conf}(n_a, n_r) = \begin{cases} 1 & \textit{if } n_a \neq 0 \textit{ and } n_r \neq 0 \\ 0 & \textit{otherwise} \end{cases}$$

To be accurate, such an effort proximity would have been better aggregated with an additive and normalized aggregation function rather than multiplication.

4.4 Precision- and Recall-oriented Measures

One can also decide to use two different similarities depending on their application for evaluating either precision or recall. We here provide two such measures and justify the given weights. Precision is normally a measure of accuracy i.e., the returned results need to be correct. Every wrong result will therefore entail a penalty. We assume the user poses a query to the system as follows: "return me all instances of e". The system then returns any instance corresponding to the alignment i.e. e'. Vice versa, for the relaxed recall we want to avoid missing any correct result. This affects the similarity relations and weights.

4.4.1 Relaxed Precision

In Table 4 and 5 we present the precision similarity for pairs and relations. The comments in each line explain the decision for the weights.

For the distance within the confidence we again use the complement of the difference.

DEFINITION 10 (PRECISION-ORIENTED PROXIMITY). *The precision-recall oriented proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \textit{ as defined in Table 4}$$
$$\sigma_{rel}(r_a, r_r) \textit{ as defined in Table 5}$$
$$\sigma_{conf}(n_a, n_r) = 1 - |n_a - n_r|.$$

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	1	returns more specialized instances, these are correct
c,c'	$sup(c)$,c'	0.5	returns more general instances, includes some correct results
c,c'	c,$sub(c')$	0.5	returns more general instances, includes some correct results
c,c'	$sub(c)$,c'	1	returns more specialized instances, these are correct
r,r'	r,$sup(r')$	1	
r,r'	$sup(r)$,r'	0.5	
r,r'	r,$sub(r')$	0.5	
r,r'	$sub(r)$,r'	1	
i,i'	i,$super(i')$	0.5	returns a more restricted instance
i,i'	$super(i)$,i'	0	returns a too broad instance
i,i'	i,$sub(i')$	0	returns a too broad instance
i,i'	$sub(i)$,i'	0.5	returns a more restricted instance

Table 4: Similarities for Relaxed Precision based on Entity Pairs

found relation	correct relation	similarity σ_{rel}	comment
e = e'	e = e'	1	correct relation
c = c'	$c \subset c'$	0.5	returns more instances than correct
c = c'	$c \supset c'$	1	returns less instances than possible, but these are correct
r = r'	$r \subset r'$	0.5	
r = r'	$r \supset r'$	1	
i = i'	i partOf i'	0.5	
i = i'	i consistsOf i'	1	

Table 5: Similarities for Relaxed Precision based on Relations

4.4.2 Relaxed Recall

In Table 6 and 7 we present the recall similarity for pairs and relations. Basically many distances are just mirrored compared to the precision case.

found e,e'	closest correct e,e'	similarity σ_{pair}	comment
e,e'	e,e'	1	correct correspondence
c,c'	c,$sup(c')$	0.5	returns more specialized instances, misses some
c,c'	$sup(c)$,c'	1	returns more general instances, includes the correct results
c,c'	c,$sub(c')$	1	returns more general instances, includes the correct results
c,c'	$sub(c)$,c'	0.5	returns more specialized instances, misses some
r,r'	r,$sup(r')$	0.5	
r,r'	$sup(r)$,r'	1	
r,r'	r,$sub(r')$	1	
r,r'	$sub(r)$,r'	0.5	
i,i'	i,$super(i')$	0	returns a more restricted instance, misses correct
i,i'	$super(i)$,i'	0.5	returns a broader instance
i,i'	i,$sub(i')$	0.5	returns a broader instance
i,i'	$sub(i)$,i'	0	returns a more restricted instance, misses correct

Table 6: Similarities for Relaxed Recall based on Entity Pairs

found relation	correct relation	similarity σ_{rel}	comment
$e = e'$	$e = e'$	0	correct relation
$c = c'$	$c \subset c'$	0	returns more instances than correct
$c = c'$	$c \supset c'$	0.5	returns less instances than possible, misses some
$r = r'$	$r \subset r'$	0	
$r = r'$	$r \supset r'$	0.5	
$i = i'$	i partOf i'	0	
$i = i'$	i consistsOf i'	0.5	

Table 7: Similarities for Relaxed Recall based on Relations

The final recall is computed as usual:

DEFINITION 11 (RECALL-ORIENTED PROXIMITY). *The recall-oriented proximity is characterized by:*

$$\sigma_{pair}(\langle e_a, e'_a\rangle, \langle e_r, e'_r\rangle) \textit{ as defined in Table 6}$$
$$\sigma_{rel}(r_a, r_r) \textit{ as defined in Table 7}$$
$$\sigma_{conf}(n_a, n_r) = 1 - |n_a - n_r|.$$

5. EXAMPLE

In the introduction of this paper we have presented a pair of ontologies, the reference alignment, and three different identified alignments. We will now apply the different proposed precision and recall measures to these example alignments. Please note that they mainly illustrate entity pair similarities, as relations and confidences are always identical. Table 8 provides the results. For the oriented measure we assume that the query is given in ontology 1 and the answer has to be retrieved in ontology 2. As the oriented measure is dissymmetric, one has to define this direction beforehand.

ω	(R, R)		(R, A_1)		(R, A_2)		(R, A_3)	
	P	R	P	R	P	R	P	R
standard	1.0	1.0	0.2	0.2	0.25	0.2	0.2	0.2
symmetric	1.0	1.0	0.4	0.4	0.375	0.3	0.2	0.2
edit	1.0	1.0	0.44	0.44	0.35	0.28	0.2	0.2
oriented	1.0	1.0	0.5	0.5	0.375	0.4	0.2	0.2

Table 8: Precision recall result on the alignments of Figure 1

The measures which have been introduced address the problems raised in the introduction and fulfill the requirements:

– They keep precision and recall untouched for the best alignment (R);
– They help discriminating between irrelevant alignments (A_3) and not far from target ones (A_1 and A_2);
– Specialized measures are able to emphasize some characteristics of alignments: ease of modification, correctness or completeness. For instance, let's consider the oriented measures. In our example A_1 has two very near misses, which leads to a relatively high precision. In A_2 however the miss is bigger, but by aligning one concept to its superconcept recall rises relatively to precision.

These results are based on only one example. They have to be systematized in order to be extensively validated. Our goal is to implement these measures within the Alignment API and to use them on the forthcoming results of the Ontology Alignment Evaluation 2005[1] in order to have real data on which the relevance of the proposed measures can be more openly debated.

6. RELATED WORK

The naturally relevant work is [2] which has considered precisely the evaluation of schema matching. However, the authors only note the other mentioned problem (having two measures instead of one) and use classical aggregation (overall and F-measure) of precision and recall.

In computational linguistics, and more precisely multilingual text alignment, [9] has considered extending precision and recall. Their goal is the same as ours: increasing the discriminating power of the measures. In this work, the mathematical formulation is not changed but the granularity of compared sets changes: instead of comparing sentences in a text, they compare words in sentences in a text. This helps having some contribution to the measures when most of the words are correctly aligned while the sentences are not strictly aligned.

In the Alignment API [5], there is another evaluation measure which directly computes a distance based on a weighted symmetric difference (weights are the confidences of each correspondence in the alignment). This measure could be used in the generalization proposed here (the distance would then be based on confidence difference and would generally satisfy $P'(A, R) \leq P(A, R)$ and $R'(A, R) \leq R(A, R)$.

The deeper proposal for extending precision and recall comes from hierarchical text categorization in which texts are attached to some category in a taxonomy [12]. Usually, texts are attached to the leaves, but when algorithms attach them to the intermediate categories, it is useful to discriminate between a category which is irrelevant and a category which is an immediate super category of the expected one. For that purpose, they introduce an extension of precision (recall is redefined similarly) such that:

$$P_{CS} = \frac{max(0, |A \cap R| + FpCon + FnCon)}{|A| + FnCon}$$

in which $FpCon$ (resp. $FnCon$) is the contribution to false positive (resp. false negative), i.e., the way incorrectly classified documents could contribute to its incorrect category anyway. The maximization is necessary to prevent the result from being negative (because the contribution is defined with respect to the average such contribution). The contribution is measured in two ways. The first one is a category similarity that is computed on the features of categories (categories and documents are represented by a vector of features and the membership to some category is based on a distance be-

[1] http://oaei.inrialpes.fr/2005/

tween these vectors). The second one is based on the distance between categories in the taxonomy.

This measure does not seem to be a generalization of standard precision and recall as the one presented here. In particular, because the contributions can be negative, this measure can be lower than standard precision and recall. The idea of retracting the contribution from wrongly classified documents is not far from the idea developed here. However, the computation of this contribution with regard to some average and the addition of some contribution to the divisor do not seem justified.

7. DISCUSSION

Evaluation of matching results is often made on the basis of the well-known and well-understood precision and recall measures. However, these measures do not discriminate accurately between methods which do not provide the exact results. In the context where the result of alignments have to be screened by humans, this is an important need.

We have proposed a framework for generalizing precision and recall when comparing ontology alignments. It keeps the advantages of usual precision and recall but helps discriminating between alignments by identifying for near misses instead of completely wrong correspondences.

The framework has been instantiated in three different measures, each one aiming at favoring some particular aspects of alignment utility. We show that these measures indeed avoid the shortcomings of standard evaluation criteria. They should however, be further investigated in order to find better formulations: more discrepancy needs to be considered, more progressive distance (e.g., not direct subclasses) and rationalized design of weights.

This generalization framework is not the only possible one since we have made a number of choices:

- on the form of the alignment similarity (Definition 4);
- on the kind of alignment matching (Definition 5);
- on the form of the correspondence similarity (Definition 6).

More work has to be done in order to assess the potential of other choices in these functions.

The most important work is to consider these proposed measures in real evaluation of alignment systems and to identify good measures for further evaluations. We plan to implement these measures within the Alignment API [5] and process the results of the Ontology Alignment Evaluation 2005.

Acknowledgements

This work has been partially supported by the Knowledge Web European network of excellence (IST-2004-507482). The authors would like to thank Diana Maynard who pointed out the problem addressed here.

8. REFERENCES

[1] B. Ashpole. Ontology translation protocol (ontrapro). In E. Messina and A. Meystel, editors, *Proceedings of Performance Metrics for Intelligent Systems (PerMIS '04)*, Gaithersburg, MD, USA, August 2004.

[2] H.-H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proc. GI-Workshop "Web and Databases", Erfurt (DE)*, 2002. http://dol.uni-leipzig.de/pub/2002-28.

[3] A.-H. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to map ontologies on the semantic web. *VLDB journal*, 2003.

[4] M. Ehrig and S. Staab. QOM - quick ontology mapping. In *Proc. 3rd ISWC, Hiroshima (JP)*, November 2004.

[5] J. Euzenat. An API for ontology alignment. In *Proc. 3rd international semantic web conference, Hiroshima (JP)*, pages 698–712, 2004.

[6] J. Euzenat, N. Layaïda, and V. Dias. A semantic framework for multimedia document adaptation. In *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco (MX)*, pages 31–36, 2003.

[7] J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proc. 15th ECAI*, pages 333–337, Valencia (ES), 2004.

[8] C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1–2):199–227, 1992.

[9] P. Langlais, J. Véronis, and M. Simard. Methods and practical issues in evaluating alignment techniques. In *Proc. 17th international conference on Computational linguistics, Montréal (CA)*, pages 711–717, 1998.

[10] I. V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 1966.

[11] N. Noy and M. Musen. Smart: Automated support for ontology merging and alignment, 1999.

[12] A. Sun and E.-P. Lin. Hierarchical text classification and evaluation. In *Proc. IEEE international conference on data mining*, pages 521–528, 2001.

[13] Y. Sure, O. Corcho, J. Euzenat, and T. Hughes, editors. *Proceedings of the 3rd Evaluation of Ontology-based tools (EON)*, 2004.

[14] C. J. van Rijsbergen. *Information retrieval*. Butterworths, London (UK), 1975. http://www.dcs.gla.ac.uk/Keith/Preface.html.

Searching Web Resources Using Ontology Mappings

Dragan Gašević and Marek Hatala
School of Interactive Arts and Technology, Simon Fraser University Surrey
2400 Central City, 10153 King George Hwy, Surrey, BC V3T 2W1, Canada
Email: {dgasevic, mhatala}@sfu.ca

ABSTRACT
This paper proposes an ontology mapping based framerowk that allows searching for web resources using multiple ontologies. The proposed solution uses a mapping ontology that is a part of a recent Semantic Web initiative called the Simple Knowledge Organization System (SKOS). On top of that, we propose the search algorithm that takes arguments from one ontology and generates queries compliant with other ontologies. We evaluated the solution on a web application that allows using a local ontology, which describes content of a web site, to search for web resources in remote digital libraries or object repositories based on more general content ontologies.

Categories and Subject Descriptors
H.3.3 Information Storage and Retrieval – Information Search and Retrieval

General Terms
Algorithms, Design

Keywords
Ontology, ontology mapping, search, interoperability

1. INTRODUCTION
In a past few years large collections of web resources became available either through the digital libraries (such as ACM Portal), community-based object repositories, or more importantly as widely dispersed web resources in many individual institutions. Several interoperability initiatives are trying to address the issue of searching across multiple object collections. However, the effectiveness of searching is hampered by the fact that individual web resources are typically not interconnected into the web and therefore lacking the context which makes the Google's PageRank algorithm [7] so effective. The libraries and repositories are overcoming this lack of context by providing explicit semantic information in the form of subject categories, taxonomies, or ideally richer ontologies.

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

However, one can hardly find two different object repositories relaying on the same classification. Furthermore, the previous research showed that community members have real difficulty of making annotations of their objects using subject taxonomies [11]. On the other hand, they are more comfortable using their own application domain space as well as with their local context than multiple ontologies used in remote repositories.

In order to address this problem here we propose the use of ontology mappings to define relations between concepts from different ontologies [9, 16, 20]. On top of such mapping relations we developed a search algorithm that uses concept of one ontology (i.e. the source ontology) as query arguments, generates queries compliant with another ontologies (i.e. target ontologies), and finally gets ranked search results semantically relevant for the source ontology. To define mapping relations among ontologies we use another ontology – mapping ontology – that specifies a set of relations for relating concepts from different ontologies. Actually, we use the Mapping Vocabulary [17] of the Simple Knowledge Organization System (SKOS), a recent W3C RDF-based initiative [18]. We implemented the search algorithm using Jess [10] and OWLJessKB [15] as a component. The component can be used in different semantic web application such as a federated search engine of object repositories/digital libraries annotated with different classifications; or applications that allow using a local web application content ontology to get relevant results from remote digital libraries based upon another ontologies.

2. METADATA, ONTOLOGIES, AND WEB RESOURCES
Although the present semantic web research try to improve most of interoperability issues, some problems still exist.

2.1 Web metadata and domain ontologies
Web resource metadata and domain ontologies (i.e. taxonomies) are often defined at different ontological levels. In order to underpin this statement let us consider an example of combining Dublin Core (DC) metadata schema [4] and domain ontologies. Technically, the DC metadata of a web resource is an instance (i.e. RDF) of the DC RDF Schema. Additionally, the metadata is enriched with keywords defined in an domain ontology (e.g. for computer science domain based on the ACM Computing Classification System – CCS [2]). If we refer to keywords

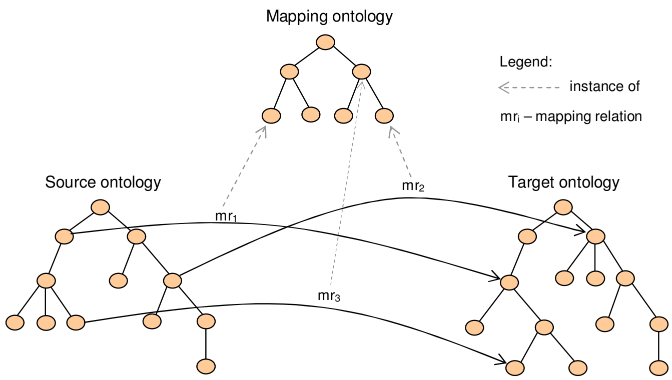


Figure 1. A general purpose mapping ontology as a way to define mappings among multiple ontologies

that are defined as classes in an RDF Schema, we annotate the metadata (i.e. schema instances) with ontology classes (i.e. schema). Those keywords are listed in the *subject* element of the DC metadata schema. Since ontology languages do not have a strict separation between ontological levels [6] this approach is completely applicable. In fact, this problem of representing classes as properties values has already been recognized by W3C Semantic Web Best Practices Working Group as classes as property values [19]. While OWL Full and RDF Schema do not put any restriction on using classes as property values, OWL DL and OWL Lite do not generally allow this use, and thus restrict the use of some Semantic Web reasoners. Apart from the solutions listed in the W3C note, we can use specialized ontologies for defining domain taxonomies with a rich set of properties for defining concept hierarchies such as SKOS [18].

2.2 Mapping among multiple domain ontologies

Currently, there are many different domain ontologies developed for the use on the Web. Very often developers are not able to reuse existing ontologies, as they were built for different purposes. For instance, some sources (e.g. object repositories, digital libraries) where we look for some web resources are based on different classifications (e.g. the ACM CCS in the ACM Digital Library [3]). We often need to build application-specific ontologies. For example, in the e-learning domain we can build an ontology of a course curriculum (e.g Information Management course) [22] to organize web resources related to the course. However, the main issue is how to use an application-specific ontology to search for web resources annotated with another ontology. In order to overcome such diversities we have to introduce an additional level of interoperability among ontologies [13]. One solution is to employ ontology mappings to define how concepts from different ontologies relate each other.

Here we describe only one way for defining mappings, although there are many practically used ontology mapping techniques [13]. It regards the use of a mapping ontology – an ontology containing classes and properties (i.e. primitives) that can express relations between ontology concepts and properties (see Figure 1). This principle is suitable for implementation since semantic web reasoning tools (e.g. FaCT, OWLJessKB) represent the mapping ontology in the same way (i.e. like facts) as both the source and target ontologies. Historically, this approach originates from the explicit representation of relationships between domain (i.e. source) and method (i.e. target) ontologies assembled in a specific knowledge application [21]. An example of such a mapping ontology was developed as a part of the project on reusable problem-solving components [9]. MAFRA (MApping FRAmework) is another solution for mapping distributed ontologies [16]. Apart from a very detailed mapping ontology called the semantic bridge ontology, MAFRA also defines two-dimensional process (5 horizontal and 4 vertical modules) ontology mappings process. Note also that a mapping ontology is used in the PROMPT Tab, a plug-in of the Protégé ontology editor for merging and mapping ontologies, to save discovered mappings [20]. However, none of these mapping ontologies is standardized. Furthermore, they do not posses a wide range of primitives for defining different levels of mappings (e.g. exact match), which can be useful in raking search results.

3. REPRESENTATION OF ONTOLOGIES AND MAPPINGS

In order to address two problems listed in previous section our solution uses the Simple Knowledge Organization System (SKOS) [18] for defining different types of ontologies (e.g. classifications, taxonomies, thesaurus) as well as mappings of concepts between different domain ontologies. The SKOS consists of the three RDF vocabularies that are still under the active development at the W3C:

- *SKOS Core* – for expressing the basic structure and content of concept schemes (taxonomies, terminologies, etc);
- *SKOS Mapping* – for describing mappings between concept schemes;
- *SKOS Extension* – containing extensions to the SKOS Core useful for specialized applications.

```
<rdf:RDF>
    <skos:ConceptScheme rdf:ID="&acm-ccs;acm-ccs">
        <skos:hasTopConcept rdf:resource="&acm-ccs;A" />
        <!-- ... -->
        <skos:hasTopConcept rdf:resource="&acm-ccs;K" />
    </skos:ConceptScheme>
    <!-- ... -->
    <skos:Concept rdf:ID="&acm-ccs;H.3">
        <skos:prefLabel xml:lang="en">Information Storage and
        Retrieval</skos:prefLabel>
        <skos:inScheme rdf:resource="&acm-ccs;acm-ccs" />
        <skos:broaderGeneric rdf:resource="&acm-ccs;H" />
        <skos:narrowerGeneric rdf:resource="&acm-ccs;H.3.1" />
        <!-- ... -->
        <skos:narrowerGeneric rdf:resource="&acm-ccs;H.3.m" />
    </skos:Concept>
    <!-- ... -->
    <skos:Concept rdf:ID="&acm-ccs;H.3.3">
        <skos:prefLabel xml:lang="en">Information Search and
    Retrieval</skos:prefLabel>
        <skos:inScheme rdf:resource="acm-ccs" />
        <skos:broaderGeneric rdf:resource="&acm-ccs;H.3" />
        <skos:narrowerGeneric rdf:resource="&acm-ccs;H.3.3.1" />
        <!-- ... -->
        <skos:narrowerGeneric rdf:resource="&acm-ccs;H.3.3.6" />
    </skos:Concept>
    <!-- ... -->
    <skos:Concept rdf:ID="&acm-ccs;H.3.3.1">
        <skos:prefLabel xml:lang="en">Information
            Filtering</skos:prefLabel>
        <skos:inScheme rdf:resource="&acm-ccs;acm-ccs" />
        <skos:broaderGeneric rdf:resource="&acm-ccs;H.3.3" />
    </skos:Concept
    <!-- ... -->
</rdf:RDF>
```

a)

```
<!--IM1.1 - History and motivation for information systems
        ->H.5.m - Miscellaneous -->
<skos:Concept rdf:about="&imc;IM1.1">
    <map:minorMatch>
        <skos:Concept rdf:about= "&acm-ccs;H.5.m"/>
    </map:minorMatch>
</skos:Concept>

<!-- IM1.2 - Information storage and retrieval (IS&R) ->
    H.3 - Information storage and retrieval-->
<skos:Concept rdf:about="&imc;IM1.2">
    <map:exactMatch>
        <skos:Concept rdf:about= "&acm-ccs;H.3"/>
    </map:exactMatch>
</skos:Concept>

<!--IM1.6 - Search, retrieval, linking, navigation ->
    Union of H.3.3 - Information Search and Retrieval
     E.2.3 - Linked representations -->
<skos:Concept rdf:about="&imc;IM1.6">
    <map:majorMatch>
        <map:OR about="#OR1"/>
    </map:majorMatch>
</skos:Concept>
<map:OR ID="OR1">
    <map:memberList rdf:parseType="Collection">
        <skos:Concept rdf:about="&acm-ccs;H.3.3"/>
        <skos:Concept rdf:about="&acm-ccs;E.2.3"/>
    </map:memberList>
</map:OR>
```

b)

Figure 2. The use of SKOS: a) An excerpt of the ACM CCS in the XML/RDF format of the SKOS. The classification comprises the 11 top level concepts marked with letters from A to K. Most of the top level concepts are further subdivided into three more levels with numbers being added to their identifiers; b) an excerpt of the mappings between the ACM CCS and another ontology (an Information Management course curriculum ontology) defined in the SKOS Mappings

SKOS Core provides a model for expressing the basic structure and content of concept schemes. Basically, the SKOS Core defines a set of both RDFS properties and RDFS classes that can be used to express the content and structure of a concept scheme (such as *Concept*, *broader*, *narrower*, *related*, *subject*, *isSubjectOf*). For example, the *broader* property is used to specify that a concept is broader than another one. In order to define subclass/superclass relations we can use the *SKOS Extension* vocabulary and its properties *narrowerGeneric* and *broaderGeneric* that are sub-properties of the *narrower* and *broader* properties, respectively. The *narrowerGeneric* property is semantically equivalent to the *rdfs:subClassOf* property, and thus has a slight different meaning from the *narrower* property.

3.1 Ontology representation

We use ACM CCS to illustrate the use of SKOS to define domain ontology. The ACM CCS is probably the most comprehensive classification in the domain of computer science [2]. An excerpt of the classification in the SKOS is shown in Figure 2a. Note that we use the *SKOS Extension* properties *broaderGeneric* and *narrowerGeneric* in order to have subclass/superclass relations between concepts.

3.2 Ontology mapping

The *SKOS Mapping* vocabulary contains a set of properties for specifying mapping relations among concepts from different domain ontologies (*broadMatch*, *narrowMatch*, *exactMatch*, *majorMatch*, *minorMatch*). Such a rich set of semantic relations for expressing mapping is useful in ranking search results to reflect the weight of the mapping. Apart from the properties, the SKOS Mapping has the three classes for defining: intersection of concepts (the AND class), union of concepts (OR), and negation (NOT).

In Figure 2b we show how we have used the SKOS Mapping to express the mapping between an e-learning relevant course curriculum ontology and the ACM CCS ontology. The curriculum ontology captures the *Information Management* course [1]. The course contains 14 units (top level SKOS concepts), and each unit contain several topics (sub-concepts of top level concepts in SKOS). One can see different match levels between concepts (i.e. *minorMatch*, *majorMatch*, and *exactMatch*) in Figure 2b. We also show how one defines the mapping relation between a concept (e.g. *IM1.6*) and a union (e.g. *OR1*) of other concepts (e.g. *H.3.3* and *E.2.3*). As mappings relations are not symmetric [17] we have to provide two mapping relations for each pair of concepts in case of two-way mappings.

4. ONTOLOGY MAPPING BASED SEARCH ALGORITHM

The substance of having mappings among different ontology-based vocabularies is to enable the use of the ontology A to search web resources annotated with concepts from another ontology B. Accordingly, we dedicate this section to the search algorithm we developed.

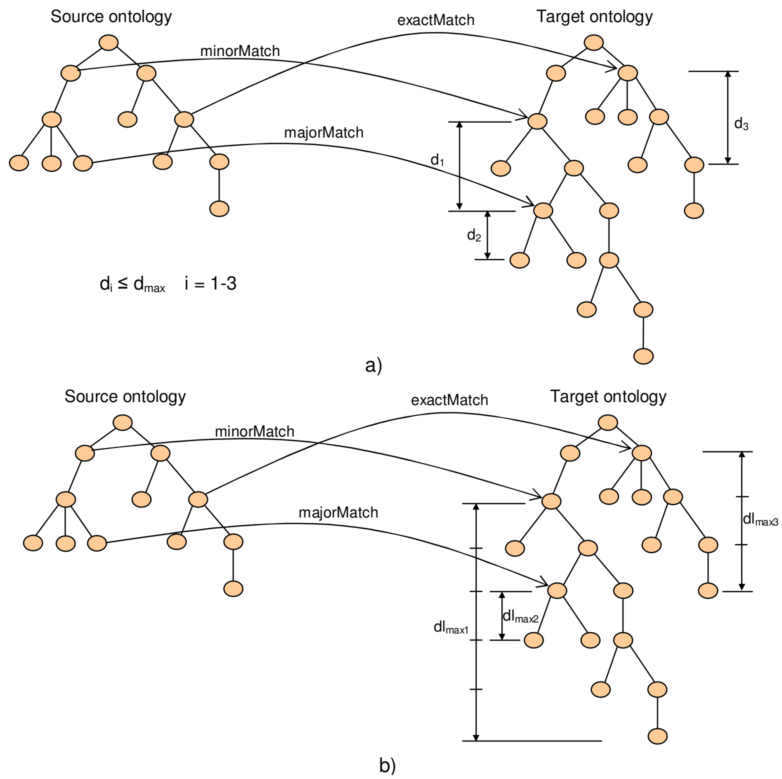


Figure 3. The search algorithm based on ontology mappings: a) only those child nodes d_{max} levels below the matched node are used; b) all child nodes of the matched node in the target ontology are used

4.1 Starting presumptions

The algorithm is based on the following presumptions:

- Input arguments of the search algorithm are concepts of the source ontology;
- Results of the search algorithm are concepts of the target ontology;
- Mapping relations among concepts from both source and target ontologies are defined using the SKOS Mappings;
- For each input argument the search algorithm looks for target ontology concepts that have defined mappings. We call those target ontology concepts – matched concepts.
- The search algorithm also looks for child concepts of matched concepts.
- When ranking search results, different kinds of the SKOS Mappings relations should be taken into account.

4.2 Initial algorithm

The input argument of the initial algorithm is a concept from the source ontology. The algorithm searches for matched concepts in the target ontology based on all types of SKOS mappings relation types. Next, the algorithm looks for child concepts of the matched concepts, but only a predefined number of levels (d_{max}) below the matched concept in the target ontology (see Figure 3a).

The algorithm creates 5 different lists of matched concepts called *clusters* (one for each mapping relation type) as well as 5 clusters of child concepts (d_{max} levels below) of the matched concepts. Finally, the algorithm merges all clusters respecting the order of clusters listed in the *cluster-names* variable in Figure 4 (NB Figure 4 does not illustrate this algorithm version, but the next one). In fact, the merging is performed by connecting clusters using the union operator.

Although the algorithm in a rather simple way searches for the matched concepts in the target ontology as well as ranks the resulting set of matched concepts, it still has some open issues: the resulting concept list is completely discrete structure due to the simple merging; the ranking procedure treats all children of the matched concepts within the same cluster in the same way, so concepts within a cluster are randomly ordered; and searching for child concepts a predefined number (d_{max}) levels below the matched node can take out of consideration some relevant child concepts.

4.3 Improved algorithm

First, the improved algorithm uses all the children of matched concepts in the target ontology regardless their depth level (see Figure 3b). Second, it uses the weight factor to determine ranks of both matched concepts and their children in the resulting list of concepts. The algorithm calculates the weight factor of a matched concept according to the type of the mapping relation connecting it with the source ontology concepts. The weight factor for each type of mapping relation is predefined (i.e. a constant number) and is subject to change depending on the tree structure of the target ontology (i.e. it can be fine tuned). Note also that referent weight factor is the *exactMatch* relation, while others (i.e. major, minor, and broad) are calculated relatively to it. That is the reason way that value is also an

```
function search-concept (input-concept, WFEM)
       cluster-names := {"exactMatch", "broadMatch", "exactMatchChildren",
              "broadMatchChildren" , "narrowMatch", "narrowMatchChildren" "majorMatch",
              "majorMatchChildren", "minorMatch", "minorMatchChildren"};
       clusters := create-hash-map();
       result := {};

       for-each name in cluster-names
              matched-concepts := get-matched-concepts(name, input-concept);
              clusters[name] := matched-concepts;
       end-for-each

       for-each name in cluster-names
              for-each concept in clusters[name]
                     put-in-sorted-list(result, concept, calculate-WF(concept, name));
              end-for-each
       end-for-each

       return result;

end-function
```

Figure 4. The search algorithm – considers all child concepts of the matched concept in the target ontology. It ranks the resulting list of concepts relying on the weight factor of the mapping relation type of the matched concept as well as the distance of child concepts from the matched (parent) concepts

input argument of the search algorithm (WF_{EM}). The weight factor for every matched concept child depends on (see Figure 3b): the maximal depth level of the matched (parent) concept; the distance of the child concept from its parent; the weight factor of its parent. Accordingly, the weight factor of the child concept is calculated using the following formula:

$$WF_{ch} = WF_p - (WF_p / (1 + dl_{max})) * d_{ch} \quad (1)$$

where:

- WF_{ch} – weight factor of the child concept;
- WF_p – weight factor of the matched (parent) concept;
- dl_{max} – maximal depth level of the matched (parent) concept;
- d_{ch} – distance of the child concept from the matched (parent) concept.

Relaying on the aforementioned facts we revise the algorithm. In Figure 4 we show a high level description of the search function (i.e. `search-concept`) in an informal pseudo-code. The first part of the algorithm is similar to the previous version. The difference is that the clusters are not merged like in the first algorithm, but they are stored to be members of a hash map – a memory structure keeping the track about all clusters. Once all clusters are created, the algorithm puts the concepts from each cluster in the resulting list (`result`) using the `put-in-sorted-list` procedure. Concepts in the resulting list are sorted according to their weight factors. Since the same concept can be in more then one cluster (e.g. `broadMatchChildren` and `majorMatchChildren`), the procedure prevents the repetition of the same concept in the resulting list by using its best weight factor.

Although this variant of the algorithm solves the most the problems we have mentioned for the first one, the algorithm still has some limitations that are referred in detail in the next subsection.

4.4 Final algorithm

The search algorithm presented in the previous section does not solve the case when mapping is not defined between the query argument and the target ontology (see Figure 5). Although the previous search algorithm variants look for children of matched concepts in the target ontology, it does not expand the query arguments that are parts of the source ontology. In fact, the solution works properly if mapping is defined between the query argument and one or more concepts from the target ontology. However, if there are no mappings defined for the query argument then the query will return an empty resulting list.

To overcome this issue, we additionally improved the search algorithm. The algorithm looks for both child and parent concepts in the source ontology that have defined mappings with concepts of the target ontology when the query argument has no defined mappings. In order to calculate weight factors of result concepts, the algorithm takes into account the fact that the distance between the

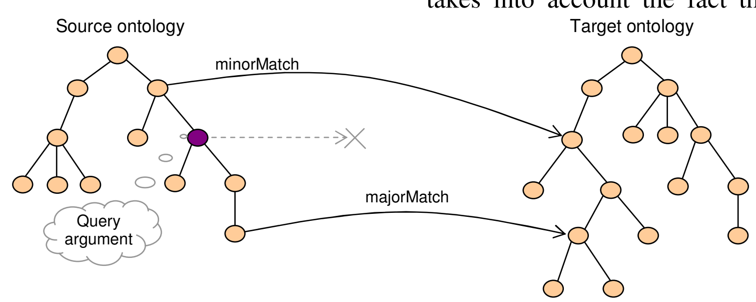


Figure 5. The case when mapping is not defined between the query argument and concepts of the target ontology

```
function search-concept-no-direct-match (input-concept, WFEM)
       result := search concept(input-concept, WFEM);
       if result == {} then
              children := get-subconcepts-with-mapping(input-concept);
              parents := get-superconcepts-with-mapping(input-concept);
              for-each c in children
                     WF := calculate-WF(c, input-concept);
                     put-in-ordered-list(result, search-concept(input-concept, WF));
              end-for-each
              for-each c in parents
                     WF := calculate-WF(c, input-concept);
                     put-in-ordered-list(result, search-concept(input-concept, WF));
              end-for-each
       end-if
       return result;
end-function
```

Figure 6. The final version of the search algorithm that captures the case exemplified in Figure 5 when none of mapping relations is defined among the query argument and concepts of the target ontology

query arguments and all its child and parent concepts with defined matching relations is not the same. Relaying on that fact we calculate the value of the weight factor (WF_i) for exact match (see the previous subsection) for each parent and child concept of the query argument in the source ontology using the following formula:

$$WF_i = WF_{EM} - abs(dl_{sc} - dl_i) * step \qquad (2)$$

where:

- WF_{EM} – weight factor of the exact match relation predefined for the case when there is a mapping relation between the query argument and the target ontology;
- dl_{sc} – depth level of the query argument;
- dl_i – depth level of a parent/child concept of the query argument that has a mapping relation with the target ontology;
- step – predefined value that specifies the impact of the distance between the query argument and its child/parent concept *i*.

Once we calculate the weight factor for the exact match relation, the weight factors of other mapping relations can be calculated as we have already explained in the previous subsection. In Figure 6 we show the final version of the search algorithm capturing the explanation given in this subsection and employing the search algorithm shown in Figure 4.

4.5 A Jess-based implementation of the proposed search algorithm

We implemented the algorithm using OWLJessKB [15], a Semantic Web reasoning tool, and Jess, a Java-implemented rule-based inference engine [10]. The use of the implemented algorithm regards invoking the corresponding Jess function whose input parameters are: a concept from the source ontology; and the weight factor for the exact match mapping relation. The function returns a ranked list of matched concepts as well as their child concepts.

5. EVALUATION

In order to evaluate the search algorithm we developed a web-based application in the domain of e-learning for an information management course. The course is based on the ACM/IEEE computer science curriculum recommendation [1]. The application has a typical organization – the left pane contains the course structure and the right pane holds the content of the one particular unit. In fact, the course structure is represented as a SKOS ontology. The bottom part of the right pane contains the context sensitive search for two different collections of web resources: the ACM Digital Library (DL) [3] and Merlot learning object repository (http://www.merlot.org). The ACM DL relays on the ACM CCS ontology while Merlot has its own classification. We encoded both classifications in SKOS. The students can search both collections of web resources by providing search keywords. However, the search action collects annotation of the current page in the course ontology (embedded in the web page in the RDF form) and applies the ontology mapping based algorithm to the annotation. Finally, the application sends an expanded query along with the keywords to chosen collection of web resources. The results received from collections of web are related to the current web page within the course.

Figure 7 contains the diagram comparing the search results we obtain when searching the ACM DL by using keywords and the combination of keywords and ACM CCS class identifiers. Note that the "full number" for each query means the overall number of objects that contain any of searched keywords in the keyword and ACM CSS classification fields of their metadata (see [3] for details). It is obvious that the combination of the keywords and ACM CCS class identifiers reduces the number of found objects, and hopefully helps find more relevant web resources. However, we noticed several peculiarities due to the use of a specific search engine (i.e. the ACM DL Advanced Search):

- *The number of results is decreased when increasing the number of classifiers in a query*. It was completely opposite to expectations as those query arguments are connected with the OR logical operator. The reason for such a behavior is that the ACM DL Advanced Search uses the Verity indexing engine (http://www.verity.com),

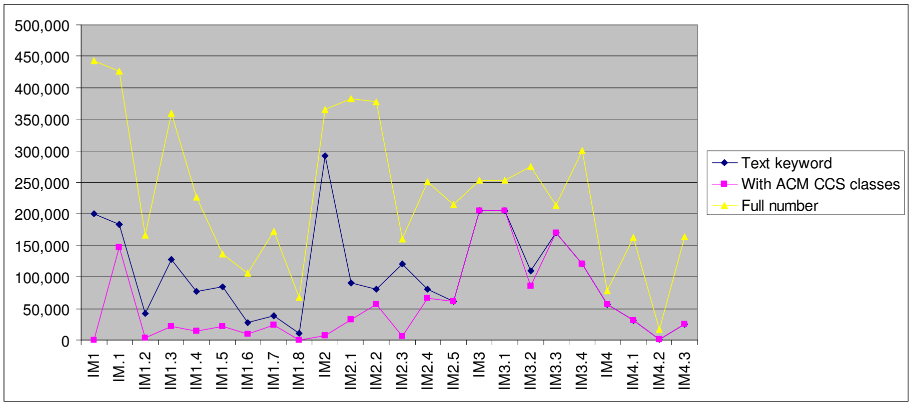


Figure 7. Comparison of the search results obtain from the ACM Digital Library by using text-based keywords and the combination of the text-based keywords and ACM CCS class identifiers

which selects only those objects whose weight factors pass over a specific threshold. Since their weight factors depend on the number of classification parameters, the less number of found objects can pass over the threshold

- *The last level of classification is omitted from the queries in the web application.* Due to a high number of classification concepts used in queries, the ACM DL Advanced Search can return an empty list of found concepts. The effect is especially stressed when using top level concepts as query arguments. The ACM DL search engine selects only those objects whose weight factors pass over a specific threshold. Increasing the number of classification parameters also increases the threshold and therefore eliminating some of the objects. However, this does not affect the best matching results.

Table 1 contains results obtained by applying the algorithm to search the Merlot learning object repository using the combination of the course ontology concepts and keywords. Unlike the ACM DL, in this case search results are in accordance with expectations, the greater number of classification tags in the query, the greater number of the found objects. Note also that the number of Merlot classification tags is not so high comparing to the experiment with ACM DL, since mapping relations between two ontologies are defined for the bottom level concepts of the Merlot classification. We found that most of concept from the course ontology did not have direct mapping relations with the target ontology, but they just inherited mapping relation from their parents.

Finally, say that we could not rank search results according to our ranking algorithm in either experiment, since we used two different digital libraries where we did not have any control in ranking of the found resources. In order to evaluate our ranking algorithm we are developing screen-scraping functions of both ACM DL and Merlot web pages showing search results.

6. CONCLUSIONS

The paper presented a way to achieve semantic interoperability when searching for web resources in web sources annotated by different domain ontologies. That way, users can get semantically relevant search results using classifications they are familiar with. The presented method exploited the idea of having one separated mapping ontology as it was already shown in [9, 16, 20]. Relations among different ontologies are not encoded into their structures, but they are represented separately. Accordingly, reusability of related ontologies is not decreased. Additionally, the evaluation examples showed benefit to have a combined keyword search with ontology annotated content to in order to provide more relevant web resources. Although we discussed the case of mapping between two ontologies, the described approach scales up to support

Table 1. Evaluation of the ontology mapping based algorithm to search Merlot learning object repository, which is based on its own classification, using a course curriculum ontology

Concept	IM1	IM.1	IM1.2	IM1.3	IM1.4	IM1.5	IM1.6	IM1.7	IM1.8	IM2	IM2.1	…
Keyword-based search	9814	10782	9760	2094	9769	9578	114	9760	9542	540	10797	…
Ontology-based search	55	59	85	22	53	80	1	9	52	25	35	…
Percent	0.56	0.55	0.87	1.05	0.54	0.84	0.88	0.09	0.54	4.63	0.32	…
Num. of classification tags	1	1	3	2	1	1	1	1	1	2	2	…
Defined match or not	Y	Y	Y	Y	Y	Y	N	Y	N	Y	N	…

Concept	…	IM2.2	IM2.3	IM2.4	IM2.5	IM3.2	IM3.3	IM3.4	IM4	IM4.1	IM4.2	IM4.3
Keyword-based	…	9449	1321	9638	12782	9614	418	1140	72	12788	9544	9563
Ontology-based	…	36	26	35	38	85	14	40	6	38	31	31
Percent	…	0.38	1.97	0.36	0.30	0.88	3.35	3.51	8.33	0.30	0.32	0.32
Num. of classification tags	…	2	2	2	2	3	3	3	2	2	2	2
Defined mapping or not	…	N	N	N	N	N	N	N	Y	N	N	N

multiple ontologies, provided we defined mapping between the source ontology and any number of target ontologies.

Comparing the search algorithm with other solutions we can find similarities with the Intelligent Product Information Search that employs ontology mapping to search for products using web services of several sellers based on different product ontologies [14]. However, this method has mappings defined in a table, while search procedure just considers direct mapping without consideration of child concepts. It further uses run-time discovery of mapping relations based upon lexical similarities defined in WordNet. Another similar approach tries to enable the use of user personalized ontologies to annotate web pages in order to compose web services [5]. The mapping rules between ontologies are defined in F-Logic. To the best of our knowledge, the approach just uses simple matching between related concepts from different concepts, without consideration of their child concepts.

In the future we plan to integrate the developed search algorithm into the eduSource Communication Layer (ECL) being developed in our research laboratory as a part of its federated search engine [12]. We also plan to research how we can automatically generate ontology mapping relations the search algorithm relays on. The idea is to apply the concept of semantic signatures as well as content of web resources to discover relation among ontology concepts [8].

7. ACKNOWLEDGMENTS

This work was funded by Canada's LORNET NSERC Research Network. The authors would especially like to thank Bernard Rous and Robert Okajima from the ACM Headquarter Office who provided details about the Advance Search engine of the ACM Digital Library.

8. REFERENCES

[1] Information Management Course Curriculum Recommandation (2001). http://www.computer.org/ education/cc2001/final/im.htm (2001)

[2] ACM Computing Classification System (CCS), http://www.acm.org /class/1998/ (2005)

[3] Association for Computing Machinery (ACM) Digital Library, http://www.acm/dl (2005)

[4] Dublin Core Metadata Schema, http://dublincore.org/documents/dcmi-terms/ (2005)

[5] Agarwal, S., Handschuh, S. and Staab, S. Surfing the Service Web, in Proc. of the 2nd Int'l Semantic Web Conf. (Sanibel Island, FL, USA, 2003), 211-226.

[6] Bechhofer, S., van Harmelen, F. and Hendler, J., et al. OWL Web Ontology Language Reference, http://www.w3.org/TR/owl-ref/ (2004)

[7] Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine, Computer Networks and ISDN Systems 30, 1-7 (1998), 107-117.

[8] Choi, A. and Hatala, M. Towards Browsing Diastant Metadata Using Semantic Signature, in Proc. of the KCAP2005 Workshop on Integrating Ontologies (Banff, Canada, 2005), (forthcoming).

[9] Crubézy, M. and Musen, M.A. Ontologies in Support of Problem Solving, *Handbook on Ontologies in Information Systems* (Berlin: Springer-Verlag, 2003), 321-341.

[10] Friedman-Hill, F. *Jess in Action* (Greenwich, USA: , Manning Publications Co., 2003).

[11] Greenberg, J. and Robertson, W.D. Semantic Web Construction: An Inquiry of Authors' Views on Collaborative Metadata Generation, in Proc. of the Int'l Conf. on Dublin Core and Metadata for e-Communities 2002 (Florence, Italy, 2002), 45-52.

[12] Hatala, M., Richards, G., Eap, T. and Willms, J. The Interoperability of Learning Object Repositories and Services: Standards, Implementations and Lessons Learned, in Proc. of the 13th World Wide Web Conf. (NY, USA, 2004), 19-27.

[13] Kalfoglou, Y. and Schorlemmer, M. Ontology mapping: the state of the art, The Knowledge Engineering Review 18, 1 (2003), 1-31.

[14] Kim, W., Choi, D.W. and Park, S. Product Information Meta-search Framework for Electronic Commerce Through Ontology Mapping, in Proc. of the 2nd European Semantic Web Conf. (Heraklion, Crete, Greece, 2005), 408-422.

[15] Kopena, J. and Regli, W.C. DAMLJessKB: A Tool for Reasoning with the Semantic Web, IEEE Intelligent Systems 18, 3 (2003), 74-77.

[16] Maedche, A., Motik, B., Silva, N. and Volz, R. MAFRA - A MApping FRAmework for Distributed Ontologies, in Proc. of the 13th Int'l Conf. on Know. Engineering and Know. Management, Ontologies and the Semantic Web (Siguenza, Spain, 2002), 235-250.

[17] Miles, A. and Brickley, D. SKOS Mapping Vocabulary Specification, W3C Document, http://www.w3.org/2004/02/skos/mapping/spec (2004)

[18] Miles, A. and Brickley, D. SKOS Core Vocabulary Specification, W3C Working Draft, http://www.w3.org/TR/swbp-skos-core-guide (2005)

[19] Noy, N.F. Representing Classes As Property Values on the Semantic Web, W3C Working Group Note, http://www.w3.org/TR/swbp-classes-as-values/ (2005)

[20] Noy, N.F. and Musen, M.A. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping, Int'l Journal of Human-Computer Studies 59, 6 (2003), 983-1024.

[21] Park, J.Y., Gennari, J.H. and Musen, M.A. Mappings for Reuse in Knowledge-based Systems, in Proc. of the 11th Workshop on Knowledge Acquisition for Knowledge-Based Systems (Banff, Canada, 1998)

[22] Rasseneur, D., Jacoboni, P. and Tchounikine, P. Enhancing a Web-based Distance-Learning Curriculum with Dedicated Tools, in Proc. of the IEEE/WIC Int'l Conf. on Web Intelligence (Halifax, Canada, 2003), 285-291

GMO: A Graph Matching for Ontologies

Wei Hu, Ningsheng Jian, Yuzhong Qu, Yanbing Wang

Department of Computer Science and Engineering
Southeast University
Nanjing 210096, P. R. China

{whu, nsjian, yzqu, ybwang}@seu.edu.cn

ABSTRACT
Ontology matching is an important task to achieve inter-operation between semantic web applications using different ontologies. Structural similarity plays a central role in ontology matching. However, the existing approaches rely heavily on lexical similarity, and they mix up lexical similarity with structural similarity. In this paper, we present a graph matching approach for ontologies, called GMO. It uses bipartite graphs to represent ontologies, and measures the structural similarity between graphs by a new measurement. Furthermore, GMO can take a set of matched pairs, which are typically previously found by other approaches, as external input in matching process. Our implementation and experimental results are given to demonstrate the effectiveness of the graph matching approach.

Categories and Subject Descriptors
D.2.12 [**Software**]: Interoperability; I.2.6 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Graph*; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*

General Terms
Algorithms, Experimentation, Measurement

Keywords
Semantic Web, Ontology Alignment, Graph Matching, Structure Similarity

1. INTRODUCTION
Web ontologies written by RDF Schema [7] or OWL [13] play a crucial role in the emerging Semantic Web, and ontology matching (or alignment) is necessary for establishing inter-operation between semantic web applications using different ontologies. Ontology matching can be seen as an operation that takes two graph-like structures and produces a mapping between elements of the two graphs that correspond semantically to each other. Due to the hardness of subgraph matching, ontology matching is a difficult issue. Some similarity-based approaches to ontology matching have been proposed in the literatures [3, 4, 8, 10, 12, 14]. As we know, structural similarity plays a central role in ontology matching. However, the existing approaches rely heavily on lexical similarity between labels of nodes and similarity of labels brought from thesaurus, e.g. WordNet [11]. And these approaches mixed lexical similarity with structural similarity.

In this paper, we present a new approach to ontology matching called GMO (Graph Matching for Ontologies). It uses bipartite graphs to represent ontologies, and measures the structural similarity between graphs by a new measurement. Usually, GMO takes a set of matched pairs, which are typically found previously by other approaches, as external input in the matching process, and output additional matching pairs by comparing the structural similarity. The input mapping given to GMO can be gained by variant approaches available, and may have big variance in size. So, our structural similarity is designed to be independent to lexical similarity, and the effectiveness of GMO is tested with variant sized input mapping. The rest of this paper is organized as follows: Ontology representation based on Bipartite Graph is presented in Section 2. A measure of structural similarity between a pair of web ontologies is proposed in Section 3. Our implementation is described in Section 4. Experimental results are reported in Section 5, and some comparison to related work is discussed in Section 6. Finally, Section 7 summarizes our work and outlines some of future work.

2. ONTOLOGY REPRESENTATION BASED ON BIPARTITE GRAPH
RDF model, a foundation of Semantic Web, has the nature of graph structure. OWL ontology can be mapped to an RDF Graph as stated in the fourth section of

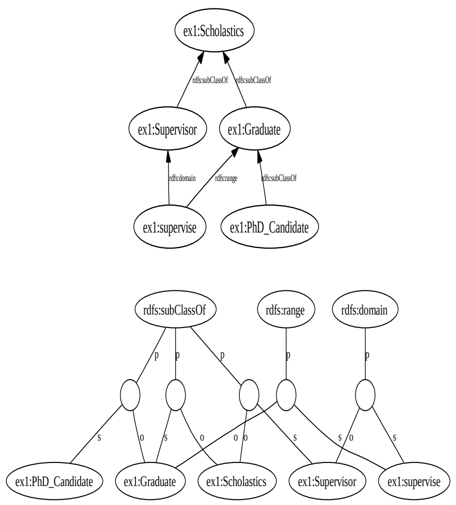


Figure 1: The RDF graph (upper) and RDF bipartite graph (lower) of $\mathcal{O}_{\mathcal{A}}$

OWL S&AS [13]. Thus, in order to compute structural similarity between ontology entities with semantic correspondence, we adopt the graph structure approach to represent web ontology in this paper.

2.1 RDF Graph vs RDF Bipartite Graph

According to the well-established methods from graph theory, we utilize the RDF Bipartite Graph model [6] to represent web ontology instead of the RDF Graph model. In this paper, we use $\mathcal{O}_{\mathcal{A}}$, $\mathcal{O}_{\mathcal{B}}$, ... to denote web ontology. As an example, Fig. 1 shows the RDF graph and RDF bipartite graph of ontology $\mathcal{O}_{\mathcal{A}}$.

2.2 Matrix Representation of Ontology

Definition 1. Let $\tilde{G}_A$ be the RDF bipartite graph of $\mathcal{O}_{\mathcal{A}}$. The directed bipartite graph of ontology $\mathcal{O}_{\mathcal{A}}$, denoted by G_A, is a derivation of $\tilde{G}_A$ by replacing the "s" edges with edges pointing to statement nodes, and the "p" and "o" edges with edges pointing from statement nodes. The adjacency matrix of G_A is called the matrix representation of ontology $\mathcal{O}_{\mathcal{A}}$, denoted by A.

Note: The matrix has the following block structure,

$$A = \begin{pmatrix} 0 & 0 & A_{ES} \\ 0 & 0 & A_S \\ A_E & A_{OP} & 0 \end{pmatrix}, \qquad (MR)$$

where A_{ES} is a matrix representing the connections from external entities to statements; A_S is a matrix representing the connections from ontology entities within $\mathcal{O}_{\mathcal{A}}$ (internal entities) to statements; A_E is a matrix representing the connections from statements to external entities of $\mathcal{O}_{\mathcal{A}}$; A_{OP} is a matrix representing the connections from statements to internal entities.

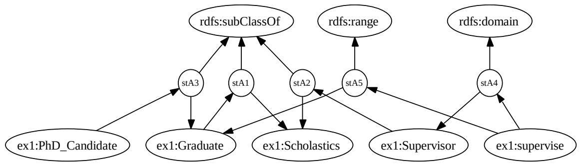


Figure 2: The directed bipartite graph of ontology $\mathcal{O}_{\mathcal{A}}$

The external entities for an ontology are usually those vocabulary defined by RDFS or OWL, built-in data types, and data literals. In some cases, the external entities may include some common ones (e.g. FOAF terms) used in two ontologies to be compared, and the ones specified in an input mapping. Therefore, the separation of "external" and "internal" is relative to the current matching process. In addition, when external entities of $\mathcal{O}_{\mathcal{A}}$ are not used as subject in $\mathcal{O}_{\mathcal{A}}$ (as in Fig. 2), A_{ES} is a zero matrix.

As an example, Fig. 2 shows the directed bipartite graph of ontology $\mathcal{O}_{\mathcal{A}}$. The matrix representation of ontology $\mathcal{O}_{\mathcal{A}}$ is as follows:

$$\begin{array}{r} \text{rdfs : subClassOf} \\ \text{rdfs : domain} \\ \text{rdfs : range} \\ \text{ex1 : supervise} \\ \text{ex1 : Scholastics} \\ \text{ex1 : Graduate} \\ \text{ex1 : Supervisor} \\ \text{ex1 : PhD\_Candidate} \\ \text{stA1} \\ \text{stA2} \\ \text{stA3} \\ \text{stA4} \\ \text{stA5} \end{array} \left[\begin{array}{ccc|ccccc|ccccc} & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ & & & & & & & & & & & & \\ \hline & & & & & & & & 0 & 0 & 0 & 1 & 1 \\ & & & & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & 1 & 0 & 0 & 0 & 0 \\ & & & & & & & & 0 & 1 & 0 & 0 & 0 \\ & & & & & & & & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & & & & & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & & & & & \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & & & & & \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & & & & & \end{array}\right].$$

3. STRUCTURAL SIMILARITY MEASURE FOR ONTOLOGY

We propose in this section a measure of structural similarity for ontology. With this measure, the structural similarity matrix is illustrated by an example. Then, the updating equations for structural similarity matrix are refined according to entity classification. Finally, a process for similarity computing is outlined.

3.1 A Measure of Structural Similarity for Ontology

Here, we conduct the structural comparison between entities in the two directed bipartite graphs for given ontologies. The idea of our measure is as follows. Similarity of two entities from two ontologies comes from the accumulation of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples, while the similarity of two statements comes from the accumulation of similarities of involved entities (including external entities) of the same role in the two statements being compared. Our formulation of structural similarity (3.1) and (3.2) is based on the similarity measurement between directed graph vertices proposed in the literature

[1].

The formulation in [1] (the equation (1.2) on page 650), uses the following updating equations for similarity matrix:

$$X_{k+1} = BX_kA^T + B^TX_kA, \qquad k = 0,\ 1,\ \ldots \qquad (3.0)$$

where X_k is the $n_B \times n_A$ matrix of entries x_{ij} at iteration k, and A and B are the adjacency matrices of G_A and G_B respectively. It is proved in the literature [1] that the normalized even and odd iterations of this updating equation converge, and that the limit $Z_{even}(1)$ is among all possible limits the only one with largest 1-norm. This limit is taken as the similarity matrix. By making use of the mentioned work, we define our similarity formulation for ontology as follows.

Definition 2. (A measure of structural similarity for ontology) Let A and B be the matrix representation of ontologies $\mathcal{O_A}$ and $\mathcal{O_B}$ respectively. Let O_k represent the similarity matrix of ontology entities within B to ontology entities within A at iteration k, S_k represent the similarity matrix of statements within B to statements within A at iteration k, and E_{BA} mean the similarity matrix of the external entities of B to the external entities of A. Suppose A, B and X_k (the structural similarity matrix of B to A at iteration k) has the following block form respectively .

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & A_{ES} \\ 0 & 0 & A_S \\ A_E & A_{OP} & 0 \end{pmatrix}, \\ B &= \begin{pmatrix} 0 & 0 & B_{ES} \\ 0 & 0 & B_S \\ B_E & B_{OP} & 0 \end{pmatrix}, \\ X_k &= \begin{pmatrix} E_{BA} & & \\ & O_k & \\ & & S_k \end{pmatrix} \end{aligned}$$

The updating equations for structural similarity matrix are defined as follows:

$$\begin{aligned} O_{k+1} &= B_SS_kA_S^T + B_{OP}^TS_kA_{OP} & (3.1) \\ S_{k+1} &= B_EE_{BA}A_E^T + B_{ES}^TE_{BA}A_{ES} & \\ &\quad +B_{OP}O_kA_{OP}^T + B_S^TO_kA_S & (3.2) \end{aligned}$$

If the limits of normalized even of iterations with $O_0 = \mathbf{1}$ and $S_0 = \mathbf{1}$ (we denote by $\mathbf{1}$ the vector or matrix whose entries are all equal to 1) of this updating equations exist, we take the limit of O_k as the structural similarity matrix of ontologies $\mathcal{O_B}$ to $\mathcal{O_A}$.

Our formulation of structure similarity, (3.1) and (3.2), differ from the one in [1] in three aspects: (i) We use directed bipartite graph instead of directed graph, (ii) Nodes are classified in different categories, (iii) The similarities between external entities are kept unchanged during updating.

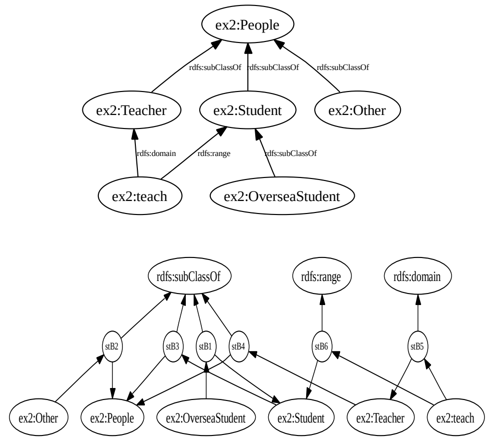


Figure 3: The RDF graph (upper) and directed bipartite graph (lower) of ontology $\mathcal{O_B}$

3.2 Structural Similarity Matrix by Example

Let $\mathcal{O_A}$ be the ontology described in section 2, $\mathcal{O_B}$ be the ontology depicted in Fig. 3.

The similarity matrix E_{BA} between external entities used in $\mathcal{O_B}$ and $\mathcal{O_A}$ is set in advance as

$$\begin{array}{r} \text{rdfs : subClassOf} \\ \text{rdfs : domain} \\ \text{rdfs : range} \end{array} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

By using the updating equations (3.1) and (3.2), we get the structural similarity matrix of $\mathcal{O_B}$ to $\mathcal{O_A}$ (after 12 iterations), as follows

$$\begin{array}{l} \\ \text{ex2 : teach} \\ \text{ex2 : Other} \\ \text{ex2 : People} \\ \text{ex2 : Student} \\ \text{ex2 : OsStudent} \\ \text{ex2 : Teacher} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{supervise} & \text{Supervisor} & \text{Graduate} & \text{Scholastics} & \text{PhD\_Candidate} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.132 & 0.132 & 0 & 0.040 \\ 0 & 0.001 & 0.220 & 1 & 0 \\ 0 & 0.579 & 0.884 & 0.025 & 0.040 \\ 0 & 0.007 & 0.007 & 0 & 0.107 \\ 0 & 0.502 & 0.579 & 9.05E-5 & 0.040 \end{bmatrix} \end{array}.$$

3.3 Refinement of Structural Similarity

For most cases, we can classify the entities described in a given ontology as properties, classes, instances (individuals and data literals). In fact, this kind of classification is guaranteed to be success for OWL DL and FOL subset of RDFS.

After successful classification, we can refine the matrix representation form (MR) of ontology $\mathcal{O_A}$, in section 2, as follows:

$$A_{ES} = \begin{pmatrix} A_{EPS} \\ A_{ECS} \\ A_{EIS} \end{pmatrix}, \quad A_S = \begin{pmatrix} A_{PS} \\ A_{CS} \\ A_{IS} \end{pmatrix},$$
$$A_E = (A_{EP},\ A_{EC},\ A_{EI}),$$
$$A_{OP} = (A_{POP},\ A_{COP},\ A_{IOP}).$$

where A_{EPS}, A_{ECS} and A_{EIS} represent the connections from external properties, classes and individuals to statements, respectively; A_{PS}, A_{CS} and A_{IS} represent the connections from internal properties, classes and individuals to statements, respectively; A_{EP}, A_{EC} and A_{EI} represent the connections from statements to external properties, classes and instances (including data literals); A_{POP}, A_{COP} and A_{IOP} represent the connections from statements to internal properties, classes and instances, respectively. As shown in above, we can also make the refinement of ontology $\mathcal{O}_{\mathcal{B}}$:

$$B_{ES} = \begin{pmatrix} B_{EPS} \\ B_{ECS} \\ B_{EIS} \end{pmatrix}, \quad B_S = \begin{pmatrix} B_{PS} \\ B_{CS} \\ B_{IS} \end{pmatrix},$$
$$B_E = (B_{EP},\ B_{EC},\ B_{EI}),$$
$$B_{OP} = (B_{POP},\ B_{COP},\ B_{IOP}).$$

The similarity matrix of external entities and the structure similarity matrix of ontologies have the diagonal structure as follows:

$$E_{BA} = \begin{pmatrix} EP_{BA} & & \\ & EC_{BA} & \\ & & EI_{BA} \end{pmatrix},$$
$$O_k = \begin{pmatrix} P_k & & \\ & C_k & \\ & & I_k \end{pmatrix}.$$

where EP_{BA}, EC_{BA} and EI_{BA} represent the similarity matrices of external properties, classes and individuals, respectively; P_k, C_k and I_k represent the similarity matrices of inner properties, classes and individuals, respectively.

The updating equations for structural similarity matrix are refined as follows:

$$P_{k+1} = B_{PS} S_k A_{PS}^T + B_{POP}^T S_k A_{POP} \quad (3.3)$$
$$C_{k+1} = B_{CS} S_k A_{CS}^T + B_{COP}^T S_k A_{COP} \quad (3.4)$$
$$I_{k+1} = B_{IS} S_k A_{IS}^T + B_{IOP}^T S_k A_{IOP} \quad (3.5)$$
$$\begin{aligned} S_{k+1} = {} & B_{EPS}^T EP_{BA} A_{EPS} + B_{ECS}^T EC_{BA} A_{ECS} \\ & + B_{EIS}^T EI_{BA} A_{EIS} + B_{EP} EP_{BA} A_{EP}^T \\ & + B_{EC} EC_{BA} A_{EC}^T + B_{EI} EI_{BA} A_{EI}^T \\ & + B_{POP} P_k A_{POP}^T + B_{COP} C_k A_{COP}^T \\ & + B_{IOP} I_k A_{IOP}^T + B_{PS}^T P_k A_{PS} \\ & + B_{CS}^T C_k A_{CS} + B_{IS}^T I_k A_{IS}. \end{aligned} \quad (3.6)$$

Note: The refined formulation of structure similarity has two advantages: (1) good computing performance due to the matrix computation with blocks; (2) avoiding the unnecessary computing of similarity between different kinds of entities, e.g. the ones between classes and properties.

4. IMPLEMENTATION

The graph matching for ontologies is implemented as an important component of our tool for aligning ontology, which is called Falcon-AO. In Falcon-AO v0.3, the input mapping to GMO is a set of matched pairs generated by another component, called LMO (A Linguistic Matching for Ontologies). In this section, the implementation of GMO is explained, followed by a brief introduction of LMO.

4.1 Matching Process of GMO

Suppose the ontologies to be matched are denoted by $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$. Given an initial mapping as input, by applying GMO, additional matching pairs will be generated. The implemented process of GMO is outlined as follows.

1. Parse $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$, and transform them to corresponding RDF bipartite graphs.
2. Classify entities (including anonymous ones) in $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ as classes, properties and instances.
3. Coordinate $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ using coordination rules described in 4.2.
4. Determine external entities for $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$ and setup external similarity matrix. In our implementation, the external entities are made of two parts: one includes built-in vocabularies of web ontology language, datatypes, data literals and URIs used in both $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$, and their similarity is preassigned; the other is identified by the input mapping.
5. Setup matrix representation for $\mathcal{O}_{\mathcal{A}}$ and $\mathcal{O}_{\mathcal{B}}$.
6. Initialize the similarity matrices P_k, C_k, I_k, S_k with **1** (we denote **1** the matrix whose entries are all equal to 1, with corresponding rows and columns suitable to the context).
7. Run the even steps of iterations with updating equations (3.3)-(3.6) till some pre-defined convergence precision is reached.
8. Find a one-one mapping by means of the similarity matrices P_k, C_k *and* I_k.
9. Output additional matching pairs.

In the current implementation, the iteration times of updating structural similarity matrix is set to 12, which is based on our primary experience. The finding of one-to-one mapping is finished when an estimated low similarity is reached.

4.2 Coordinating Ontologies with GMO

Ontologies to be matched may be represented differently, due to the heterogeneous ways in expressing semantics and the inference capability brought from ontology languages. Therefore, it is necessary to coordinate the two ontologies before mapping them.

Here, we outline several coordination rules, which are implemented in GMO. These rules can be classified into four categories presented as follows:

- Discarding: Some statements (triples) within an ontology may become redundant and/or worthless for computing structural similarity. For example, some typing statements such as (ex:A rdf:type owl:Class) become redundant after we successfully classify entities, and ontology header is worthless to structural comparing. Some rules are designed in GMO to discard such kinds of statements.
- Merging: Two entities could be stated to be same or equivalent to each other, e.g. (ex:A owl:equival entClass ex:B), then these entities should be merged in the RDF bipartite graph. There are some coordination rules to deal with this issue.
- Inference: In some situations, adding some inferred triples to the RDF bipartite graph with some inference rules would be helpful to structural comparing. For example, if there exist two triples, (ex:p owl:inverseOf ex:q) and (ex:q rdfs:domain ex:A), then, we could add one triple, (ex:p rdfs:range ex:A), if there is no triple to state the range of ex:p.
- List: To avoid heterogeneous in expressing a list using rdf:List, a List rule is presented. All members of a list are collected, and we use rdfs:member property to express the relation between the list and each of its members, instead of using RDF collection vocabularies(rdf:first, rdf:rest and rdf:nil).

More coordination rules will be introduced in later version of GMO. It is also worthy of note that there is a tradeoff between the cost of inference and the quality of mapping.

4.3 LMO – A Linguistic Matching for Ontologies

As is presented above, our GMO can be fed by an input mapping. In Falcon-AO v0.3, the input mapping to GMO is generated by LMO (A Linguistic Matching for Ontologies).

LMO includes two parts, one is based on string comparison, and the other is based on VSM (Vector Space Model). For string comparison, we use edit distance approach to calculate similarities between entities. For VSM, we treat the ontology entities (classes, properties and instances) as virtual documents. These virtual documents are constructed as "bags of terms" by using entity names, labels and comments, as well as neighbors' names or labels. Then, we can use VSM to gain the similarity matrix between entities. The details of LMO are out of the scope of this paper.

LMO brings some effectiveness to Falcon-AO, as demonstrated by the experimental results shown in section 5.2.

5. EXPERIMENTAL RESULTS

We have so far performed the GMO approach on OAEI 2005 benchmark test suite [1] and used standard information retrieval metrics to assess the results of our tests:

$$\begin{aligned}
\text{Precision} &= \frac{\sharp\text{correct_found_alignments}}{\sharp\text{found_alignments}}, \\
\text{Recall} &= \frac{\sharp\text{correct_found_alignments}}{\sharp\text{existing_alignments}}, \\
\text{F}-\text{Measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\end{aligned}$$

5.1 Effect of GMO

We tested the effectiveness of GMO on OAEI 2005 benchmark test cases, by taking some percentage of standard matched pairs as input mapping to GMO. The experimental results are demonstrated in Fig.4 by average precisions and recalls of all the test cases.

As shown below, with input matched pairs being fed increasingly, the GMO can find more additional correct matching pairs. The average precisions and recalls of test case #101-304 are indicated in y-axis, and the percentages of matched pairs as input mapping are shown in x-axis. It is worth noting that even with no input mapping, GMO still performs well, and the overall average precision and recall are 0.62 and 0.59 respectively.

We have categorized all the test cases into four groups: test case #101-104, #201-210, #221-266 and #301-304. Their average F-Measure are shown in Fig.5.

The results of test case #101-104 and #201-210 demonstrate that GMO is more suitable for those ontologies with similar structure than others. For these two categories of test cases, GMO still performs very nice without input mapping.

The weakness of GMO is also explicit. It performs not so well when the ontologies to be matched have a great

[1] http://oaei.inrialpes.fr/2005/benchmarks/

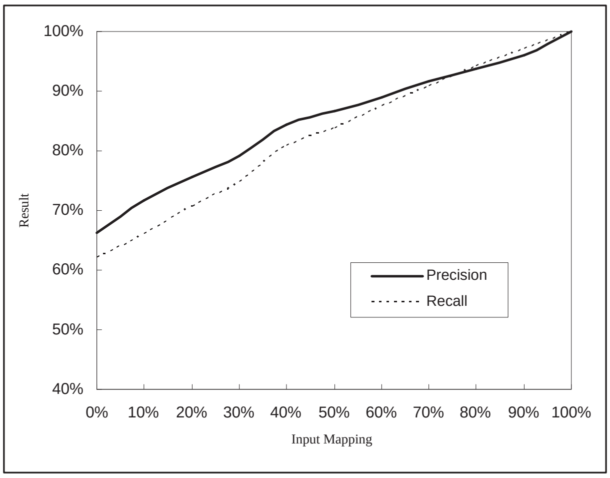


Figure 4: Average precision and recall

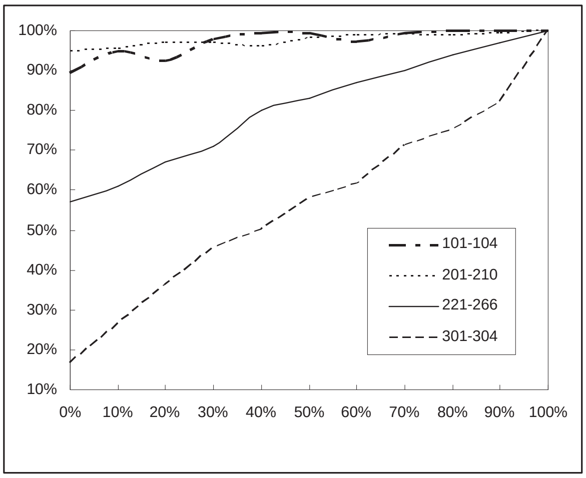


Figure 5: Average F-Measure of four categories

difference in structure (e.g. test case #221-266 and #301-304). In such cases, more matched pairs should provided as input.

5.2 Performance of Falcon-AO

The partial experiment results of our Falcon-AO are presented in Table 1, and you will see that Falcon-AO performs well for all these test cases.

The matched pairs generated by LMO are fed into GMO as input. In this step, we must make a decision between precision and effect, that is to say, the precision of input matched pairs should be as high as possible, and as is shown above, the amount will also affect the matching effect. The details of the decision will be presented in our experimental paper accompanied.

As can be seen from Table 1, our tool Falcon-AO works very well for test case #101-104 and test case #201-210, and performs pretty good for other two categories of test cases.

Table 1: The average performance on OAEI 2005 benchmark test suite

	101-104	201-210	221-266	301-304	Total
Prec.	1.0	0.96	0.86	0.93	0.89
Reca.	1.0	0.95	0.82	0.81	0.85
F-M.	1.0	0.95	0.83	0.86	0.87

6. RELATED WORK

Our presented work falls into the scope of similarity-based approaches to ontology matching. Logic based approach, e.g. Semantic Matching [5], and some others are beyond the scope of this paper. Here we present the closed-related work on similarity-based approaches. Among them, QOM [3] has a distinguished feature in efficiency with an emphasis on the alignment of RDFS ontologies. Anchor-PROMPT (included in PROMPT [12]) can produce new concept mapping by analyzing similar paths between a set of anchor matches, which are identified earlier (manually or automatically). OLA [4] and ASCO [8] are dedicated to the alignment of OWL ontologies (with an emphasis on OWL-Lite), and try to use as much as possible all of the information extracted from two given ontologies. In the literature [14], semantic-neighborhood matching is combined with word matching for class comparison. SF [10] is based on the idea that elements of two distinct models are similar when their adjacent elements are similar. The principle of our approach is similar to the basic idea of SF, but with very different measurement. In general, with these approaches and some others [9], entity features are setup based on labeled graphs or RDF graphs, and entity similarity is computed by counting feature-matches based on Tversky's contrast model [16], and then entity mapping is established based on (aggregated) similarities comparison and some specific heuristics rules (or user's interaction). Usually, those approaches mixed up lexical similarity and structural similarity, and/or heavily rely on lexical similarity to proceed with structural comparison.

Compared with them, our presented GMO approach uses bipartite graphs to represent web ontologies instead of using labeled graph or RDF graph, and measures the structural similarity between graphs by a new measurement. Our similarity model emphasizes the structural similarity based on the connection similarity, and does not depend on or mix up with lexical similarity. In addition, GMO approach can make use of a set of matched pairs found previously by other approaches. In fact, our similarity model also makes use of connections to "external" entities as well as matches between "external" entities identified earlier automatically or manually. This idea is similar to Anchor-PROMPT, but the method of similarity computing is very different. Furthermore, our work is targeted to web ontologies, including RDFS and OWL. Currently, with an emphasis on FOL subset of RDFS and OWL Lite.

The experimental results reported on EON Ontology Alignment Contest [15] also show that those reported ontology alignment tools rely heavily on lexical similarity between labels of nodes (node identifier, rdfs:label, and rdfs:comment). For the five tests with test number 201, 202, 204, 205 and 206, where property and class names were disturbed, the average f-measure of these tools is 0.61. When there is very little similarity found from lexical analysis, some tools will fail to proceed with structural comparison effectively. Our experimental results, as in Fig.5, show that GMO works very well for test #201-210 with average F-Measure more than 0.95, though some improvement is needed to enhance the overall effectiveness of GMO.

7. CONCLUSION

The GMO approach (a Graph Matching for Ontologies) presented in this paper has two distinguished features from early works as follows:

(i) It uses bipartite graphs to represent ontologies instead of using labeled graph or RDF graph. The bipartite graph model can reveal the real structure of web ontologies to be compared.

(ii) A new measure of structural similarity for web ontology. This measure will play an important role in ontology matching, especially when lexical similarity could not be gained.

Our GMO approach has been implemented in our ontology matchers. The experimental results demonstrated the feasibility and the effectiveness of GMO. As illustrated in Section 5, GMO is irreplaceable when there is little gain from lexical comparison. In addition, GMO-based matcher can be integrated with other matchers. Therefore, GMO is also a complement to other related work in the area of ontology matching.

As we pointed out in Section 3, ontologies should be coordinated before comparison due to the heterogeneous ways in expressing semantics and the inference capability brought from ontology languages. However, it is not easy to select appropriate coordination rules due to the tradeoff between the cost of inference and the quality of mapping. Another issue is the interaction between similarities at conceptual layer and instance layer. In current stage of our implementation, we separate these two layers, and use matches at conceptual layer as inputs to compute similarities at instance layer. In the case of comparing instance-intensive ontologies, machine learning (e.g. GLUE [2]) is a promising approach to make use of instance information in aligning classes or properties.

As part of future research, we are going to improve the GMO approach and related algorithms in some aspects, e.g. coordination issue and layers issue. We plan to integrate GMO with techniques in machine learning and natural language processing to realize more powerful ontology matchers.

Acknowledgments

This work is jointly supported by National Key Basic Research and Development Program (2003CB317004) and JSNSF (BK2003001). We would like to thank Prof. Jianming Deng and Dr. Zhiqiang Gao for their suggestions. We are also grateful to other members in our team for their contributions. In the end, we would like to thank anonymous reviewers for their helpful suggestions. Further information about our tools and the experimental results can be found at [2].

8. REFERENCES

[1] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. *SIAM Review, 46(4) (2004) 647-666*

[2] A. Doan, J. Madhavan, P. Domingos, A. Halevy. Learning to Map between Ontologies on the Semantic Web. *The Eleventh International World Wide Web Conference (WWW'2002), Hawaii, USA (2002)*

[3] M. Ehrig, S. Staab. QOM-Quick Ontology Mapping. *In Proc. of the 3rd International Semantic Web Conference (ISWC2004), November, 2004, Hiroshima, Japan, LNCS 3298, Springer (2004)*

[4] J. Euzenat, P. Valtchev. Similarity-based Ontology Alignment in OWL-Lite. *Proc. of ECAI, 333-337, 2004*

[5] F. Giunchiglia, P. Shvaiko. Semantic Matching. *The Knowledge Engineering Review Journal 18(3) (2004) 265-280*

[6] J. Hayes, C. Gutiérrez. Bipartite Graphs as Intermediate Model for RDF. *International Semantic Web Conference (2004) 47-61*

[7] P. Hayes (ed.). RDF Semantics. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/rdf-mt/*

[2] http://xobjects.seu.edu.cn/project/falcon/falcon.html

[8] B. T. Le, R. Dieng-Kuntz, F. Gandon. On Ontology Matching Problems for Building a Corporate Semantic Web in a Multi-communities Organization. *ICEIS (4) (2004) 236-243*

[9] A. Maedche, S. Staab. Measuring Similarity between Ontologies. *EKAW (2002) 251-263*

[10] S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. *In Proc. of ICDE (2002) 117-128*

[11] A. G. Miller. Wordnet: A Lexical Database for English. *Communications of the ACM 38(11) (1995) 39-41*

[12] N. F. Noy, M. A. Musen. The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. *International Journal of Human-Computer Studies 59 (2003) 983-1024*

[13] P. F. Patel-Schneider, P. Hayes, I. Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/*

[14] M. A. Rodriguez, M. J. Egenhofer. Determining Semantic Similarity among Entity Classes from Different Ontologies. *IEEE Trans. Knowl. Data Eng. 15(2) (2003) 442-456*

[15] Y. Sure, O. Corcho, J. Euzenat, T. Hughes (eds.). Proceedings of the 3rd International Workshop on Evaluation of Ontology Based Tools (EON2004). *Vol. 128, CEUR-WS Publication. November (2004)*

[16] A. Tversky. Features of Similarity. *Psychological Review, 84(4) (1977) 327-352*

Towards Semantic Based Information Exchange and Integration Standards: the art-E-fact ontology as an extension to the CIDOC CRM (ISO/CD 21127) Standard

Carlos Lamsfus
VICOMTech
San Sebastián, Spain, E-20009
clamsfus@vicomtech.es

María Teresa Linaza
VICOMTech
San Sebastián, Spain, E-20009
mtlinaza@vicomtech.es

Tim Smithers
VICOMTech
San Sebastián, Spain, E-20009
tsmithers@vicomtech.es

ABSTRACT

Ontologies have been established as effective and efficient means of knowledge sharing and are being widely used to conceptually model domains of knowledge. With the growing use of ontologies in various domains of interest, the problem of overlapping knowledge in a common domain becomes critical. In this context, much work has already been done developing semi-automated applications that enable the merging, mapping or alignment of ontologies.

On the other hand, a big effort is being currently developed by many communities (e.g. eLearning, telemedicine, cultural heritage) in order to standardize their contents and data models facilitating the integration and exchange of content coming from heterogeneous data sources.

This paper presents the work that is currently being carried out in two different directions: first, to align two existing ontologies (the art-E-fact project domain ontology and the CRM ontology); second, to pursue a culture of re-using existing ontologies and content, generalizing the alignment method so that it becomes standard and applicable in other domains so that existing knowledge can be re-used and shared easily.

Keywords

Ontology, alignment, cultural heritage, standards, re-use.

INTRODUCTION

A unified representation for Web data and resources is needed in today's large scale Internet data management systems. This unification of standards will allow machines to meaningfully process the available information and to exchange and integrate data coming from distributed databases and information management systems. This has been occurring, e.g. in the context of eLearning with the development of the SCORM (http://www.adl.net) and AICC (http://www.aicc.org) standards, or in the context of telemedicine applications with the development of standard data transport protocols such as HL7 and ISO/IEEE/CEN 11073, among others.

In the area of cultural heritage, there have also been some initiatives to enable distrubuted data exchange and integration. Interoperability between databases has to be provided on both technical and informational (semantic) levels. Problems that may arise due to heterogeneity of the data are already well-known within the distributed database systems community: structural heterogeneity and semantic heterogeneity.

Problems related to structural heterogeneity of distributed databases have already been solved by the database management systems community. Furthermore, in order to achieve semantic interoperability, i.e. achieve communication between two agents that work in overlapping domains, the meaning of the information that is being interchanged has to be understood across both systems. The use of ontologies for the description of implicit hidden knowledge is a possible approach to overcome the problem of semantic heterogeneity.

This paper presents the work that is currently being carried out in two important lines of research:

- First, research on information integration and exchange standards in the context of cultural heritage: suitability of alignment of the art-E-fact domain ontology into the CIDOC CRM ontology;
- Second, research on ontology merging and alignment, in order to align the art-E-fact ontology into the CIDOC CRM ontology.

This paper is organized as follows. Section 2 presents previous work upon which this article is based briefly describing the CIDOC Conceptual Reference Model and the art-E-fact project and domain ontology. Section 3 is a comparative analysis of both the art-E-fact and CRM ontologies and summarizes the differences between them. Section 4 describes the working method that is currently under development. Finally, Section 5 gives some conclusions.

PREVIOUS WORK

The CIDOC Conceptual Reference Model (CRM)

The CIDOC CRM, a core ontology explaining the extended meaning of data structures from humanities and cultural heritage, including history of science, is the outcome of a long-term disciplined knowledge engineering activity, which excels in its ontological commitment, i.e. acceptance of its constructs by domain experts.

The primary role of the CRM is to enable information exchange and integration between heterogeneous sources of cultural heritage information [Doe03]. It aims at providing the semantic definitions and clarifications needed to transform disparate, localised information sources into a coherent global resource within a larger institution, in intranets or in the Internet. More concretely, it defines and it is restricted to the underlying semantics of database schema and document structures used in cultural heritage and museum documentation in terms of a formal ontology.

The following are some of the most important functionalities of the CRM:

- To serve as a common language for domain experts and IT developers to formulate requirements and to agree on system functionalities with respect to the correct handling of cultural contents;
- To support the implementation of automatic data transformation algorithms from local to global structures without loss of meaning. This is useful for data exchange or data information integration; as well as,
- To support associative queries against integrated resources by providing a global model of the basic classes and their associations to formulate such queries.

The success of the CIDOC CRM lies in the fact that the explanation of common meaning can be done by a very small set of primitive concepts and relationships, in contrast to the data structures that suggest to the user what to say about an object. The relations in data structures that connect items directly by highly specific, diverse kinds of relationship can frequently be expressed by data paths composed of a few fundamental relationships defined in the core ontology.

The CIDOC CRM has become the most promising core element for realizing semantic interoperability in Archives, Libraries and Museums, by its capability to link the intellectual structure of highly diverse sources and products of scientific and scholarly discourse with the elements formally handled by information systems. The CIDOC CRM is currently being elaborated by the International Standards Organization as Committee Draft ISO/DIS21127 and the CIDOC CRM Special Interest Group (SIG) to become an ISO standard.

The overall scope of the CIDOC CRM can be summarised in simple terms as the curated knowledge of museums [CDG03]. The Intended Scope of the CRM may be defined as all information required for the exchange and integration of heterogeneous scientific documentation of museum collections.

The art-E-fact project

The objective of the art-E-fact (IST-2001 37924) project was to create a generic platform for Interactive Storytelling in Mixed Reality that allowed artists to create stories in an original way within a cultural context between the virtual and the physical reality.

In other words, art-E-fact's general purpose was to make art accessible in a different way from traditional methods. For example, a piece of art could be introduced, explained and even discussed by two virtual characters in the form of a story. They set up a conversation first with each other and then with the visitor or participant, who will feel involved in the virtual world generated by the platform.

Furthermore, intuitive and easy to use Mixed Reality based interaction techniques enable the visitor to explore the art work in depth by using physical devices that will be recognized by a gesture recognition system. With art-E-fact, the visitors of the museum will deepen their understanding of the complex issues surrounding the history, techniques, and social circumstances behind the individual artworks.

So, if we want to make art accessible in the form of stories to visitors of museum exhibitions we have to provide artists, users and content generators in general with a tool (the art-E-fact Authoring Tool) that allows them to create this kind of art with the features we are describing.

Creation of art is the genesis of an original expression of feelings, thoughts, passions etc. Expression is an output of what creators obtain in their internal worlds, through their cultural background and environment, as well as through their technical skills. The huge amount of experiences and the stochastic way of assimilating and mixing them is the kernel of the final expressions that arise.

Therefore, if art-E-fact's target is "to tell" stories about (existing) artworks the author or content generator (using the art-E-fact Authoring-Tool) should be aware of this rich internal world which is provided to him/her through the art-E-fact description-metalevel ontology. The author's technical skills aided by the Authoring Tool, which retrieves the required information from the ArtWorks Database (AWDB) using the metalevel ontology, arise in an optimal way following the memory of creating art.

Description of the conception of the art-E-fact ontology

The art-E-fact domain ontology is composed of 84 classes and 173 properties and has been implemented in RDF Schema. It represents the artworks and its relational data stored in the AWDB and it is referred to five levels of knowledge, enriched with a set of metadata or descriptors of the data of the diagnosis. All these levels of knowledge or "thematic entities" in the ontology conception are supported by the scientific diagnosis results and the related documentation:

- The entity "Work identification" consists of general historical data, identifying aspects such as subject, title, category, type, dimensions, current location, context, ownership or creator of the artwork;
- The entity "Description" consists of information concerning the descriptive details of the theme and forms of representation, providing a better understanding of the context, such as representation, people, background, decorative elements, inscriptions or sceneries;
- The entity "Aesthetic appearance" is concerned mainly with plastic elements, which provide the appreciation of the style/aesthetic appearance of the artwork, such as the style, manner, composition set-up, colour, drawing style or texture;
- The entity "Technical" includes technical information both revealing the techniques and the materials used in the creation of the artwork, such as support, preparatory layers, underdrawings, painting materials, varnishes or stratigraphy, and also concerning exams of the condi-

tion, such as diagnosis or conservation treatments history;

- The entity "Interpretation" is provided compared or associated with analogous or totally unlike artworks, such as thematic relationships, persons, symbols, styles or techniques;

These main entities and their metadata are supported, documented and provided by the scientific diagnosis that has been applied to the artworks.

COMPARISON OF THE CRM AND art-E-fact ONTOLOGIES

The CIDOC CRM and the art-E-fact ontologies reflect a serious commitment to the expression of common concepts underlying the data structures used by their users. The art-E-fact model, driven by requirements of artists and content generators was motivated by the need to describe added-value content for the creation of stories, whereas the CIDOC CRM model, motivated by cultural artifacts, documentation experts and museum requirements, focuses on documentation processes among cultural institutions. These are some of the most relevant differences between the art-E-fact and the CRM ontologies:

- The intended scope of the CIDOC CRM has been defined as all the information required for the scientific documentation of cultural heritage collections, with a view to enabling wide area information exchange and integration of heterogeneous sources. The main objective of the art-E-fact ontology is not devoted to documentation, but to content description and comprehension.
- In the context of the CRM, the term cultural heritage collections is intended to cover all types of material collected and displayed by museums and related institutions, as defined by ICOM. This includes collections, sites and monuments relating to natural history, ethnography, archaeology, historic monuments, as well as collections of fine and applied arts. The art-E-fact ontology is also valid for interpretation centres and humanistic research institutions, which may have access to data and are not included among the ICOM concept.
- The scope of the CIDOC CRM is the curated knowledge of museums, while the scope of the art-E-fact project is the content generation by the artists.
- The CIDOC CRM is specifically intended to cover contextual information: the historical, geographical and theoretical background in which individual items are placed and which gives them much of their significance and value. On the other hand, the art-E-fact ontology takes into account different levels of knowledge in order to provide rich content to build interactive amazing stories.

Therefore, the main difference between both ontologies is the application domain. There is no incompatibility between both models. Moreover, we believe it should be possible to consider the art-E-fact ontology as an extension in the area of content description and generation of the CRM.

ALIGNMENT OF THE ART-E-FACT AND CRM ONTOLOGIES

As proved before the art-E-fact domain ontology is (conceptually) complementary to the CRM and could therefore be considered as an extension of the standard model. We are proposing to incorporate the art-E-fact domain ontology into the CRM as part of the standard to study and research different ontology alignment methods, developing a new one if necessary.

We have previously presented the different levels of knowledge or thematic entities the art-E-fact domain ontology covers as well as the scope of the CIDOC CRM. The CRM covers ("only") information about cultural heritage collections which would be equivalent to the first thematic entity (Work Identification) defined in the art-E-fact ontology. Therefore, we are going to link (unite) common concepts of both ontologies, allowing this way the CRM ontology to access to the rest of knowledge levels covered by art-E-fact but not contemplated in it. Figure 1 (conceptually) shows the work that we are currently carrying out.

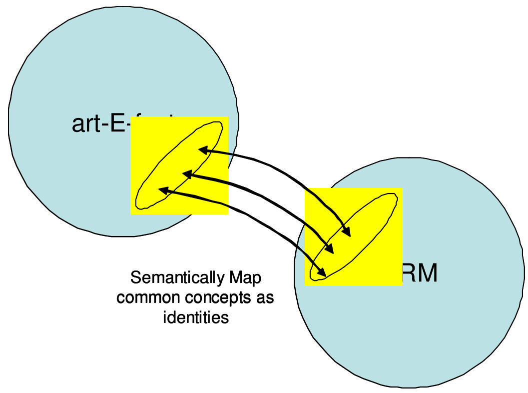


Figure 1: alignment of the art-E-fact and CRM ontologies

For the alignment of both ontologies we are going to use a rule-based methodology by the means of the emerging Semantic Web rule languages. Reasoning languages for the Web are an emerging technology that does not exist today. This technology will soon represent an essential breakthrough for Web systems and applications. One possible rule-based ontology language that we can use in this process is the Web Rule Language (WRL) for the Semantic Web. This language is located in the Semantic Web stack next to the Description Logic based Ontology Language (OWL).

The ontology vocabulary can be specified using WRL or OWL, or using their common semantic subset, denoted by the WRL-Core subset of WRL and the OWL-DP subset of OWL [Grosof *et al.*, 2003]. With common semantic subset we mean in this context that every WRL-Core has a corresponding OWL-DP ontology and vice versa, where both ontologies entail exactly the same set of ground facts.

So, the alignment work of both ontologies can roughly be summarized in the following tasks:

- Since the alignment of both ontologies is going to be carried out using the WRL language, we have to use the OWL DL version of the CRM and art-E-fact ontologies;
- The art-E-fact ontology was built using the RDF(s) language, therefore we have to represent this ontology using OWL DL;
- We have to standardize the art-E-fact ontology. Here, the word standardize stands for giving to the art-E-fact domain ontology's classes, slots etc. the same name they would have in the CRM ontology;
- Then, we have to identify the common concepts (classes) in both the art-E-fact and CRM domain ontologies and link them. To do so, we are going to follow a semantic rule-based process with WRL as rule language. This is the reason because we have to use the OWL (DL) version of both ontologies;
- In this linking process we have to be specially careful with the possible subclasses and slots each of the linked classes has, so that no information at all is lost in the alignment process;

We have now presented a very specific example of ontology alignment. In order for this process to be generic enough, we should also work on the standardization of the linking process so that it can also be used in the alignment of other ontologies.

Much previous work related to ontology mapping and merging has been done. To generate the bridging axioms, we must first find out the correspondence between the concepts of the two ontologies, which is the target of ontology mapping. Lots of systems have been implemented to map ontologies, e.g. CUPID, GLUE, Chimaera, PROMT and many others.

SUMMARY AND CONCLUSIONS

Technology, in the wider sense of its meaning, is tending to standards in order to enable and ease information exchange across different and usually distributed information management systems, should they be mobile devices or desktop computers. Thus, standards are not just a need but a must.

In this paper we have presented the CIDOC CRM and the art-E-fact domain ontologies. Then, we have compared them and, finally we have justified why we built a new ontology from scratch after analyzing their differences. Now, we are trying to align the art-E-fact ontology with the CRM standard in our commitment with standards and the re-use of previously developed work. In order to achieve this goal, we will study different methodologies, tools and ways of doing it and we will apply the most suitable one.

This is, however, a very concrete example of aligning ontologies. Therefore, another work that is under development is the standardization of the mapping process so that this alignment process can be carried out in a semi-automated way for other kinds of ontologies.

Ontology alignment, merging or mapping processes are very tedious and time consuming. There are various (semi-automated) tools that have already been used in some previous initiatives, however at some point of the alignment process there has to be some kind of manual intervention. We shall very carefully study their functionality and see whether they are finally suitable or if we should develop a new aligning tool.

Standards are becoming a very important issue within the new Information and Communication Technologies (ICT) context. With this research project, we would like to contribute to the extension of the most important information and data exchange format there is in the area of cultural heritage, i.e. the CIDOC CRM, giving the possibility to general cultural heritage institutions to exchange and integrate added value data related to artworks. Moreover, some research is going to be done in the alignment of ontologies and we expect to contribute with methodologies, e.g. standard mapping methodologies.

ACKNOWLEDGEMENTS

The art-E-fact project is funded by the Information Society Technologies Programme of the European Commission and conducted by a consortium of eight participating institutions and companies from five European countries. The author would like to thank the IST Programme for their financial support.

REFERENCES

[1] Advanced Distributed Learning, http//www.adlnet.org

[2] Aviation Industrie CBTI Committee, http://www.aicc.org

[3] Crofts, N.; Dörr M.; Gill T.; Stead S.; Stiff M.: *"Definition of the CIDOC Conceptual Reference Model"*. ICOM/CIDOC CRM Special Interest Group, 2003.

[4] Dörr M.; Hunter J.; Lagoze C.: "Towards a core ontology for information integration. Journal of Digital Information 4(1)" (2003).

[5] Dörr M.: *"The cidoc conceptual reference module: An ontological approach to semantic interoperability of metadata"*. AI Magazine 24(3) (2003), 75–92.

[6] Dou, D; McDermott; Qi, P, "Ontology translation by Ontology Merging and Automated Reasoning"

[7] Elvimaria Terzi, Athna Vakali, Mohand-Saïd Hacid, *"Knowledge Representaioin, ontologies and the Semantic Web"*

[8] Grosof, B. N.; Horrocks, I.; Volz, R.;, and Decker, S. *Description logic programs: Combining logic programs with description logic.* In Proc. of the Twelfth International World Wide Web Conference (WWW 2003), pages 48-57. ACM, 2003.

[9] Guarino, N; Welty, C. *"Evaluation Ontological decisions with OntoClean"*. Communications of the ACM, Vol 45, Nr. 2, pp. 61-66. February, 2002.

[10] Iurgel I.: *"From another point of view: art-E-fact"*. In Proc. TIDSE '04 (2004), vol. 1, pp. 26–35.

[11] Noy N.F.; Musen M.A., *"SMART: Automated Support to Ontology Merging and Alignment"*, Twelth Workshop on Knowledge Acquisition, Modeling, and Management, 1999. Banff, Canada.

[12] Noy N.F.; Klein M; *"Ontology evolution: not the same as Schema Evolution"*, July 2004

[13] REWERSE, Reasoning on the Web with Rules and Semantics, http://rewerse.net/

[14] The CIDOC Conceptual Reference Model (CRM), http://cidoc.ics.forth.gr

[15] The Web Rule Language, http://www.wsmo.org/wsml/wrl/wrl.html#wsml

An approach to ontology mapping negotiation

Nuno Silva
DEI & GECAD – ISEP - IPP
Porto, Portugal
Nuno.Silva@dei.isep.ipp.pt

Paulo Maio
DEI & GECAD – ISEP - IPP
Porto, Portugal
maiopaulo@sapo.pt

João Rocha
DEI & GECAD – ISEP - IPP
Porto, Portugal
Joao.Rocha@dei.isep.ipp.pt

ABSTRACT

Ontology mapping negotiation aims to achieve consensus among real-world entities about the process of transforming information between different models (ontologies). This paper describes a novel approach for ontology mapping negotiation, in which agents representing the real-world entities are able to achieve consensus among agents, about the mapping rules defined between two different ontologies. The proposed approach is based on utility functions that evaluate the confidence in a certain mapping rule. According to the confidence value, the mapping rule is accepted, rejected or negotiated. Since the negotiation process requires relaxation of the confidence value, a meta-utility function is applied, evaluating the effort made in relaxing (increasing) the confidence value, so that the mapping rule might be accepted. This convergence value is further applied by each agent in the evaluation of the global agreement.

Categories and Subject Descriptors

I.2.4 Knowledge Representation Formalisms and Methods – *representation languages, semantic networks.*

H.3.5 - Online Information Services - *Data sharing.*

Keywords

Ontology, ontology mapping, negotiation.

INTRODUCTION

The ontology mapping process aims to define a mapping between a source and target ontology ($\mathcal{M}: O_s \rightarrow O_t$). This mapping is composed of a set of semantic bridges (mapping rules) and their inter-relations. In our particular case, the mapping and its semantic bridges are defined and respect the SBO - Semantic Bridging Ontology [1]. Each semantic bridge describes the semantic relation between a set of entities (concepts or properties) of the source ontology and a set of entities of the target ontology. This description is further applied in transforming instances of the entities of the source ontology into instances of the entities of the target ontology. According to the required transformation, different transformation Services are applied in the semantic bridge. The discovery and specification of the semantic bridges are performed respectively by the Similarity Measuring and Semantic Bridging phases of the MAFRA – MApping FRAmework [1] (Figure 1).

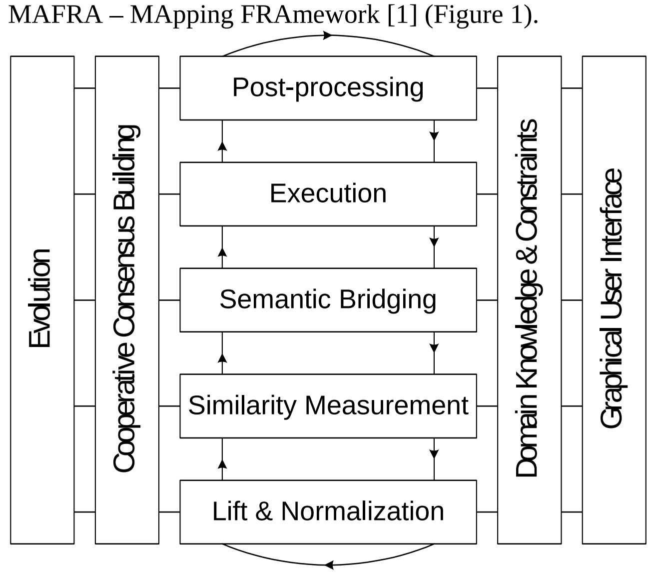


Figure 1. The MApping FRAmework

However, the semantic bridges resulting from these phases represent the perspective of an agent on the semantic relations defined between the entities of two ontologies. Due to the intrinsic subjective nature of the ontologies, different agents might have (and usually do have) different perspectives on the same mapping scenario. This leads to conflicts when interoperability occurs between such agents. A consensus building mechanism is required to overcome these conflicts. This mechanism corresponds to the Cooperative Consensus Building module of MAFRA [1].

The user-based process is naturally applied in offline semantic bridging scenarios, i.e. when the semantic bridging phase is carried out and proofed by human domain experts, prior to any data exchange phase. Yet, in scenarios where online semantic bridging is required, an automatic consensus building mechanism is necessary, in order to supply the necessary consensus and speed up the interoperability process. Applications in context of the semantic web, information retrieval, web services, e-commerce and e-business, are application scenarios where ontology mapping and online semantic bridging are highly recommended.

K-CAP'05, October 2-5, 2005, Banff, Canada.

This paper addresses the problem of the automatic consensus building among two agents about an ontology mapping. The proposed mechanism, named ontology mapping negotiation, is based on the relaxation of the agents' goals.

The rest of this paper runs as follows: the next section presents the state of the art on the subject and related fields. The third section defines and constraints the ontology mapping negotiation problem according to the characterization of other types of negotiation. The fourth section takes into account the general notion of negotiation and introduces the conceptual approach to the problem. The fifth section describes the so called service-oriented architecture, envisaged as potential approach to the problem, namely the semantic bridging competencies already developed. The sixth and seventh sections describe the proposed solution. Finally, the conclusions section gives an overview of the proposed solution and emphasizes the major contributions of the paper.

STATE OF THE ART

Basically there is no research on the topic of ontology mapping negotiation. Instead, long run research exists in the general topic of negotiation, but it is fundamentally concerned with electronic commerce and resource allocation, which is poorly related to this problem. Some ontology-based negotiation research is running [2;3], but this is related to the application of ontologies in the traditional research areas of resource allocation or e-commerce. Supporting this premise, it has be noticed that in two of the most specific and relevant research events in the subject, MCN'2004 (Meaning Coordination and Negotiation Workshop at ISWC-2004) and MeaN'2002 (Meaning Negotiation Workshop at AAAI-02), while many research papers on ontology coordination (mapping) have been presented, none has been presented about ontology negotiation.

While resource allocation and e-commerce research field may contribute to the negotiation of ontology mappings, no research exists about the specific characteristics of the negotiation of ontology mappings. In particular, it is necessary to determine and characterize the variables of the negotiation [4;5]:

- Number and type of the negotiation entities.
- Object of the negotiation (single/multi-object, uniqueness, granularity).
- Domain of the negotiation (single/multi-attribute).
- Characteristics and constraints of the negotiation process (visibility, honesty, mechanisms, information, strategy).

The definition and characterization of the negotiation context is the subject of the next section.

DEFINITION OF THE PROBLEM

Any negotiation process aims to achieve a consensus that, explicitly or implicitly, corresponds to a commonly agreed contract between two entities.

While the contract is the goal of the negotiation, its content is subject to change during the negotiation and, in the end, it might not be the best possible contract for any of the agents. I.e., the optimal contract, defined by each of the agents, might not be achieved. Besides, it is good enough and advantageous to both agents so that it can be accepted by them. However, the optimal contract is a function that might not be explicitly or implicit defined by any of the agents. This is normally the case in ontology mapping, especially due to:

- The differences between both ontologies.
- The subjective nature of both ontologies.
- The goal and requirements of the interoperability.

In the context of this project, the real-world agents are represented by artificial agents that act on behalf of the real-word agents during the negotiation. Considering that the real world agents (and therefore the artificial agents too) most probably have different perspectives on the ontology mapping scenario, one of the major questions is how to supply to the (artificial) agents the capability to converge on a consensus.

As in any negotiation process, the ontology mapping negotiation problem is mainly characterized by the type of object to negotiate. According to the developed semantic bridging phase [6;7], several types of objects might be considered:

- The mapping ($\mathcal{M}$), when the whole specification is subject of negotiation.
- The semantic bridges, when each of the semantic bridges composing the mapping are subject of negotiation.
- Parameters of the semantic bridges (e.g. the set of related entities).

However the more elements are subject of negotiation, the longer and more difficult it is to achieve a consensus among agents. Notice that a coarse grained negotiation (upon the mapping) is very fast, but a consensus is very hard to achieve, due to the lack of relaxation parameters. On the other hand, a fine grained negotiation (on the semantic bridges parameters) is easier to achieve, but it might be too long and therefore unfeasible.

Another important dimension to consider is the value associated to the object of negotiation. In the ontology mapping negotiation scenario, the value of the object is a function relating to the:

- Correctness of the object, either the correction of the mapping, of the semantic bridges or of their parameters.
- Pertinence of the object in respect to its envisaged application.

Other dimensions are also relevant for the negotiation process, but in order to reduce the negotiation space, the following constraints have been decided and stated:

- The negotiation always occurs between two honest, non-bluffing agents.
- The ontology mapping to agree on is unidirectional, which means that for a bi-directional conversation, two ontology mapping negotiation processes are required.
- The negotiation objects are the semantic bridges only. It means that no internal parameter of the semantic bridge is independently negotiable.

HYPOTHESIS

The proposed negotiation process bases on the idea that each entity is able to derive the correct semantic bridges and decide which semantic bridges are required in order to interoperate with the other entity.

The suggested approach aims to further exploit the multi-dimensional service-oriented architecture adopted in the semi-automatic semantic bridging process described and introduced in [6].

As referred previously, one of the major problems faced in negotiation scenarios relates to the difficulty in determining and supplying convergence mechanisms to the agents. In that respect, it is important to analyse the notion of negotiation.

Negotiation suggests the need for relaxation of the goals to be achieved by one (or both) of the intervenients in the negotiation, so that both achieve an acceptable contract, and an as good as possible one.

This introduces two distinct concepts:

- The goals of the negotiation (the features of the contract to achieve).
- The possibilities of relaxing the goals.

Mathematically, these concepts might be represented respectively as:

- A utility function (u), representing the overall goal of the negotiation of the semantic bridge, in which each parameter of the function is a sub-goal of the negotiation:

$$\mathrm{u}(\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_n)$$

- A meta-utility function (U) of the parameters of the utility function, defining the conditions in which the parameters may vary:

$$\mathrm{U}(\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_n)$$

Since the parameters of the utility function are the basic concept of this approach, it is fundamental to identify the possible elements that might play this role in the ontology mapping negotiation process.

It is our conviction that this role might be played by the same parameters that contribute to determine the correctness and completeness of the semantic bridges in the semantic bridging phase.

SERVICE-ORIENTED ARCHITECTURE

In scope of the semantic bridging phase, this role is played by the Matches, which are the outcome of the similarity measurement phase. Matches represent the confidence that specific and specialized algorithms, called Matchers (e.g. Resnik, H-Match, MOMIS), have concerning the semantic similarity of two entities (one from the source ontology and the other from the target ontology). A match corresponds therefore to the following tuple:

$match := (\mathcal{E}_s, \mathcal{E}_t, Matcher, Confidence) : \mathcal{E}_s \in O_s, \mathcal{E}_t \in O_t$ These matches are then grouped together by Services into semantic bridges. Since each semantic bridge associates one single Service that determines most of the characteristics of the semantic bridge, Services are perceived as the decision makers of the semantic bridging phase. Among others, Services define:

- The matchers whose matches are considered for evaluation of the Service confidence in the semantic bridge.
- The matches threshold (t_{match}), below which the matches are not considered.
- The confidence evaluation function (u) that evaluates the Service confidence in the semantic bridge (c_{sb}).
- The Service confidence threshold (t_r), below which the semantic bridge is rejected.

Table 1 represents the definition of these previous parameters for three Services.

Table 1. Some services parameterization

Service	Matches	t_{match}	u	t_r
CopyInstance	Resnik-like	0.7	u_{ci}	0,6
	H-Match	0.7		
CopyRelations	Resnik-like	0.5	u_{cr}	0,67
	H-Match	0.7		
CopyAttribute	Resnik-like	0.8	u_{ca}	0,85
	MOMIS-like	0.8		

In case the evaluate confidence value (c_{sb}) is above the respective threshold t_r, the semantic bridge is proposed to the user by the Service. Otherwise, the semantic bridge is rejected.

Extrapolating this approach to other phases of the ontology mapping process, the service-oriented architecture gives rise to the so called multi-dimensional service-oriented architecture [6] (Figure 2). In this architecture, Services provide specific functionalities to each phase of the process, thus contributing decisively to more tasks of the process than simply in transforming source instances into target instances. Services are then perceived as competent and decision makers in multiple phases of the process.

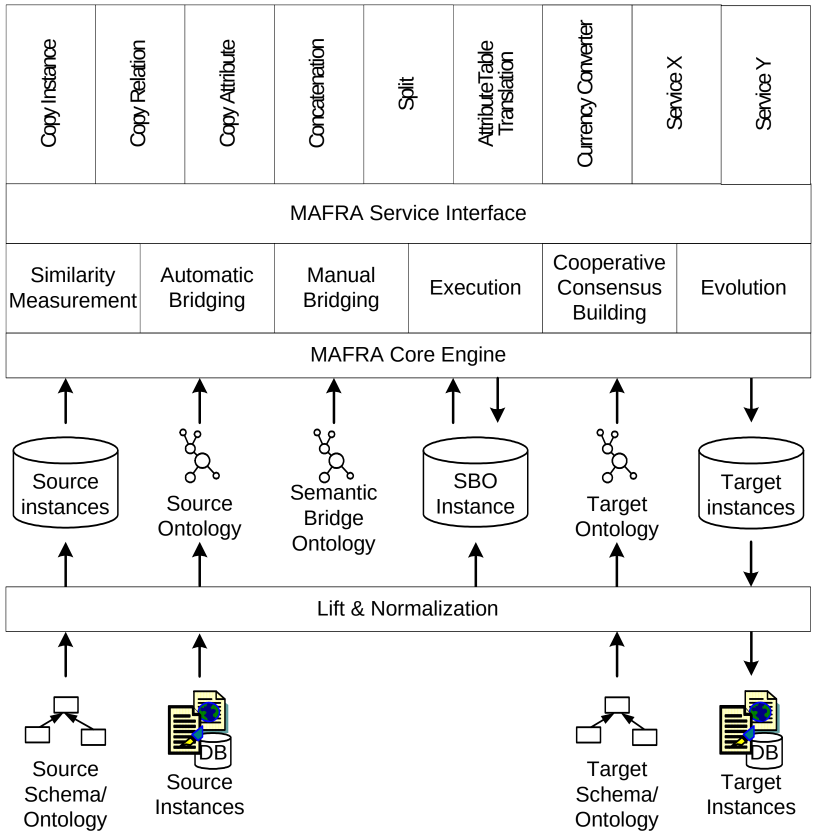


Figure 2. The Multi-Dimensional Service-Oriented Architecture of the MAFRA Toolkit

SERVICE-ORIENTED NEGOTIATION

The confidence evaluation function introduced above, generically referred to as utility function (u) plays a major role in the negotiation process. In fact, the proposed negotiation process suggests applying the confidence evaluation function as the utility function introduced in the hypothesis.

Reusing the utility function reduces the efforts of Services parameterization and customization, two very human demanding tasks. However, it is our proposal to distinguish the semantic bridging from the negotiation phase, i.e. both phases occur consecutively. First, each agent performs its own semantic bridging process, generating a valid and meaningful mapping. After that, the set of semantic bridges composing the mapping are subject to negotiation between both agents.

The confidence value evaluated for each semantic bridge (c_{sb}) is then used as the negotiation value of the semantic bridge, corresponding to the agent confidence in proposing the semantic bridge to the other agent.

Several situations might occur when negotiating a specific semantic bridge:

- Both agents propose the semantic bridge.
- Only one of the agents proposes the semantic bridge.

In case last situation occurs, one of two situations occurs:

- The other agent relaxes the confidence value and accepts the semantic bridge.
- The other agent cannot relax the confidence value and rejects the semantic bridge.

In case last situation occurs, one of two situations occurs:

- The agent proposing the semantic bridge cannot accept the rejection. In this case, the proposed semantic bridge is considered mandatory.
- The agent proposing the semantic bridge can accept the rejection.

Since the goal of the process is to negotiate, it is important to provide the mechanisms so that the agents are able to propose, reject and revise their perspective on the semantic bridges. In fact, throughout the negotiation, it is important that agents relax their sub-goals in favour of a larger and wider goal. In this sense, the agent should not decide *a piori* on the acceptance/rejection of the semantic bridge. Instead, it should admit that certain semantic bridges are neither accepted nor rejected: they are negotiable.

Consequently, it is necessary to define confidence categories, so that the agent can judge the semantic bridge pertinence to the mapping and to the interoperability. As a consequence, the rejection threshold borderline (t_r) is insufficient and should be replaced by a multi-threshold approach:

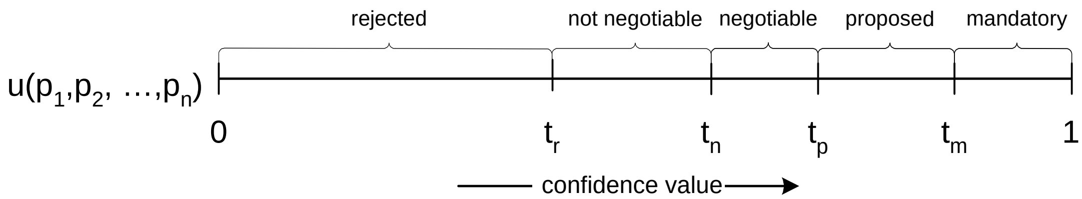


Figure 3. Categorization of semantic bridges according to the utility function and thresholds

- Mandatory threshold (t_m) that determines the utility function value above which it is fundamental that the semantic bridge is accepted by the other agent.
- Proposition threshold (t_p), above which the semantic bridge is proposed to the other agent.
- Negotiation threshold (t_n), above which the semantic bridge is negotiable.

Therefore, five distinct categories of semantic bridges are defined according to the confidence value and the previously identified thresholds (Figure 3):

- Rejected semantic bridges are those that $c_{sb} < t_r$. Rejected semantic bridges are not even proposed to the user.
- Non-negotiable semantic bridges are those that $t_r \le c_{sb} < t_n$. These semantic bridges are proposed to the user but unless he/she changes explicitly its category, they are not negotiated.
- Negotiable semantic bridges (SB^n) are those that $t_n \le c_{sb} < t_p$. It means that the agent confidence in the semantic bridge is sufficient to consider relaxing c_{sb}, but not enough to propose it to the other entity. In successful relaxing cases, the semantic bridge might be accepted.
- Proposed semantic bridges (SB^p) are those that $t_p \le c_{sb} < t_m$. It means the agent is confident enough upon the semantic bridge so that it proposes it to the other agent.
- Mandatory semantic bridges (SB^m) are those that $c_{sb} \ge t_m$. The agent is so confident of the pertinence and correctness of this semantic bridge, that the semantic bridge may not be rejected by the other agent.

It is therefore necessary to provide the mechanisms, so that the agent is able to revise its perception of the negotiable semantic bridges. These mechanisms should be embodied in the meta-utility function, as defined in the hypothesis, but not yet contemplated in the applied service-oriented approach of the semantic bridging phase.

The meta-utility function (U) is responsible for the definition of:

- The parameters variation possibilities.
- The priorities over parameters variation.
- The conditions under which the variation may take place.

Through these elements, an updated confidence value is evaluated (c^u_{sb}) for the negotiable semantic bridges that were proposed by the other agent. If $c^u_{sb} \ge t_a$, the negotiable semantic bridge is categorized as tentatively agreed (SB^t). Tentatively agreed semantic bridges are subject of a definitive decision phase.

Since the meta-utility function determines priorities and conditions for the variation of the parameters, it is possible that, for some variations, $c^u_{sb} < t_a$. It is therefore necessary to iterate across the different variation possibilities, following the defined priorities and conditions. In case it is impossible to evaluate $c^u_{sb} \ge t_a$, the semantic bridge is not re-categorized and is therefore rejected.

The effort made by the agent to re-categorize a semantic bridge to SB^t varies according to the priorities conditions and values of the parameters. The meta-utility function is also responsible for the evaluation of this effort, named convergence effort (e_{sb}). This convergence effort value is further applied in the definitive agreement phase, as described in the next section.

NEGOTIATION PROCESS

The negotiation process described in this section exploits the service-oriented elements introduced in previous sections. The main idea behind the proposed negotiation process is that each agent must maximize the number of proposed semantic bridges (SB^p) that are agreed to by the other agent.

The negotiation runs in two consecutive phases (Figure 4). The first one intends to build a consensus on mandatory semantic bridges (SB^m). The second intends to build a consensus on the proposed semantic bridges (SB^p).

In the first phase, each agent proposes every $sb^m \in SB^m$ to the other agent. If one sb^m is not accepted by the other agent, the negotiation is closed without a consensus.

In the second phase, each agent proposes every $sb^p \in SB^p$ (not yet negotiated) to the other agent. Three situations may occur:

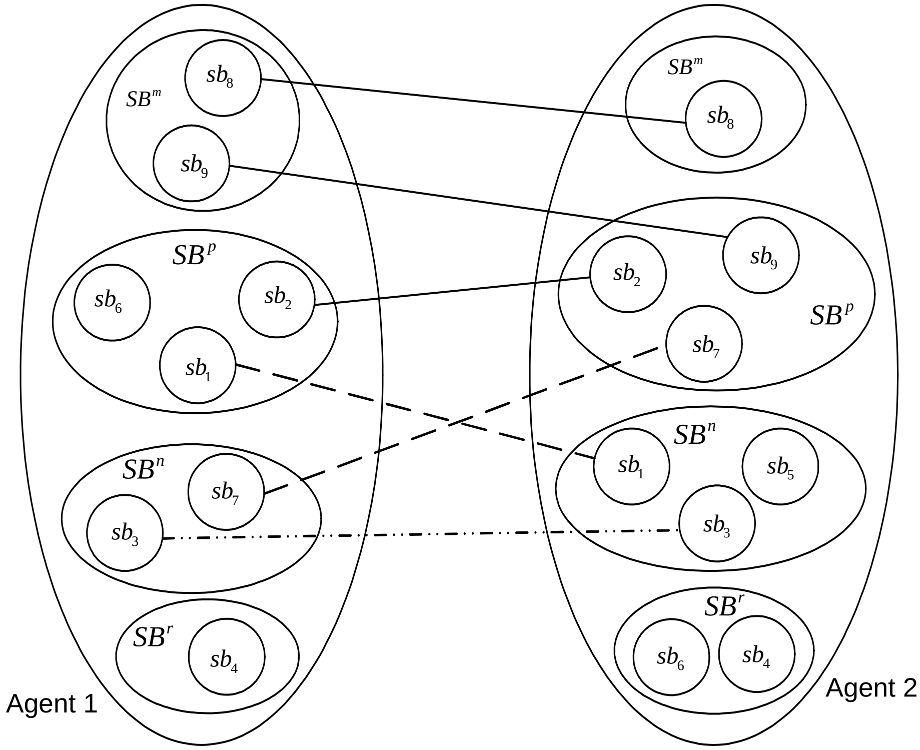


Figure 4. Semantic bridges negotiation process

1. The semantic bridge is also proposed by the other agent, thus categorized as agreed semantic bridge (SB^a). This situation is represented in Figure 4 by the sb_2 semantic bridge.
2. The semantic bridge is rejected by the other agent, and is therefore rejected (sb_6).
3. The semantic bridge is negotiable by the other agent, therefore categorized as tentatively agreed (SB^t). This is the case of sb_1 and sb_7 semantic bridges.

When both entities categorize certain semantic bridge as negotiable, it is suggested that they forward the decision on a potential agreement to the user (sb_3 semantic bridge).

The semantic bridges included in the third situation are subject to a definitive agreement phase in order to ensure that the proposed agreement is advantageous for both agents. The problem consists in deciding if the achieved agreement is globally advantageous (mapping granularity) and not only locally advantageous (semantic bridge granularity).

The problem arises due to the convergence efforts made during the negotiation process. For every $sb \in SB^n$ re-categorized as SB^t a convergence effort has been evaluated by the meta-utility function (e_{sb}). Convergence efforts should be considered inconvenient to the agent and treated as a loss. Instead, the agreement upon the same semantic bridge provided some profit for the agent when it is re-categorized. This profit is denoted by the confidence value (c_{sb}). In that sense, the balance between profits and losses is a function such:

$$balance = \sum c_{sb} - \sum e_{sb} : sb \in SB^t$$

Depending on the balance value the entity decides to agree on the negotiation agreement or to propose a revision of the mapping.

The balance value ultimately depends on the evaluation of the convergence effort made by the meta-utility function. In its simplest evaluation form, the convergence effort may correspond to the difference between c^u_{sb} and c_{sb} (i.e. $e_{sb} = c^u_{sb} - c_{sb}$).

However, the convergence effort should not be a linear measure between these two values. In fact, the linear difference between c^u_{sb} and c_{sb} it is typically too small in comparison to the values of c_{sb}. As a consequence, the balance value would be constantly positive.

A potential solution is the evaluation of the convergence effort using an exponential function defined under the parameters variation of the meta-utility function. Such exponential function would be helpful in taking into account the distinct efforts made in varying the different parameters in the meta-utility function. Instead, the difficulties in configuring and customizing the meta-utility function would be a considerable inconvenient.

CONCLUSIONS

The Multi-dimensional Service-Oriented Architecture advocates that ontology mapping system capabilities and its supported semantic relations are ultimately dependent on the type of transformations allowed/available in the system. Services represent the transformation capabilities in SBO, in semantic bridging and in the execution system, but the proposed architecture suggests that their capabilities should be expanded to support the requirements of other phases of the process. Services embody useful and eventually fundamental competencies for distinct phases of the process, which were originally an exclusive competence of the domain expert. Yet, instead of a monolithic structure representing such knowledge, multiple independent and dynamically evolving modules are used. However, these modules, instead of adopting a task-oriented structure, are orthogonal to multiple phases of the ontology mapping process, providing different functionalities depending on the requesting phase.

The service-oriented negotiation process introduced in this paper exploits such architecture. Services are empowered with competencies to negotiate the agreement on semantic bridges previously generated by the same Services. Services are able to revise their perspectives on the previously categorized semantic bridges, providing therefore the ability to relax their requirements in order to agree on a semantic bridge.

Consequently, it is our conviction that this paper will contribute with a set of novelties to the ontology engineering research area:

- The conceptualization of the ontology mapping negotiation problem based on the utility and meta-utility functions.
- The identification of matches as parameters of these functions.
- The service-oriented negotiation process based on the categorization of semantic bridges.

While the negotiation process is relatively simple and the utility functions have already been developed from the semantic bridging process, the major effort consists in configuring and customizing the meta-utility function. Nevertheless, tests are being carried out in parallel with customization, so that effective results are expected in the near future.

ACKNOWLEDGEMENTS

This work is partially supported by the Portuguese MCT-FCT project SANSKI (POCTI/2001/GES/41830).

Many thanks to Paulo Sousa for the fruitful discussions we had about negotiation. Thanks to Ana Barata for her revisions of this text.

REFERENCES

[1] A. Maedche, B. Motik, N. Silva, and R. Volz, "MAFRA - A MApping FRAmework for Distributed Ontologies", in 13th International Conference on Knowledge Engineering and Knowledge Management, Vol. 2473, pp. 235-250, Springer (2002).

[2] V. Tamma, M. Wooldridge, and I. Dickinson, "An Ontology-based Approach to Automated Negotiation", in AAMAS 2002 Fourth International Workshop on Agent-Mediated Electronic Commerce, pp. 219-237, Springer (2002).

[3] S. C. Bailin and W. Truszkowski, "Ontology Negotiation between Agents Supporting Intelligent Information Management", in Workshop on Ontologies in Agent Systems at the 5th International Conference on Autonomous Agents, pp. 13-20, (2001).

[4] A. Lomuscio, M. Wooldridge, and N. Jennings, "A Classification Scheme for Negotiation in Electronic Commerce", pp. 19-33, Springer-Verlag (2001).

[5] J. S. Rosenschein and G. Zlotkin, "Rules of encounter - designing conventions for automated negotiation among computers", MIT Press (1994).

[6] N. Silva and J. Rocha, "Multi-Dimensional Service-Oriented Ontology Mapping", Vol. 2, 1, pp. 50-80, Inderscience Publishers (2005).

[7] N. Silva and J. Rocha, "Semantic Web Complex Ontology Mapping", Vol. 1, 3-4, pp. 235-248, IOS Press (2004).

Introduction to the Ontology Alignment Evaluation 2005

Jérôme Euzenat
INRIA Rhône-Alpes
655 avenue de l'Europe
38330 Monbonnot, France
Jerome.Euzenat@inrialpes.fr

Heiner Stuckenschmidt
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam, The Netherland
heiner@cs.vu.nl

Mikalai Yatskevich
Dept. of Information and Communication Technology
University of Trento
Via Sommarive, 14
I-38050 Povo, Trento, Italia
yatskevi@unitn.it

The increasing number of methods available for schema matching/ontology integration suggests the need to establish a consensus for evaluation of these methods.
The Ontology Alignment Evaluation Initiative[1] is now a coordinated international initiative that has been set up for organising evaluation of ontology matching algorithms.
After the two events organized in 2004 (namely, the Information Interpretation and Integration Conference (I3CON) and the EON Ontology Alignment Contest [4]), this year one unique evaluation campaign is organised. Its outcome is presented at the Workshop on Integrating Ontologies held in conjunction with K-CAP 2005 at Banff (Canada) on October 2, 2005.
Since last year, we have set up a web site, improved the software on which the tests can be evaluated and set up some precise guidelines for running these tests. We have taken into account last year's remarks by (1) adding more coverage to the benchmarck suite and (2) elaborating two real world test cases (as well as addressing other technical comments).
This paper serves as a presentation to the 2005 evaluation campaign and introduction to the results provided in the following papers.

1. GOALS

Last year events demonstrated that it is possible to evaluate ontology alignment tools.

One intermediate goal of this year is to take into account the comments from last year contests. In particular, we aimed at improving the tests by widening their scope and variety. Benchmark tests are more complete (and harder) than before. Newly introduced tracks are more 'real-world' and of a considerable size.

[1] `http://oaei.inrialpes.fr`

K-CAP'05, Integrating ontologies workshop, October 2, 2005, Banff, Alberta, Canada.

The main goal of the Ontology Alignment Evaluation is to be able to compare systems and algorithms on the same basis and to allow drawing conclusions about the best strategies. Our ambition is that from such challenges, the tool developers can learn and improve their systems.

2. GENERAL METHODOLOGY

We present below the general methodology for the 2005 campaign. In this we took into account many of the comments made during the previous campaign.

2.1 Alignment problems

This year's campaign consists of three parts: it features two real world blind tests (anatomy and directory) in addition to the systematic benchmark test suite. By blind tests it is meant that the result expected from the test is not known in advance by the participants. The evaluation organisers provide the participants with the pairs of ontologies to align as well as (in the case of the systematic benchmark suite only) expected results. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in a standard format expressed in RDF/XML [2].

Like for last year's EON contest, a systematic benchmark series has been produced. The goal of this benchmark series is to identify the areas in which each alignment algorithm is strong and weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

The directory real world case consists of alignming web sites directory (like open directory or Yahoo's). It is more than two thousand elementary tests.

The anatomy real world case covers the domain of body anatomy and consists of two ontologies with an approximate size of several 10k classes and several dozen of relations.

The evaluation has been processed in three successive steps.

2.2 Preparatory phase

The ontologies and alignments of the evaluation have been provided in advance during the period between June 1st and July 1st. This was the occasion for potential participants to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this primary period is to be sure that the delivered tests make sense to the participants. The feedback is important, so all participants should not hesitate to provide it. The final test base has been released on July 4th. The tests did only change after this period for ensuring a better and easier participation.

2.3 Execution phase

During the execution phase the participants have used their algorithms to automatically match the ontologies of both part. The participants were required to only use one algorithm and the same set of parameters for all tests. Of course, it is regular to select the set of parameters that provide the best results. Beside the parameters the input of the algorithms must be the two provided ontology to align and any general purpose resource available to everyone (that is no resourse especially designed for the test). In particular, the participants should not use the data (ontologies and results) from other test sets to help their algorithm.

The participants have provided their alignment for each test in the Alignment format and a paper describing their results[2].

In an attempt to validate independently the results, they were required to provide a link to their program and parameter set used for obtaining the results.

2.4 Evaluation phase

The organizers have evaluated the results of the algorithms used by the participants and provided comparisons on the basis of the provided alignments.

In the case of the real world ontologies only the organizers will do the evaluation with regard to the withheld alignments.

The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures we have computed a true global precision and recall (not a mere average). We have also computed precision/recall graphs for some of the participants (see below).

Finally, in an experimental way, we will attempt this year at reproducing the results provided by participants (validation).

3. COMMENTS ON THE EXECUTION

We had more participants than last year's event and it is easier to run these tests (qualitatively we had less comments and the results were easier to analyse). We summarize the list of participants in Table 1. As can be seen, not all participants provided results for all the tests and not all system were correctly validated. However, when the tests are straightforward to process (benchmarks and directory), participants provided results. The main problems with the anatomy test was its size. We also mentioned the kind of results sent by each participant (relations and confidence).

We note that the time devoted for performing these tests (three months) and the period allocated for that (summer) is relatively short and does not really allow the participants to analyse their results and improve their algorithms. On the one hand, this prevents having algorithms really tuned for the contests, on the other hand, this can be frustrating for the participants. We should try to allow more time for participating next time.

Complete results are provided on http://oaei.inrialpes.fr/2005/results/. These are the only official results (the results presented here are only partial and prone to correction). The summary of results track by track is provided below.

4. BENCHMARK

The benchmark test case improved on last year's base by providing new variations of the reference ontology (last year the test contained 19 individual tests while this year it contains 53 tests). These new tests are supposed to be more difficult. The other improvement was the introduction of other evaluation metrics (real global precision and recall as well as the generation of precision-recall graphs).

4.1 Test set

The systematic benchmark test set is built around one reference ontology and many variations of it. The participants have to match this reference ontology with the variations. These variations are focussing the characterisation of the behaviour of the tools rather than having them compete on real-life problems. The ontologies are described in OWL-DL and serialized in the RDF/XML format.

Since the goal of these tests is to offer some kind of permanent benchmarks to be used by many, the test is an extension of last year EON Ontology Alignment Contest. Test numbering (almost) fully preserves the numbering of the first EON contest.

The reference ontology is based on the one of the first EON Ontology Alignment Contest. It is improved by comprising a number of circular relations that were missing from the first test. The domain of this first test is Bibliographic references. It is, of course, based on a subjective view of what must be a bibliographic ontology. There can be many different classifications of publications (based on area, quality, etc.). We choose the one common among scholars based on mean of publications; as many ontologies below (tests #301-304), it is reminiscent to BibTeX.

The reference ontology is that of test #101. It contains 33

[2] Andreas Hess from the UCDublin has not been able to provide a paper in due time. Description of his system can be found in [3]

Name	System	Benchmarks	Directory	Anatomy	Validated	Relations	Confidence
U. Karlsruhe	FOAM	√	√			=	cont
U. Montréal/INRIA	OLA	√	√		√	=	cont
IRST Trento	CtxMatch 2	√	√			=, ≤	1.
U. Southampton	CMS	√	√	√		=	1.
Southeast U. Nanjin	Falcon	√	√	√	√	=	1.
UC. Dublin	?	√	√			=	cont
CNR/Pisa	OMAP	√	√			=	1.

Table 1: Participants and the state of the state of their submissions. Confidence is given as 1/0 or continuous values.

named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals.

The reference ontology is put in the context of the semantic web by using other external resources for expressing non bibliographic information. It takes advantage of FOAF (http://xmlns.com/foaf/0.1/) and iCalendar (http://www.w3.org/2002/12/cal/) for expressing the People, Organization and Event concepts. Here are the external reference used:

- http://www.w3.org/2002/12/cal/#:Vevent (defined in http://www.w3.org/2002/12/cal/ical.n3 and supposedly in http://www.w3.org/2002/12/cal/ical.rdf)
- http://xmlns.com/foaf/0.1/#:Person (defined in http://xmlns.com/foaf/0.1/index.rdf)
- http://xmlns.com/foaf/0.1/#:Organization (defined in http://xmlns.com/foaf/0.1/index.rdf)

This reference ontology is a bit limited in the sense that it does not contain attachement to several classes.

Similarly the kind of proposed alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1.

There are still three group of tests in this benchmark:

- simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;
- systematic tests (2xx) that were obtained by discarding some features of the reference ontology. The considered features were (names, comments, hierarchy, instances, relations, restrictions, etc.). The tests are systematically generated to as to start from some reference ontology and discarding a number of information in order to evaluate how the algorithm behave when this information is lacking. These tests were largely improved from last year by combining all feature discarding.
- four real-life ontologies of bibliographic references (3xx) that were found on the web and left mostly untouched (they were added xmlns and xml:base attributes).

Table 5 summarize what has been retracted from the reference ontology in the systematic tests. There are here 6 categories of alteration:

Name Name of entities that can be replaced by (R/N) random strings, (S)ynonyms, (N)ame with different conventions, (F) strings in another language than english.

Comments Comments can be (N) suppressed or (F) translated in another language.

Specialization Hierarchy can be (N) suppressed, (E)xpansed or (F)lattened.

Instances can be (N) suppressed

Properties can be (N) suppressed or (R) having the restrictions on classes discarded.

Classes can be (E)xpanded, i.e., relaced by several classes or (F)latened.

4.2 Results

Table 2 provide the consolidated results, by groups of tests. Table 6 contain the full results.

We display the results of participants as well as those given by some very simple edit distance algorithm on labels (edna). The computed values here are real precision and recall and not a simple average of precision and recall. This is more accurate than what has been computed last year.

As can be seen, the 1xx tests are relatively easy for most of the participants. The 2xx tests are more difficult in general while 3xx tests are not significantly more difficult than 2xx for most participants. The real interesting results is that there are significant differences across algorithms within the 2xx test series. Most of the best algorithms were combining different ways of finding the correspondence. Each of them is able to perform quite well on some tests with some methods. So the key issue seems to have been the combination of different methods (as described by the papers).

One algorithm, Falcon, seems largely dominant. But a group of other algorithms (Dublin, OLA, FOAM) are computing against each other. While the CMS and CtxMatch currently perform at a lower rate. Concerning these algorithm, CMS

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.96	1.00	1.00	1.00	0.98	0.65	0.10	0.34	1.00	0.99	0.74	0.20	0.96	1.00	1.00	1.00
2xx	0.41	0.56	0.90	0.89	0.89	0.69	0.08	0.23	0.94	0.71	0.81	0.18	0.31	0.68	0.80	0.73
3xx	0.47	0.82	0.93	0.83	0.92	0.69	0.08	0.22	0.67	0.60	0.93	0.18	0.93	0.65	0.50	0.48
H-means	0.45	0.61	0.91	0.89	0.90	0.69	0.08	0.24	0.92	0.72	0.81	0.18	0.35	0.70	0.80	0.74

Table 2: Means of results obtained by participants (corresponding to harmonic means)

seems to priviledge precision and performs correctly in this (OLA seems to have privileged recall with regard to last year). CtxMatch has the difficulty of delivering many subsumption assertions. These assertions are taken by our evaluation procedure positively (even if equivalence assertions were required), but since there are many more assertions than in the reference alignments, this brings the result down.

These results can be compared with last year's results given in Table 3 (with aggregated measures computed at new with the methods of this year). For the sake of comparison, the results of this year on the same test set as last year are given in Table 4. As can be expected, the two participants of both challenges (Karlsruhe2 corresponding to foam and Montréal/INRIA corresponding to ola) have largely improved their results. The results of the best participants this year are over or similar to those of last year. This is remarkable, because participants did not tune their algorithms to the challenge of last year but to that of this year (more difficult since it contains more test of a more difficult nature and because of the addition of cycles in them).

So, it seem that the field is globally progressing.

Because of the precision/recall trade-off, as noted last year, it is difficult to compare the middle group of systems. In order to assess this, we attempted to draw precision recall graphs. We provide in Figure 1 the averaged precision and recall graphs of this year. They involve only the results of all participants. However, the results corresponding to participants who provided confidence measures different of 1 or 0 (see Table 1) can be considered as approximation. Moreover, for reason of time these graphs have been computed by averaging the graphs of each tests (instead to pure precision and recall).

These graphs are not totally faithful to the algorithms because participants have cut their results (in order to get high overall precision and recall). However, they provide a rough idea about the way participants are fighting against each others in the precision recall space. It would be very useful that next year we ask for results with continuous ranking for drawing these kind of graphs.

4.3 Comments

A general comments, we remarks, that it is still difficult for participants to provide results that correspond to the challenge (incorrect format, alignment with external entities). Because time is short and we try to avoid modifying provided results, this test is still a test of both algorithms and their ability to deliver a required format. However, some teams are really performant in this (and the same teams generally have their tools validated relatively easily).

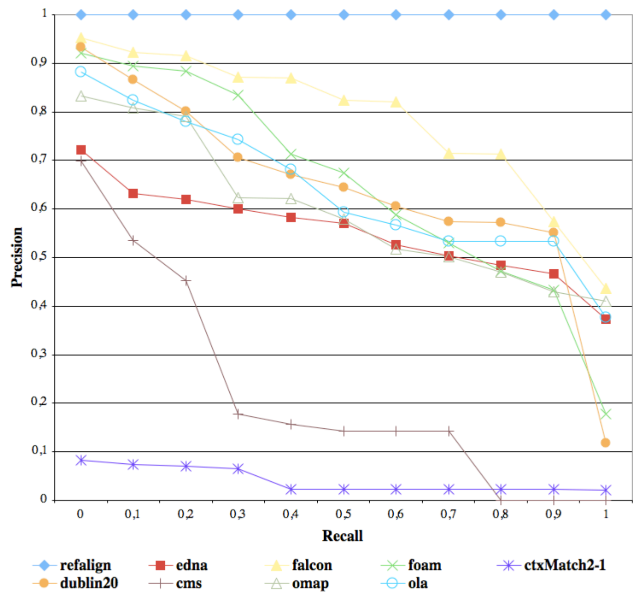


Figure 1: Precision-recall graphs

The evaluation of algorithms like ctxMatch which provide many subsumption assertions is relatively inadequate. Even if the test can remain a test of inference equivalence. It would be useful to be able to count adequately, i.e., not negatively for precision, true assertions like owl:Thing subsuming another concept. We must develop new evaluation methods taken into account these assertions and the semantics of the OWL language.

As a side note: all participants but one have used the UTF-8 version of the tests, so next time, this one will have to be the standard one with iso-latin as an exception.

5. DIRECTORY

5.1 Data set

algo	karlsruhe2		umontreal		fujitsu		stanford	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	NaN	0.00	0.57	0.93	0.99	1.00	0.99	1.00
2xx	0.60	0.46	0.54	0.87	0.93	0.84	0.98	0.72
3xx	0.90	0.59	0.36	0.57	0.60	0.72	0.93	0.74
H-means	0.65	0.40	0.52	0.83	0.88	0.85	0.98	0.77

Table 3: EON 2004 results with this year's aggregation method.

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.96	1.00	1.00	1.00	0.98	0.65	0.10	0.34	1.00	0.99	0.74	0.20	0.96	1.00	1.00	1.00
2xx	0.66	0.72	0.98	0.97	0.87	0.73	0.09	0.25	0.98	0.92	0.91	0.20	0.89	0.79	0.89	0.86
3xx	0.47	0.82	0.93	0.83	0.92	0.69	0.08	0.22	0.67	0.60	0.93	0.18	0.93	0.65	0.50	0.48
H-means	0.66	0.78	0.97	0.96	0.74	0.59	0.09	0.26	0.94	0.88	0.65	0.18	0.90	0.81	0.85	0.83

Table 4: This year's results on EON 2004 test bench.

The data set exploited in the web directories matching task was constructed from Google, Yahoo and Looksmart web directories as described in [1]. The key idea of the data set construction methodology was to significantly reduce the search space for human annotators. Instead of considering the full mapping task which is very big (Google and Yahoo directories have up to $3*10^5$ nodes each: this means that the human annotators need to consider up to $(3*10^5)^2 = 9*10^{10}$ mappings), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the dataset described in [1] human annotators consider only 2265 mappings instead of the full mapping problem.

The major limitation of the current dataset version is the fact that it contains only true positive mappings (i.e., the mappings which tell that the particular relation holds between nodes in both trees). At the same time it does not contain true negative mappings (or zero mappings) which tell that there are no relation holding between pair of nodes. Notice that manually constructed mapping sets (such as ones presented for systematic tests) assume all the mappings except true positives to be true negatives. This assumption does not hold in our case since dataset generation technique guarantee correctness but not completeness of the produced mappings. This limitation allows to use the dataset only for evaluation of Recall but not Precision (since Recall is defined as ratio of correct mappings found by the system to the total number of correct mappings). At the same time measuring Precision necessarily require presence of the true negatives in the dataset since Precision is defined as a ratio of correct mappings found by the system to all the mappings found by the system. This means that all the systems will have 100% Precision on the the dataset since there are no incorrect mappings to be found.

The absence of true negatives has significant implications on the testing methodology in general. In fact most of the state of the art matching systems can be tuned either to produce the results with better Recall or to produce the results with better Precision. For example, the system which produce the equivalence relation on any input will always have 100% Recall. Therefore, the main methodological goal in the evaluation was to prevent Recall tuned systems from getting of unrealistically good results on the dataset. In order to accomplish this goal the double validation of the results was performed. The participants were asked for the binaries of their systems and were required to use the same sets of parameters in both web directory and systematic matching tasks. Then the results were double checked by organizers to ensure that the latter requirement is fulfilled by the authors. The process allow to recognize Recall tuned systems by analysis of systematic tests results.

The dataset originally was presented in its own format. The mappings were presented as pairwise relationships between the nodes of the web directories identified by their paths to root. Since the systems participating in the evaluation all take OWL ontologies as input the conversion of the dataset to OWL was performed. In the conversion process the nodes of the web directories were modelled as classes and classification relation connecting the nodes was modelled as rdfs:subClassOf relation. Therefore the matching task was presented as 2265 tasks of finding the semantic relation holding between pathes to root in the web directories modelled as sub class hierarchies.

5.2 Results

The results for web directory matching task are presented on Figure 2. As from the figure the web directories matching task is a very hard one. In fact the best systems found about 30% of mappings form the dataset (i.e., have Recall about 30%).

The evaluation results can be considered from two perspec-

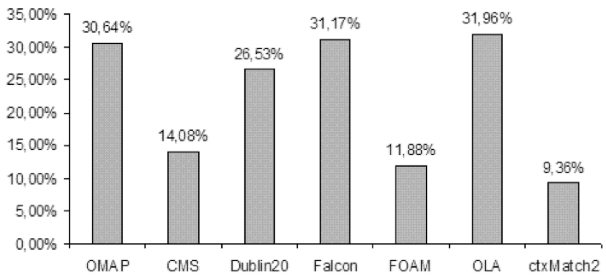


Figure 2: Recall for web directories matching task

tives. On the one hand, they are good indicator of real world ontologies matching complexity. On the other hand the results can provide information about the quality of the dataset used in the evaluation. The desired mapping dataset quality properties were defined in [1] as *Complexity*, *Discrimination capability*, *Incrementality* and *Correctness*. The first means that the dataset is "hard" for state of the art matching systems, the second that it discriminates among the various matching solutions, the third that it is effective in recognizing weaknesses in the state of the art matching systems and the fourth that it can be considered as a correct one.

The results of the evaluation give us some evidence for *Complexity* and *Discrimination capability* properties. As from Figure 2 TaxME dataset is hard for state of the art matching techniques since there are no systems having Recall more than 35% on the dataset. At the same time all the matching systems together found about 60% of mappings. This means that there is a big space for improvements for state of the art matching solutions.

Consider Figure 3. It contains partitioning of the mappings found by the matching systems. As from the figure 44% of the mappings found by any of the matching systems was found by only one system. This is a good argument to the dataset *Discrimination capability* property.

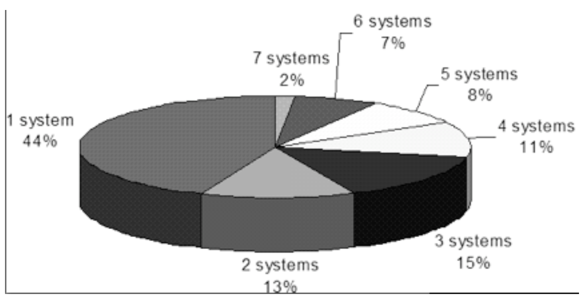


Figure 3: Partitioning of the mappings found by the matching systems

5.3 Comments

The web directories matching task is an important step towards evaluation on the real world matching problems. At the same time there are a number of limitations which makes the task only an intermediate step. First of all the current version of the mapping dataset provides correct but not complete set of the reference mappings. The new mapping dataset construction techniques can overcome this limitation. In the evaluation the mapping task was split to the the tiny subtasks. This strategy allowed to obtain results form all the matching systems participating in the evaluation. At the same time it hides computational complexity of "real world" matching (the web directories have up to 10^5 nodes) and may affect the results of the tools relying on "look for similar siblings" heuristic.

The results obtained on the web directories matching task coincide well with previously reported results on the same dataset. According to [1] generic matching systems (or the systems intended to match any graph-like structures) have Recall from 30% to 60% on the dataset. At the same time the real world matching tasks are very hard for state of the art matching systems and there is a huge space for improvements in the ontology matching techniques.

6. ANATOMY

6.1 Test set

The focus of this task is to confront existing alignment technology with real world ontologies. Our aim is to get a better impression of where we stand with respect to really hard challenges that normally require an enormous manual effort and requires in-depth knowledge of the domain.

The task is placed in the medical domain as this is the domain where we find large, carefully designed ontologies. The specific characteristics of the ontologies are:

- Very large models: be prepared to handle OWL models of more than 50MB !
- Extensive Class Hierarchies: then thousands of classes organized according to different views on the domain.
- Complex Relationships: Classes are connected by a number of different relations.
- Stable Terminology: The basic terminology is rather stable and should not differ too much in the different model
- Clear Modelling Principles: The modelling principles are well defined and documented in publications about the ontologies

This implies that the task will be challenging from a technological point of view, but there is guidance for tuning matching approach that needs to be taken into account.

The ontologies to be aligned are different representations of human anatomy developed independently by teams of medical experts. Both ontologies are available in OWL format and mostly contain classes and relations between them. The use of axioms is limited.

6.1.1 The Foundational Model of Anatomy

The Foundational Model of Anatomy is a medical ontology developed by the University of Washington. We extracted an OWL version of the ontology from a Protege database. The model contains the following information:

- Class hierarchy;
- Relations between classes;
- Free text documentation and definitions of classes;
- Synonyms and names in different languages.

6.1.2 The OpenGalen Anatomy Model

The second ontology is the Anatomy model developed in the OpenGalen Project by the University of Manchester. We created an OWL version of the ontology using the export functionality of Protege. The model contains the following information:

- Concept hierarchy;
- Relations between concepts.

The task is to find alignment between classes in the two ontologies. In order to find the alignment, any information in the two models can be used. In addition, it is allowed to use background knowledge, that has not specifically been created for the alignment tasks (i.e., no hand-made mappings between parts of the ontologies). Admissible background knowledge are other medical terminologies such as UMLS as well as medical dictionaries and document sets. Further, results must not be tuned manually, for instance, by removing obviously wrong mappings.

6.2 Results

At the time of printing we are not able to provide results of evaluation on this test.

Validation of the results on the medical ontologies matching task is still an open problem. The results can be replicated in straightforward way. At the same time there are no sufficiently big set of the reference mappings what makes impossible calculation of the matching quality measures.

We are currently developing an approach for creating such a set is to exploit semi-automatic reference mappings acquisition techniques. The underlying principle is that the task of creating such a reference alignment is fundamentally different from the actual mapping problem. In particular, we believe that automatically creating reference alignments is easier than solving the general mapping problem. The reason for this is, that methods for creating general mappings have to take into account both, correctness and completeness of the generated mappings. This is difficult, because allying very strict heuristics will lead to correct, but very incomplete mappings, using loose heuristics for matching nodes will create a rather complete, but often incorrect set of mappings. In our approach for generating reference alignments, we completely focus on the correctness. The result is a small set of reference mappings that we can assume to be correct. We can evaluate matching approaches against this set of mappings. The idea is that the matching approaches should at least be able to determine these mappings. From the result, we can extrapolate the expected completeness of a matching algorithm.

We assume that the task is to create a reference alignment for two a number of known conceptual models. In contrast to existing work [1] we do not assume that instance data is available or that the models are represented in the same way or using the same language. Normally, the models will be from the same domain (eg. medicine or business). The methodology consists of four basic steps. In the first step, basic decisions are made about the representation of the conceptual models and instance data to be used. In the second step instance data is created by selecting it from an existing set or by classifying data according to the models under consideration. In the third step, the generated instance data is used to generate candidate mappings based on shared instances. In the forth step finally, the candidate mappings are evaluated against a set of quality criteria and the final set of reference mappings is determined.

6.2.1 Step 1. Preparation

The first step of the process is concerned with data preparation. In particular, we have to transform the conceptual models into a graph representation and select and prepare the appropriate instance data to be used to analyze overlap between concepts in the different models. We structure this step based on the KDD process for Knowledge Discovery and Data Mining.

6.2.2 Step 2. Instance Classification

In the second step the chosen instance data is classified according to the different conceptual models. For this purpose, an appropriate classification method has to be chosen that fits the data and the conceptual model. Further, the result of the classification process has to be evaluated. For this step we rely on established methods from Machine Learning and Data Mining.

6.2.3 Step 3. Hypothesis Generation

In the third step, we generate hypothesis for reference mappings based on shared instances created in the first two steps. In this step, we prune the classification by removing instances that are classified with a low confidence and selecting subsets of the conceptual models that show sufficient overlap. We further compute a degree of overlap between concepts in the different models and based on this degree of overlap select a set of reference mappings between concepts with a significant overlap.

6.3 Step 4. Evaluation

In the last step, the generated reference mapping is evaluated against the result of different matching systems as described in **??** using a number of criteria for a reference mapping. These criteria include correctness, complexity of the mapping problem and the ability of the mappings to discriminate between different matching approaches.

We are testing this methodology using a data set of medical documents called OHSUMED. The data set contains 350.000 articles from medical journals covering all aspects of medicine. For classifying these documents according to the two ontologies of anatomy, we use the collexis text indexing and retrieval system that implements a number of automatic methods for assigning concepts to documents. Currently, we are testing the data set and the system on a subset of UMLS with known mappings in order to assess the suitability of the methodology. The generation of the reference mappings for the Anatomy case will proceed around the end of 2005 and we are hopeful to have thoroughly tested set of reference mappings for the 2006 alignment challenge.

6.4 Comments

We had very few participants able to even produce the alignments between both ontologies. This is mainly due to their inability to load these ontologies with current OWL tools (caused either by the size of the ontologies or errors in the OWL).

7. RESULT VALIDATION

As can be seen from the procedure, the results published in the following papers are not obtained independently. The results provided here have been computed from the alignment provided by the participants and can be considered as the official results of the evaluation.

In order to go one step further, we have attempted, this year, to generate the results obtained by the participants from their tools. The tools for which the results have been validated independently are marked in Table 1.

8. LESSON LEARNED

A) It seems that there are more and more tools able to jump in this kind of tests.

B) Contrary to last year it seems that the tools are more robusts and people deal with more wider implementation of OWL. However, this can be that we tuned the tests so that no one has problems.

C) Contrary to what many people think, it is not that easy to find ontological corpora suitable for this evaluation test. From the proposals we had from last year, only one proved to be usable and with great difficulty (on size, conformance and juridical aspects).

D) The extension of the benchmark tests towards more coverage of the space is relatively systematic. However, it would be interesting and certainly more realistic, instead of crippling all names to do it for some random proportion of them (5% 10% 20% 40% 60% 100% random change). This has not been done for reason of time.

E) The real world benchmarks were huge benchmarks. Two different strategies have been taken with them: cutting them in a huge set of tiny benchmark or providing them as is. The first solution brings us away from "real world", while the second one raised serious problems to the participants. It would certainly be worth designing these tests in order to assess the current limitation of the tools by providing an increasingly large sequence of such tests (0.1%, 1%, 10%, 100% of the corpus for instance).

F) Validation of the results are quite difficult to establish.

9. FUTURE PLANS

The future plans for the Ontology Alignement Evaluation Initiative are certainly to go ahead and improving the functioning of these evaluation campaign. This most surely involves:

- Finding new real world cases;
- Improving the tests along the lesson learned;
- Accepting continuous submissions (through validation of the results);
- Improving the measures to go beyond precision and recall.

Of course, these are only suggestions and other ideas could come during the wrap-up meeting in Banff.

10. CONCLUSION

In summary, the tests that have been run this year are harder and more complete than those of last year. However, more teams participated and the results tend to be better. This shows that, as expected, the field of ontology alignment is getting stronger (and we hope that evaluation is contributing to this progress).

Reading the papers of the participants should help people involved in ontology matching to find what make these algorithms work and what could be improved.

The Ontology Alignment Evaluation Initiative will continue these tests by improving both test cases and test methodology for being more accurate. It can be found at:

`http://oaei.inrialpes.fr`.

11. ACKNOWLEDGEMENTS

We warmly thank each participant of this contest. We know that they worked hard for having their results ready and they provided insightful papers presenting their experience. The best way to learn about the results remains to read what follows.

Many thanks are due to the teams at the University of Washington and the University of Manchester for allowing us to use their ontologies of anatomy.

The other members of the Ontology Alignment Evaluation Initiative Steering committee: Benjamin Ashpole (Lockheed Martin Advanced Technology Lab.), Marc Ehrig (University of Karlsruhe), Lewis Hart (Applied Minds), Todd Hughes (Lockheed Martin Advanced Technology Labs), Natasha Noy (Stanford University), and Petko Valtchev (Université de Montréal, DIRO)

This work has been partially supported by the Knowledge Web European network of excellence (IST-2004-507482).

12. REFERENCES

[1] Paolo Avesani, Fausto Giunchiglia, and Michael Yatskevich. A large scale taxonomy mapping evaluation. In *Proceedings of International Semantic Web Conference (ISWC)*, 2005.

[2] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd international semantic web conference, Hiroshima (JP)*, pages 698–712, 2004.

[3] Andreas Heß and Nicholas Kushmerick. Iterative ensemble classification for relational data: A case study of semantic web services. In *Proceedings of the 15th European Conference on Machine Learning*, Pisa, Italy, 2004.

[4] York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the 3rd Evaluation of Ontology-based tools (EON)*, 2004.

Montbonnot, Amsterdam, Trento, September 7th, 2005

#	Name	Com	Hier	Inst	Prop	Class	Comment
101							Reference alignment
102							Irrelevant ontology
103							Language generalization
104							Language restriction
201	R						No names
202	R	N					No names, no comments
203		N					No comments (was missspelling)
204	C						Naming conventions
205	S						Synonyms
206	F	F					Translation
207	F						
208	C	N					
209	S	N					
210	F	N					
221			N				No specialisation
222			F				Flatenned hierarchy
223			E				Expanded hierarchy
224				N			No instance
225					R		No restrictions
226							No datatypes
227							Unit difference
228					N		No properties
229							Class vs instances
230						F	Flattened classes
231*						E	Expanded classes
232			N	N			
233			N		N		
236				N	N		
237			F	N			
238			E	N			
239			F		N		
240			E		N		
241			N	N	N		
246			F	N	N		
247			E	N	N		
248	N	N	N				
249	N	N		N			
250	N	N			N		
251	N	N	F				
252	N	N	E				
253	N	N	N	N			
254	N	N	N		N		
257	N	N		N	N		
258	N	N	F	N			
259	N	N	E	N			
260	N	N	F		N		
261	N	N	E		N		
262	N	N	N	N	N		
265	N	N	F	N	N		
266	N	N	E	N	N		
301							Real: BibTeX/MIT
302							Real: BibTeX/UMBC
303							Real: Karlsruhe
304							Real: INRIA

Table 5: Structure of the systematic benchmark test-case

algo	edna		falcon		foam		ctxMatch2-1		dublin20		cms		omap		ola	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
101	0.96	1.00	1.00	1.00	n/a	n/a	0.10	0.34	1.00	0.99	n/a	n/a	0.96	1.00	1.00	1.00
103	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	0.67	0.25	0.96	1.00	1.00	1.00
104	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	0.80	0.34	0.96	1.00	1.00	1.00
201	0.03	0.03	0.98	0.98	n/a	n/a	0.00	0.00	0.96	0.96	1.00	0.07	0.80	0.38	0.71	0.62
202	0.03	0.03	0.87	0.87	0.79	0.52	0.00	0.00	0.75	0.28	0.25	0.01	0.82	0.24	0.66	0.56
203	0.96	1.00	1.00	1.00	1.00	1.00	0.08	0.34	1.00	0.99	1.00	0.24	0.96	1.00	1.00	1.00
204	0.90	0.94	1.00	1.00	1.00	0.97	0.09	0.28	0.98	0.98	1.00	0.24	0.93	0.89	0.94	0.94
205	0.34	0.35	0.88	0.87	0.89	0.73	0.05	0.11	0.98	0.97	1.00	0.09	0.58	0.66	0.43	0.42
206	0.51	0.54	1.00	0.99	1.00	0.82	0.05	0.08	0.96	0.95	1.00	0.09	0.74	0.49	0.94	0.93
207	0.51	0.54	1.00	0.99	0.96	0.78	0.05	0.08	0.96	0.95	1.00	0.09	0.74	0.49	0.95	0.94
208	0.90	0.94	1.00	1.00	0.96	0.89	0.09	0.28	0.99	0.96	1.00	0.19	0.96	0.90	0.94	0.94
209	0.35	0.36	0.86	0.86	0.78	0.58	0.05	0.11	0.68	0.56	1.00	0.04	0.41	0.60	0.43	0.42
210	0.51	0.54	0.97	0.96	0.87	0.64	0.05	0.08	0.96	0.82	0.82	0.09	0.88	0.39	0.95	0.94
221	0.96	1.00	1.00	1.00	1.00	1.00	0.12	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
222	0.91	0.99	1.00	1.00	0.98	0.98	0.11	0.31	1.00	0.99	1.00	0.23	0.96	1.00	1.00	1.00
223	0.96	1.00	1.00	1.00	0.99	0.98	0.09	0.34	0.99	0.98	0.96	0.26	0.96	1.00	1.00	1.00
224	0.96	1.00	1.00	1.00	1.00	0.99	0.10	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
225	0.96	1.00	1.00	1.00	0.00	0.00	0.08	0.34	1.00	0.99	0.74	0.26	0.96	1.00	1.00	1.00
228	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.74	0.76	0.92	1.00	1.00	1.00
230	0.71	1.00	0.94	1.00	0.94	1.00	0.08	0.35	0.95	0.99	1.00	0.26	0.89	1.00	0.95	0.97
231	0.96	1.00	1.00	1.00	0.98	0.98	0.10	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
232	0.96	1.00	1.00	1.00	1.00	0.99	0.12	0.34	1.00	0.99	1.00	0.27	0.96	1.00	1.00	1.00
233	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.81	0.76	0.92	1.00	1.00	1.00
236	0.38	1.00	1.00	1.00	1.00	1.00	0.09	1.00	1.00	1.00	0.74	0.76	0.92	1.00	1.00	1.00
237	0.91	0.99	1.00	1.00	1.00	0.99	0.11	0.31	1.00	0.99	1.00	0.23	0.95	1.00	0.97	0.98
238	0.96	1.00	0.99	0.99	1.00	0.99	0.07	0.34	0.99	0.98	0.96	0.26	0.96	1.00	0.99	0.99
239	0.28	1.00	0.97	1.00	0.97	1.00	0.14	1.00	0.97	1.00	0.71	0.76	0.85	1.00	0.97	1.00
240	0.33	1.00	0.97	1.00	0.94	0.97	0.10	1.00	0.94	0.97	0.71	0.73	0.87	1.00	0.97	1.00
241	0.38	1.00	1.00	1.00	1.00	1.00	0.12	1.00	1.00	1.00	0.81	0.76	0.92	1.00	1.00	1.00
246	0.28	1.00	0.97	1.00	0.97	1.00	0.14	1.00	0.97	1.00	0.71	0.76	0.85	1.00	0.97	1.00
247	0.33	1.00	0.94	0.97	0.94	0.97	0.10	1.00	0.94	0.97	0.71	0.73	0.87	1.00	0.97	1.00
248	0.06	0.06	0.84	0.82	0.89	0.51	0.00	0.00	0.71	0.25	0.25	0.01	0.82	0.24	0.59	0.46
249	0.04	0.04	0.86	0.86	0.80	0.51	0.00	0.00	0.74	0.29	0.25	0.01	0.81	0.23	0.59	0.46
250	0.01	0.03	0.77	0.70	1.00	0.55	0.00	0.00	1.00	0.09	0.00	0.00	0.05	0.45	0.30	0.24
251	0.01	0.01	0.69	0.69	0.90	0.41	0.00	0.00	0.79	0.32	0.25	0.01	0.82	0.25	0.42	0.30
252	0.01	0.01	0.67	0.67	0.67	0.35	0.00	0.00	0.57	0.22	0.25	0.01	0.82	0.24	0.59	0.52
253	0.05	0.05	0.86	0.85	0.80	0.40	0.00	0.00	0.76	0.27	0.25	0.01	0.81	0.23	0.56	0.41
254	0.02	0.06	1.00	0.27	0.78	0.21	0.00	0.00	NaN	0.00	0.00	0.00	0.03	1.00	0.04	0.03
257	0.01	0.03	0.70	0.64	1.00	0.64	0.00	0.00	1.00	0.09	0.00	0.00	0.05	0.45	0.25	0.21
258	0.01	0.01	0.70	0.70	0.88	0.39	0.00	0.00	0.79	0.32	0.25	0.01	0.82	0.25	0.49	0.35
259	0.01	0.01	0.68	0.68	0.61	0.34	0.00	0.00	0.59	0.21	0.25	0.01	0.82	0.24	0.58	0.47
260	0.00	0.00	0.52	0.48	0.75	0.31	0.00	0.00	0.75	0.10	0.00	0.00	0.05	0.86	0.26	0.17
261	0.00	0.00	0.50	0.48	0.63	0.30	0.00	0.00	0.33	0.06	0.00	0.00	0.01	0.15	0.14	0.09
262	0.01	0.03	0.89	0.24	0.78	0.21	0.00	0.00	NaN	0.00	0.00	0.00	0.03	1.00	0.20	0.06
265	0.00	0.00	0.48	0.45	0.75	0.31	0.00	0.00	0.75	0.10	0.00	0.00	0.05	0.86	0.22	0.14
266	0.00	0.00	0.50	0.48	0.67	0.36	0.00	0.00	0.33	0.06	0.00	0.00	0.01	0.15	0.14	0.09
301	0.48	0.79	0.96	0.80	0.83	0.31	0.10	0.07	0.74	0.64	1.00	0.13	0.94	0.25	0.42	0.38
302	0.31	0.65	0.97	0.67	0.97	0.65	0.14	0.27	0.62	0.48	1.00	0.17	1.00	0.58	0.37	0.33
303	0.40	0.82	0.80	0.82	0.89	0.80	0.04	0.29	0.51	0.53	1.00	0.18	0.93	0.80	0.41	0.49
304	0.71	0.95	0.97	0.96	0.95	0.96	0.11	0.26	0.75	0.70	0.85	0.22	0.91	0.91	0.74	0.66
H-means	0.45	0.61	0.91	0.89	0.90	0.69	0.08	0.24	0.92	0.72	0.81	0.18	0.35	0.70	0.80	0.74

Table 6: Full results

FOAM – Framework for Ontology Alignment and Mapping Results of the Ontology Alignment Evaluation Initiative

Marc Ehrig
Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
ehrig@aifb.uni-karlsruhe.de

York Sure
Institute AIFB
University of Karlsruhe
76128 Karlsruhe, Germany
sure@aifb.uni-karlsruhe.de

ABSTRACT
This paper briefly introduces the system FOAM and its underlying techniques. We then discuss the results returned from the evaluation. They were very promising and at the same time clarifying. Concisely: labels are very important; structure helps in cases where labels do not work; dictionaries may provide additional evidence; ontology management systems need to deal with OWL-Full. The results of this paper will also be very interesting for other participants, showing specific strengths and weaknesses of our approach.

1. PRESENTATION OF THE SYSTEM

1.1 State, purpose, general statement
In recent years, we have seen a range of research work on methods proposing alignments [1; 2]. When we tried to apply these methods to some of the real-world scenarios we address in other research contributions [3], we found that existing alignment methods did not suit the given requirements:

- high quality results;
- efficiency;
- optional user-interaction;
- flexibility with respect to use cases;
- and easy adjusting and parameterizing.

We wanted to provide the end-user with a tool taking ontologies as input and returning alignments (with explanations) as output meeting these requirements.

1.2 Specific techniques used
We have observed that alignment methods like QOM [4] or PROMPT [2] may be mapped onto a generic alignment process (Figure 1). Here we will only mention the six major steps to clarify the underlying approach for the FOAM tool. We refer to [4] for a detailed description.

1. Feature Engineering, i.e. select excerpts of the overall ontology definition to describe a specific. This includes individual features, e.g. labels, structural features, e.g. subsumption, but also more complex features as used in OWL, e.g. restrictions.
2. Search Step Selection, i.e. choose two entities from the two ontologies to compare (e_1,e_2).
3. Similarity Assessment, i.e. indicate a similarity for a given description (feature) of two entities (e.g., $sim_{superConcept}(e_1,e_2)=1.0$).
4. Similarity Aggregation, i.e. aggregate the multiple similarity assessments for one pair of entities into a single measure.
5. Interpretation, i.e. use all aggregated numbers, a threshold and an interpretation strategy to propose the alignment ($align(e_1)$=‘ e_2’). This may also include a user validation.
6. Iteration, i.e. as the similarity of one alignment influences the similarity of neighboring entity pairs; the equality is propagated through the ontologies.

Finally, we receive alignments linking the two ontologies.

This general process was extended to meet the mentioned requirements.

- High quality results were achieved through a combination of a rule-based approach and a machine learning approach. Underlying individual rules such as, if the super-concepts are similar the entities are similar, have been assigned weights by a machine learnt decision tree [5]. Especially steps 1, 3 and 4 were adjusted for this. Currently, our approach does not make use of additional background knowledge such as dictionaries here.
- Efficiency was mainly achieved through an intelligent selection of candidate alignments in 2, the search step selection [4].
- User-interaction allows the user intervening during the interpretation step. By presenting the doubtable alignments (and only these) to the user, overall quality can be considerably increased. Yet this happens in a minimal invasive manner.
- The system can automatically set its parameters according to a list of given use cases, such as ontology merging, versioning, ontology mapping, etc. The parameters also change according to the ontologies to align, e.g., big ontologies always require the efficient approach, whereas smaller ones do not [6].
- All these parameters may be set manually. This allows using the implementation for very specific tasks as well.
- Finally, FOAM has been implemented in Java and is freely available, thus extensible.

1.3 Adaptations made for the contest
No special adjustments have been made for the contest. However, some elements have been deactivated. Due to the small size of the benchmark and directory ontologies efficiency was not used, user-interaction was removed for the initiative, and no specific use case parameters were taken. A general alignment procedure was applied.

The system used for the evaluation is a derivative of the ontology alignment tool used in last year’s contests I3Con [7] and EON-OAC [8].

2. RESULTS

All tests were performed on a standard notebook under Windows. FOAM has been implemented in Java with all its advantages and disadvantages.

The individual results of the benchmark ontologies were grouped. Further, one short section describes the testing of the directory and anatomy ontologies. The concrete results can be found in Section 6.3 of this paper.

2.1.1 Tests 101 to 104

These tests are basic tests for ontology alignment.

As the system assumes that equal URIs mean equal objects an alignment of an ontology with itself always returns the correct alignments. The alignment with and irrelevant ontology does not return any results. Language generalization or restriction does not affect the results. Our approach is robust enough to cope with these differences. Considering the differences which occur in real world ontology modeling this is a very desirable feature.

2.1.2 Tests 201 to 210

Tests 201 through 210 focus on labels and comments of ontological entities.

The labels are the most important feature to identify an alignment. In fact, everything else can be neglected, if the labels indicate an alignment (e.g. also the comments in Test 203). Vice versa, changed labels do seriously affect the outcomes. As our approach currently does not make use of any dictionaries, this is critical. Small changes as occurring through a different naming convention can be balanced-out (Test 204 is only slightly worse than the ideal result). Synonyms or translations, possibly also with removed comments, lower especially recall considerably (between 0.57 and 0.87). Nevertheless, the structure alignment does find many of the alignments, despite the differing labels. For the mentioned recalls, precision stays between 0.80 and 0.96.

2.1.3 Tests 221 to 247

For all these tests the structure is changed.

However, as the labels remain, alignment is very good. Again, this indicates that labels are the main distinguishing feature. Only smaller irritations result from the differing structures. In specific, more false positives are identified resulting in a precision of in the worst case “only” 0.94. Recall stays above0.97. According to the amount of structure also the processing time changes. Please note that first results are returned almost instantaneously (less than 5 seconds). The times presented in the table represent the total time until the approach stops its search for alignments.

2.1.4 Tests 248 to 266

These tests were the most challenging ones for our approach. Labels and comments had been removed and different structural elements as well.

Precision reaches levels of 0.61 to 0.95. Recall is in the range of 0.18 to 0.55. Unfortunately, the evaluation results did not show a clear tendency of which structural element is most important for our alignment approach. It seems that the structural features can be exchanged to a certain degree. If one feature is missing, evidence is collected from another feature. This is a nice result for our approach, as it indicates that the weighting scheme of the individual features has been assigned correctly. One tendency that could be identified was that with decreasing semantic information the found alignments become sparser. However, most of the identified alignments were correct (see precision).

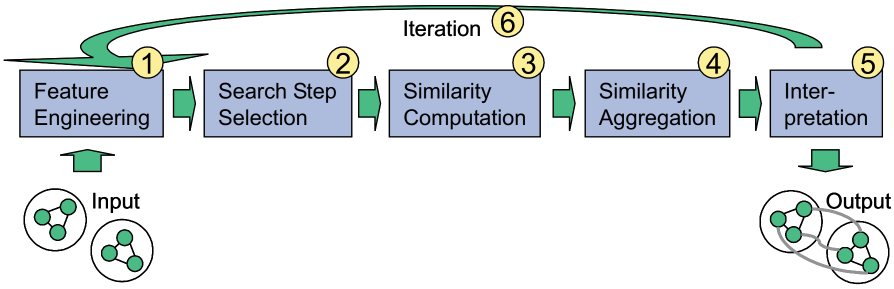


Figure 1: Ontology Alignment Process

We will briefly mention one test for which our approach performed surprisingly well. Ontology 262 has practically everything removed: no labels; no comments; no properties; no hierarchies. Nevertheless, some alignments have been identified. The only information that remained was the links between instances and their classes. By checking whether instance sets were the same (at least in terms of numbers, the instance labels actually differed), some concepts could be correctly aligned.

2.1.5 Tests 301 to 304

Ontologies 301 through 304 represent schemas modeled by other institutions but covering the same domain of bibliographic metadata. From the evaluation perspective, these real world ontologies combine the difficulties of the previous tests.

Especially test case 301 differs both in terms of structure and labels. Its labels generally use the term “has”, i.e. “hasISBN” instead of “ISBN”. This results in a rather low term similarity, as our approach does not split the strings into individual terms. Combined with the differing structure this results in a rather low quality. Also for the other ontologies, both precision and recall do not reach perfect levels. However, the results are satisfactory. In fact, preliminary tests using our semi-automatic approach showed that results could be noticeably increased with very little effort. The question that will partially also be answered by this initiative, is what can maximally be reached. We hope to gain these insights by comparing our results to other participants’ results.

2.2 Directory Ontologies

The directory ontologies are subsumption hierarchies. They could be easily processed. The evaluation results at the workshop will presumably show the following main effects: Subsumption helps to identify some alignments correctly. Our missing usage of dictionaries misses some alignments. As this dataset only uses subsumption, we cannot rely on the more complex ontology features which our approach normally also tries to exploit. Thus, results will not be ideal.

2.3 Anatomy Ontologies

We were very interested in running our ontology alignment on the big real world anatomy ontologies. Especially for our efficient approach, this would have been a deep evaluation. Unfortunately, the ontologies were modeled in OWL-Full. Our approach is based on the KAON2-infrastructure[1] that only allows for OWL-DL. As this interaction is very deep, it was not possible to change to an ontology environment capable of OWL-Full for the contest. We could not run these tests. One result, for us, was the realization that ontologies will probably not stay in the clean world of OWL-DL. We will have to draw consequences from this.

3. GENERAL COMMENTS

3.1 Comments on the results

An objective comment on strengths or weakness requires the comparison with other participants, which will not be available before the workshop. However, some conclusions can be drawn.

Strengths:

- Labels or identifiers are important and help to align most of the entities.
- The structure helps to identify alignments, if the labels are not expressive.
- A more expressive ontology results in better alignments; an argument in favor of ontologies compared to simple classification structures.
- The generally learnt weights have shown very good results.

Weaknesses:

- The approach cannot deal with consequently changed labels. Especially translations, synonyms, or other conventions make it difficult to identify alignments.
- The system is bound to OWL-DL or lesser ontologies.

3.2 Discussions on the way to improve the proposed system

Possible improvements are directly related to the weaknesses in the previous section.

- Extending the handling of labels (strings) can presumably increase overall effectiveness. Usage of dictionaries is widely applied and will be added to our approach as well.
- The tight interconnection of FOAM with KAON2 restricts the open usage of it. Currently efforts are being made to decouple them by inserting a general ontology management layer.

3.3 Comments on the test cases

The benchmark tests have shown very interesting general results on how the alignment approach behaves. These systematic tests are one good underlying test base. For our approach, the directory tests are less interesting, as they are restricted to subsumption hierarchies, rather than complete ontologies. Many of the specific advantages of our approach cannot be applied. It was very unfortunate, that we could not run the anatomy tests. However, we think it is very important to have some real world ontologies, and we hope to test them at a latter point in time.

For future work, it might be interesting to add some user-interaction component to the tests. It would also be interesting to not only have real world ontologies, but also see which alignment approach performs how for specific ontology alignment applications.

3.4 Comments on the measures

Precision and recall are without any doubt the most important measures. Some balancing measure needs to be added as well, as we have done with the f-measure. Otherwise, it is very difficult to draw conclusions on which approach worked best on which test set. For future evaluation it would also be interesting to make use of some less strict evaluation measure, as presented in [9].

4. CONCLUSION

In this paper, we have briefly presented an approach and a tool for ontology alignment and mapping - FOAM. This included the general underlying process. Further, we have mentioned how specific requirements are realized with this tool. We then applied FOAM to the test data. The results were carefully analyzed. We also discussed some future steps for both our own approach and the evaluation of alignments in general.

The main conclusions from the experiments were:

- It is possible to create a good automatic ontology alignment approaches.
- Labels are most important.
- Structure helps, if the labels are not expressive.
- Due to the importance of labels, our approach needs to be extended with e.g. dictionaries in the background.
- One general conclusion from the real world ontologies, was that an ontology system has to be able to also manage OWL-Full, as the real world does not provide the clean ontologies of OWL-DL.

In general, the evaluation has shown us where our specific strengths and weaknesses are, and how we can continue on improving. The results of other participants will give us some further guidelines.

5. REFERENCES

[1] Agrawal, R., Srikant, R.: On integrating catalogs. In: Proceedings of the Tenth International Conference on the World Wide Web (WWW-10), ACM Press (2001) 603–612

[2] Noy, N.F., Musen, M.A.: The PROMPT suite: interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies 59 (2003) 983–1024

[3] Ehrig, M., Haase, P., van Harmelen, F., Siebes, R., Staab, S., Stuckenschmidt, H., Studer, R., Tempich, C.: The SWAP data and metadata model for

[1] http://kaon2..semanticweb.org

semantics-based peer-to-peer systems. In: Proceedings of MATES-2003. First German Conference on Multiagent Technologies. LNAI, Erfurt, Germany, Springer (2003)

[4] Ehrig, M., Staab, S. QOM - quick ontology mapping. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, Proceedings of the Third International Semantic Web Conference (ISWC2004), LNCS, pages 683–696, Hiroshima, Japan, 2004. Springer.

[5] Ehrig, M., Staab, S., Sure, Y. Supervised learning of an ontology alignment process. In Proceedings of the Workshop on IT Tools for Knowledge Management Systems: Applicability, Usability, and Benefits (KMTOOLS) at 3. Konferenz Professionelles Wissensmanagement, Kaiserslautern, Germany, April 2005.

[6] Ehrig, M., Sure, Y. Adaptive Semantic Integration. In Proceedings of the Workshop on Ontologies-based techniques for DataBases and Information Systems at VLDB 2005, Trondheim, Norway, August 2005.

[7] Hughes, T. Information Interpretation and Integration Conference (I3CON) at PerMIS-2004, Gaithersburg, MD, USA, August 2004.

[8] Sure, Y., Corcho, O., Euzenat, J., Hughes, T. (editors), 3rd International Workshop on Evaluation of Ontology based Tools (EON2004), Volume 128, CEUR-WS Publication. Workshop at the 3rd International Semantic Web Conference (ISWC 2004), 7th-11th November 2004, Hiroshima, Japan

[9] Ehrig, M., Euzenat, J. Relaxed Precision and Recall for Ontology Alignment. In Proceedings of the Integrating Ontologies Workshop at K-Cap ’05, Banff, Alberta, Canada, October 2005.

6. RAWRESULTS

6.1 Link to the system and parameters file

The FOAM system may be downloaded at

http://www.aifb.uni-karlsruhe.de/WBS/meh/foam.

The system is continuously improved, so results may slightly differ from the results provided in this paper. The interested reader is encouraged to download, test, and use the system.

6.2 Link to the set of provided alignments (in align format)

The results are also available through the website: http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/results.zip.

6.3 Matrix of results

The following results were achieved in the evaluation runs. As FOAM only allows identifying equality relations, precision and recall only refer to these.

#	Name	Prec.	Rec.	F-measure	Time
101	Reference alignment	1.0	1.0	1.0	2.96
102	Irrelevant ontology	-	-	-	207.14
103	Language generalization	1.0	1.0	1.0	180.95
104	Language restriction	1.0	1.0	1.0	177.63
201	No names	0.90	0.65	0.75	175.99
202	No names, no comments	0.85	0.57	0.68	176.59
203	No comments	1.0	1.0	1.0	174.21
204	Naming conventions	0.96	0.93	0.94	185.09
205	Synonyms	0.80	0.67	0.73	174.46
206	Translation	0.93	0.76	0.84	172.15
207		0.95	0.78	0.86	167.89
208		0.96	0.87	0.92	164.20
209		0.81	0.57	0.67	168.63
210		0.92	0.67	0.77	164.31
221	No specialization	1.0	1.0	1.0	172.92
222	Flattened hierarchy	1.0	1.0	1.0	127.63
223	Expanded hierarchy	0.99	1.0	0.99	142.70
224	No instance	1.0	0.99	0.99	42.09
225	No restrictions	1.0	1.0	1.0	171.13
228	No properties	1.0	1.0	1.0	112.60
230	Flattened classes	0.94	1.0	0.97	137.60
232		1.0	0.99	0.99	45.50
233		1.0	1.0	1.0	110.57
236		1.0	1.0	1.0	12.77
237		1.0	1.0	1.0	87.94
238		1.0	1.0	1.0	106.29
239		0.94	1.0	0.97	73.14
240		0.95	0.97	0.97	84.63
241		1.0	1.0	1.0	11.15
246		0.94	1.0	0.97	51.14
247		0.94	1.0	0.97	70.27
248		0.85	0.48	0.62	251.65
249		0.73	0.46	0.57	150.39
250		0.95	0.55	0.69	114.00
251		0.88	0.41	0.56	132.39
252		0.62	0.34	0.44	145.59

253		0.80	0.44	0.57	83.96
254		0.75	0.18	0.29	103.56
257		0.76	0.48	0.59	28.43
258		0.86	0.39	0.53	133.79
259		0.75	0.45	0.56	149.39
260		0.85	0.38	0.52	71.21
261		0.61	0.33	0.43	82.89

262		0.78	0.21	0.33	21.70
265		0.85	0.38	0.52	70.50
266		0.63	0.36	0.46	81.68
301	BibTeX/MIT	0.78	0.35	0.48	23.43
302	BibTeX/UMBC	0.88	0.74	0.80	21.31
303	Karlsruhe	0.84	0.90	0.87	61.08
304	INRIA	0.94	0.97	0.95	43.32

CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest

Yannis Kalfoglou
Advanced Knowledge Technologies (AKT)
School of Electronics and Computer Science
University of Southampton, UK
y.kalfoglou@ecs.soton.ac.uk

Bo Hu
School of Electronics and Computer Science
University of Southampton, UK
bh@ecs.soton.ac.uk

ABSTRACT
In this results report we summarize our experiences from running the CROSI Mapping System (CMS) over three test cases for this year's OAEI contest: bibliography, Web directories and medical ontologies alignment case studies. CMS successfully parsed and aligned all input ontologies in all three case studies. We also elaborate on the insights gained and potential research directions towards building more robust alignment systems to cope with the increasing diversity of alignment requirements.

1. PRESENTATION OF THE SYSTEM
The CROSI Mapping System (hereafter, CMS) has been developed in the context of the CROSI project (which stands for Capturing Representing and Operationalising Semantic Iinteroperability). CROSI, which is funded by HP, started in November of 2004 and will run until November of 2005[1]. It aims to develop a systematic approach upon which semantic interoperability can be studied and operationalised by (a) capturing and exposing semantics, (b) codify them in Knowledge Representation formats, and (c) operationalise them for the benefit of integration. One of the CROSI deliverables that we used in the early stages of the CMS design, was the notion of *semantic intensity spectrum*[2] which helped us identify what kind of tools and algorithms we could employ in CMS for the OAEI contest. These were used in a modular architecture we devised, reminiscent of similar architectures proposed in the past (see, for example, [4]), which we depict schematically in figure 1

[1]more can be found at: `www.aktors.org\crosi`
[2]more on: `www.aktors.org\crosi\si-spectrum`

In the core of this architecture lies the CMS system. CMS is a structure matching system that capitalizes on the rich semantics of the OWL constructs found in source ontologies and on its modular architecture that allows the system to consult external linguistic resources.

Most of these resources use various families of algorithms which aim to compute similarity based on string distance, e.g. SecondString packages [1]. String distance is one of the widely used techniques in finding correspondences between ontologies. It normally takes as input the names of two concepts calculating the distance, by editing the distance in its simplest form or hybrid distance functions in a more sophisticated form, and output a numeric value to represent the confidence of the similarity. Sometimes, natural language processing methods are employed to cut down the number of string tokens that need to compute the similarity for.

However, string similarity is not sufficient to capture the subtle differences between classes with similar names but different meanings and it can produce misleading results. To alleviate the situation we can work with Natural Language Processing (NLP) packages that exploit synonymy at the: 1) lexical-level, e.g. the use of WordNet [3] to provide a source of synonyms, hypernyms and hyponyms; the 2) phrase- and sentence-level, e.g. phrases and sentences in the active and passive forms but having the same meanings; and the 3) semantic-level synonymy.

Although WordNet-based approaches equip themselves with the lexical synonymy of the names of classes, they do not have the right measure to capture the structural information that is conveyed in most taxonomies. Structural information is exploited in different ways. Heuristic rules is the most common way to take structures into account, e.g. identifying similarity of two entities based on the status of their parents and siblings.

The modular architecture depicted in figure 1 employs a multi-strategy system comprising of four modules, namely, *Feature Generation*, *Feature Selection and Pro-*

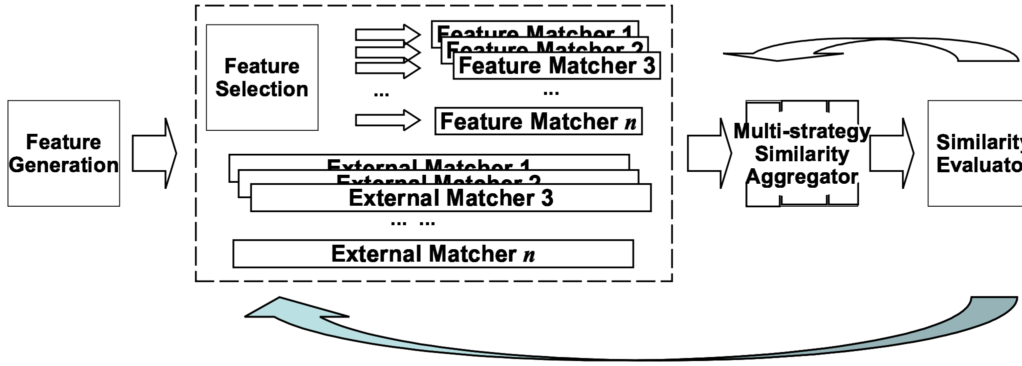


Figure 1: CMS ontology alignment system and its modular architecture.

cessing, *Aggregator* and *Evaluator*. In this system, different features of the input data are generated and selected to fire off different sorts of feature matchers. The resultant similarity values are compiled by multiple similarity aggregators running in parallel or consecutive order. The overall similarity is then evaluated to initiate iterations that backtrack to different stages.

CMS, is an instantiation of such a system. We include a screenshot of the Web-based interface of CMS in figure 2. The system is still under development and we only used the first two components, *Feature generation* and *Feature Selection and Processing*, for aligning the ontologies in the three case studies of the OAEI contest. The alignment algorithms and techniques used are described in later sections but first we elaborate, in the next section, on the purpose of CMS and highlight some of its key characteristics, like the robust features extraction module.

1.1 State, purpose, general statement

The process of ontology mapping (or alignment), can be summarised as: given two ontologies, a system measures the similarity of the source ontological entities against the target ones and produces a list of correspondences, i.e. `mapping` : $O_s, O_t \rightarrow \mathcal{C}_s \times \mathcal{C}_t \cup \mathcal{P}_s \times \mathcal{P}_t \cup \mathcal{I}_s \times \mathcal{I}_t$ where O_i is the input ontologies with $i \in \{s,\ t\}$, subscript s indicating the source and t indicating the target, $\mathcal{C}_i$ the set of classes, $\mathcal{P}_i$ the set of properties and $\mathcal{I}_i$ the set of instances. Hence, the first step when deploying CMS was to extract characteristics that can be used to identify similar entities from different ontologies. We summarize the characteristics we extracted in table 1.

There are several points that need further explanation. First, in many cases, identifying corresponding instances is considered to be an easier task than identifying corresponding classes. This is because instances are expected to have more grounded variables. Corresponding instances provide a ground on which the number of candidate mapping classes can be narrowed down to a few (as we discovered in our past work with the IF-Map instance-based system [?]). Second, in case of complement classes, let c_s be a class from the source ontology and c_t from the target ontology, if $sim(c_s, c_t) = a$ and $d = \neg c$, we can safely conclude that $sim(d, c_s) = 1 - a$, where $sim/2$ is the similarity function and a, a real number, gives the confident value.

1.2 Specific techniques used

To fit the requirements of different applications, we developed and implemented a series of mapping techniques, which are regarded as independent components that made up the CMS.

Name matchers

Ranging from pure syntactical approaches to more semantic enriched ones, name matchers are categorised as: String (tokenised) distance, Thesaurus, and WordNet hierarchical distance.

Levenstain distance is the simplest implementation of string distance. More sophisticated ones are: *Monge-Elkan* distance optimises edit-distance functions with well-tuned editing cost and *Jaro* Metric and its variants computes an accumulated similarity of s and t from the order and number of common characters between s and t, just to name a few. In our system thesaurus comes into play in two forms: WordNet[3] and a predefined corpora that are implemented as `WNNameMatcher` and `CorpusNameMatcher` respectively. To facilitate the use of WordNet, we assume that the local names of classes are either nouns or noun phrases while the local names of properties are phrases starting with verbs followed by either nouns or adjectives. Elements in the retrieved synsets are then compared against each other using either exact string matching or one of the string-distance based algorithms discussed in the previous section. WordNet arranges it entries in hierarchical structures. Hence, the similarity between names can be computed as followings: let w_i and w_j be the corresponding WordNet entries of $name_i$ and $name_j$, w

[3] http://wordnet.princeton.edu/

Local features	
class *labels* and *URIs*	classes with same local names but different name spaces need to be treated with caution, as there is a risk that they might be different in different contexts.
equivalent classes	equivalent classes give the alternatives of a class that can be regarded as hints for identifying new mapping candidates.
related *property names*	both declared and inherited properties contribute to the meaning of a class and thus should be extracted.
complement classes	complement classes indicates semantic dissimilarity.
property *labels* and *URIs*	same as for classes.
property *domain* and *range*	the domain and range of a property can pin down the meaning of a class when name matching is not sufficient.
inverse (transitive) property	both inverse and transitive properties are regarded as hints for similar properties and thus indirect hints for similar classes.
functional property	functional properties play the same role in identifying corresponding classes as *keys* do in element level database schema matching.
instance *labels* and *URIs*	same as for classes.
instantiated classes	instances are treated as a source of understanding semantics.
comments	well documented design rationale is a reliable source for revealing semantics.
Global features	
super and *sub classes*	subsumption relationship help to identify the location of a class in the taxonomy and thus capture the structural semantics.
sibling classes	sibling classes provide the hint of how the parent class is defined.
super and *sub properties*	properties' hierarchy is useful in matching both properties and classes
disjoint classes	disjoint cover should be treated as a special case.
comments	comments sometimes are also given at the global level.
version information	the record of modifications and authentication provides alternatives.

Table 1: Features extracted for Ontology Mapping.

be the least common hypernym of w_i and w_j, r be the root of the underlying WordNet hierarchy, and h_i, h_j, h be the distances between w_i and r, w_j and r, w and r, respectively, the similarity between w_i and w_j is approximated as $2 \times h/h_i + h_j$.

Semantic matchers

In CMS, the flavour of semantic is added in two different ways: namely structure-aware matchers and intension-aware matchers.

Structure-awareness refers to the capability of traversing class hierarchies and accumulating similarities along the sub-class (sub-property) relationships. Let c and d be two classes from source and target ontologies, c_i and d_i are their direct parents in respective ontologies, the similarity between c and d is recursively defined as $\texttt{sim}(c, d) = \alpha\texttt{sim}_{local}(c, d) + \beta\texttt{sim}(c_i, d_i)$, where α and β are arbitrary weights and $\texttt{sim}_{\text{local}}/2$ gives the local similarity with regard to c and d which can be computed using one or a combination of techniques discussed above.

Intension-awareness takes into account the definitions of classes. A class c are regarded as a tuple $\langle S, P\rangle$ where S is a set of classes of which c is a subclass and P is a set of properties having c as the domain and other classes or concrete data types as the range. Hence, finding the semantic similarity between $c = \langle S_c, P_c\rangle$ and $d = \langle S_d, P_d\rangle$ amounts to finding the similarity between S_c and S_d as well as P_c and P_d, i.e. $\texttt{sim}(c, d) = \alpha\texttt{sim}(S_c, S_d) + \beta\texttt{sim}_{property}(P_c, P_d)$, where α and β are arbitrary weights and $\texttt{sim}_{property}/2$ computes the property similarity. More specifically, we differentiate the following situations:

- classes with matching property names, property domains and property ranges: $\mathcal{L}_{p_c} = \mathcal{L}_{p_d}$ and $sim_{set}(\Delta_{p_c}, \Delta_{p_d}) \geq v$ and $sim_{set}(\Phi_{p_c}, \Phi_{p_d}) \geq v$ where $sim_{set}/2$ computes the similarity of two sets of entities and v is a predefined threshold.
- classes with matching property names and property domains but different property ranges: $\mathcal{L}_{p_c} = \mathcal{L}_{p_d}$ and $sim_{set}(\Delta_{p_d}, \Delta_{p_d}) \geq v$, $sim_{set}(\Phi_{p_c}, \Phi_{p_d}) < v$, and
- classes with matching property names but different property domains as well as ranges: $\mathcal{L}_{p_c} = \mathcal{L}_{p_d}$ and $sim_{set}(\Delta_{p_c}, \Delta_{p_d}) < v$ and $sim_{set}(\Phi_{p_c}, \Phi_{p_d}) < v$.

The first situation contributes the most to the similarity of c and d. We regard classes with matching names and exact matching properties, i.e., properties with same name, domain and range, as semantically equivalent classes.

In many cases, matching between Δ_{P_c} and Δ_{P_d} (Φ_{P_c} and Φ_{P_c}, respectively) can only be concluded after traversing several levels upwards or downwards the class hierarchy. Although not as strong as exact matching of property domains and ranges, matching classes of Δ_{P_c} (Φ_{P_c}) to remote ancestors or descendants of classes of Δ_{P_d} (Φ_{P_d}) provides a hint on how close the different properties are, and thus how similar the two concepts c and d are. Such an idea is implemented in our system as a `ClassDefPlusMatcher` method.

1.3 Adaptations made for the contest

We didn't do any major adaptations to CMS in order to align the OAEI contest ontologies. We only did minor, routine programmatic adjustments, as for example running the CMS system from the command line prompt in a batch mode to parse and align the hundreds of ontologies in the Web directories case or include specific Java heap size adjustment flags in order to run the system over the vast FMA ontology. Other than that, the system ran as normal.

2. RESULTS

CMS benefits from the plug and play of modular matchers. In this contest, four different matchers were used, namely `ClassDef` for examining the domain and range of properties associated with classes, `CanoName` for accumulating similarities among class hierarchies, `WNDisSim` for computing the distance between two class names based on WordNet structures and `HierarchyDisSim` for distributing similarity among class hierarchies. The four major matchers were invoked both in parallel and sequentially. When invoked in parallel their results were then aggregated as weight average. On the other hand, when invoked in sequence, `CanoName` and `WNDisSim` give a list of corresponding classes whose similarities were then refined by `ClassDef` and `HierarchyDisSim`. CMS ran each test case with different configurations (combination and sequencing) of the aforementioned four mapping modules and precision and recall values were calculated for each run. In this report, we include the the configurations with the highest precision and recall values.

2.1 Case 1: benchmark/BibTex ontologies

For all the ontologies in this case we used a threshold of 0.8.

ontology 202: CMS fails to produce any mapping candidates with high similarity score in test case 202 due to the naming convention. We consider class names as the foundation on which other techniques can be applied (although not the sole and dominant clue for finding mapping candidates). Similarly, cases 248 to 266 also fall into this category: no candidates with high similarity value were found.

ontology 205: CMS does not achieve a high recall rate for benchmark test case 205 due to the restriction of WordNet. In case 205, class names are replaced by randomly selected synonyms. CMS relies heavily on external resources, e.g. WordNet, to provide lexical alternatives for class and property names and thus fails to respond well for synonyms that are not recognised by WordNet. A customised corpus might alleviate the problem and improve the performance with significant efforts and domain expertise.

ontology 301: In test case 301, smaller similarity scores were assigned to mapping candidates. This is due to the fact that although classes have similar names, they are defined with different properties which have different names, domains and/or ranges. It is our contention that for classes restricted with different properties, they should either not be considered as equivalent classes or their similarity value should be reduced to reflect such difference.

2.2 Case 2: Web directories ontologies

We do not have any specific comments for Case 2. All 2265 were parsed successfully by CMS and fetched for alignment. However, 29 ontologies did not produced any alignment results due to circular definitions in the original `source.owl` and `target.owl` files. So, a total of 2236 pairs of `source.owl`/`target.owl` were aligned. The system parsed them from the command line in a batch mode, and the results produced after 2 hours and 53 minutes. Each cycle involved reading and parsing the source and target ontologies, find alignments (if any) and save and write the results in the common alignment format in a file. This was repeated 2265 times.

2.3 Case 3: Medical ontologies

This case was the most interesting. The sheer size of the input ontologies (especially that of FMA), the modelling style of OWL, the conventions used, and the complexity of the paradigm made it an interesting adventure from the research point of view. We report in more detail about our experiences in section 3.3.

3. GENERAL COMMENTS

Performance tuning and hardware settings: As we were facing some really large ontologies (i.e., the 72k classes FMA ontology), we had to do certain optimizations to the code and to the computer settings in order to obtain alignment results in acceptable time. We ran the tests on a stand-alone PC running Microsoft Windows XP operating system, service pack II, 2003 version. The PC had 1GB of memory installed (DDR400-SDRAM), an 80GB Serial ATA hard disk, and a Pen-

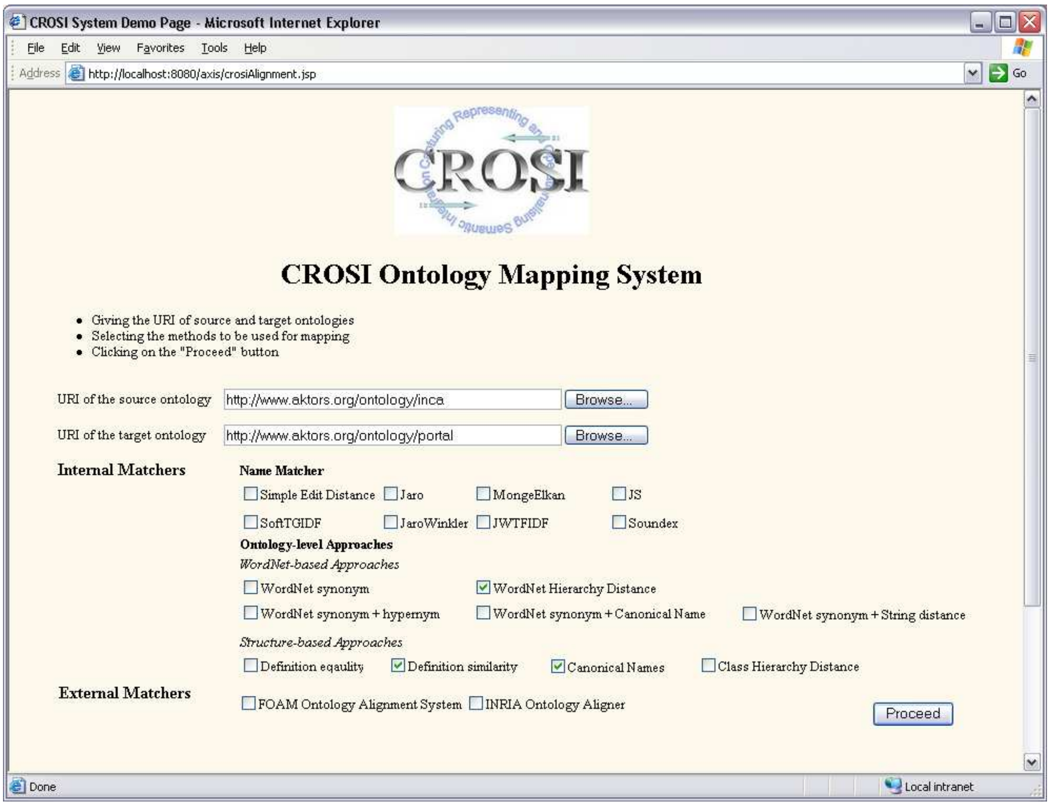


Figure 2: Web-based Interface for CMS.

tium 4, 3.0GHz processor. We used Java VM (version 1.5.0_04) and we had to do certain configurations to adjust the heap size in Java. For example, the standard Java heap size is 64MB. This was not enough though for the Web directory and medical ontologies case. In fact, for the medical ontologies case, the sheer size of the input ontologies (especially that of FMA) forced us to use a 768MB heap size. Settings lower than this threshold caused the system to run out of memory.

Parsing and extracting experiences: FMA owl is a 31MB .owl file comprising of 72545 declarations of owl classes and 100 relations (object and data type properties). These numbers were obtained when using our Jena 2.2 API and probably deviate slightly from other parsers. Parsing and extracting features from the FMA ontology took 9 minutes and 17 seconds with Java Heap Size adjusted to 512MB. However, in order to run the CMS and find alignments with the OpenGALEN we had to use a 768MB heap size setting. While parsing, Jena API was complaining about the syntax idioms used. For example we had a lot of warnings from Jena's RDF syntax handler, or the form "bad URI in qname XXX: no scheme found". We elaborate on the reasons behind this parsing warnings in section 3.3.

OpenGALEN.owl is a 4MB .owl file comprising of 24 declarations of owl classes and 30 relations (as previously, object and data type properties, and these numbers were obtained from Jena 2.2 API). Parsing and extracting features from OpenGALEN took just a few seconds. There was no need to adjust the Java heap size.

3.1 Comments on the results

Different combinations of CMS plug-in matchers perform significantly differently due to the nature of benchmark test cases. Table 3.1 lists the choice of matchers with regard to each test cases while Table 3.2 shows performance values of different matchers[4] with regard to alignment of ontology 303 in case 1, in terms of precision and recall.

3.2 Discussions on the way to improve the proposed system

CMS is expected to be improved on the following aspects: a more sophisticated aggregation mechanism, a unified alignment representation formalism, and parameterised algorithms for class hierarchy distance.

Firstly, as discussed in previous sections, results from multi-matchers are aggregated as weighted average with arbitrary weights to start with. Thus far, the weights are fine-tuned manually relying on the knowledge of the

[4] Results are obtained with equal weights for matchers.

CMS Matchers	Test Case #
A	103, 201, 210,
A, B	205, 206, 207, 209, 301, 303
A, C, D	225, 228, 233, 236, 239-241, 246, 247, 248-266, 302
A, B, C, D	104, 203, 204, 208, 221, 222, 223, 224, 230, 231, 232, 237, 238, 304

A–Class Definition,
B–Canonical Name,
C–WordNet Hierarchy Distance,
D–Class Hierarchy Distance

Table 2: CMS matchers combinations.

CMS Matchers for #303	Precision	Recall
Class Definition (A)	0.6923	0.4736
Canonical Name (B)	0.3243	0.6315
WordNet (WN) synonym	0.06	0.7894
WN Hierarchy Dis (C)	0.24	0.3157
Class Hierarchy Dis (D)	1.0	0.5263
WN synonym + hypernym (E)	0.04	0.8421
A + B	0.9	0.4736
A + E	1.0	0.4736
A + B + E	1.0	0.4736
A + B + D	1.0	0.3684
B + C + D	0.8	0.4210
B + C + D	0.8	0.4210

Table 3: Performance of different matchers for test case #303.

domain of discourse and the underlying algorithms of CMS. A more sophisticated approach would hire machine learning techniques to work out the most appropriate weights with regard to different matchers aiming to solve different sort of mappings. Furthermore, results from different matchers can be sorted locally first which could make accumulating results from different matchers to be reduced to ranking aggregation [2].

Secondly, the heterogeneous nature of different matchers – some external matchers produce pairwise equivalence with numeric values stating the similarity score while others output high level relationships, e.g. *same entity as*, *more specific than*, *more general than* and *disjoint with* expressed in high level languages such as OWL and RDF – suggests that output from different matchers has to be lifted to the same syntactical and semantic level. A unified representation formalism equipped with both numeric and abstract expressivity can facilitate the aggregation of heterogeneous matchers.

Thirdly, CMS takes into account the exact position of classes in the class hierarchy. We would like to develop algorithms that penalise mapping candidates that are found to be quite apart from each other, and then propagate their similarity values upwards and downwards in the hierarchy to their descendants and/or ancestors. There could also be pre-defined parameters that as we go up or down the hierarchy we change the similarity values of their descendants and/or ancestors accordingly. We expect that this could reduce the number of false positive results.

3.3 Comments on the test cases

We do not have any specific comments for test cases on BibTex and Web directories alignments. However, we found interesting the last test case, that of medical ontologies alignment, and we summarize our experiences below.

FMA.owl was a different case altogether. The ontology describes the domain of human anatomy and it aims to provide "a reference ontology in biomedical informatics for correlating different views of anatomy, aligning existing and emerging ontologies in bioinformatics" [6]. However, there are two notable facts regarding the syntactic and modelling idioms of FMA and existing results from previous efforts in trying to align FMA and GALEN. As far as the former is concerned, the OWL version we had to work with was a result of translation from Protege. Previous work has shown that this result is not always a faithful representation of the original FMA Protege model. For instance, it has been reported that FMA DL constructs are often ill-defined and they lead to inconsistencies when a reasoner parses the ontology [5]. Consistency checking for FMA is an acknowledged problem though, even by its authors: "[...] feedback from these investigators revealed an aggregate of a few hundred errors, many of which related to spelling and only a few to cycles in the class subsumption and partonomy hierarchies." [6].

Leaving aside this fact of life (as it is natural for an ontology that big and so close to human practice to be inconsistent), we point to a couple of syntactic idioms that we found interesting when parsing the ontology with our Jena-based CMS system. Firstly, the rather unusual use of unique frame IDs for class names (`<owl:Class rdf:ID>` constructs) and the textual description of a class in an `rdfs:label` construct. We also noticed some unusual uses of references to frame IDs. For instance, the declaration of "arterial supply" as an object property: `<owl:ObjectProperty rdf:ID="arterial_supply" rdfs:label="arterial supply">` is used in other parts of the ontology where it refers to a `rdf:resource` which points to a different resource:
`<arterial_supply rdf:resource="#frame_14586"/>`. Tracing that frame ID leads us to a definition of a "Tissue" class, and not the "arterial supply": `<owl:Class rdf:ID="frame_14586" rdfs:label="Tissue">`. The definition of an instance (with frame ID 14586) of an ob-

ject property ("arterial supply") that is a class ("Tissue") could lead to modelling misunderstandings and confusion (although, syntactically speaking, it is allowed in some versions of OWL).

Going back to our argument for the notable facts, we found that previous efforts for aligning FMA to GALEN reported rather controversial results. For example, in [7], the authors employed two different alignment methods to map FMA to GALEN. Despite of the subtle differences of OpenGALEN with GALEN, the similarity of their work with that of the OAEI contest 3rd case study is high but some of their findings are questionable from the semantics point of view: for example, it was reported that "Pancreas" in FMA matches "Pancreas" in OpenGALEN with 1.0 similarity value which "indicates a perfect match" [7]. When we looked carefully at the definitions of "Pancreas" in both ontologies we saw that "Pancreas" is defined as a class in FMA (<owl:Class rdf:ID="frame_12280" rdfs:label="Pancreas">) whereas in GALEN (OpenGALEN) as an instance of class "Body Cavity Anatomy"

```
<owl:Class rdf:ID="Body_Cavity_Anatomy">
<rdfs:subClassOf
rdf:resource="#OpenGALEN_Anatomy_Metaclass"/>
<Body_Cavity_Anatomy rdf:ID="Pancreas">
```

Even if OWL semantics allow to map an individual to a class (when dealing with OWL Full), such an alignment is misleading especially when we consider the high level of abstraction for the "Pancreas" class in OpenGALEN. It seems that the "lexical phase" parsing used in [7] was the main contributor to this high similarity value when relatively little structure information was taken into account. As a final comment on the case, we also point the reader to observations made by the FMA authors when trying to validate mapping results and differences in terminologies with these two ontologies: "[...]the reasons for the differences have not yet been explored, but at least some of them may be the different contexts of modelling. GALEN represents anatomy in the context of surgical procedures, whereas FMA has a strictly structural orientation." [6].

3.4 Comments on the measures

The proposed measures of precision and recall have been studied and practiced in the NLP community for years and they are a *de facto* standard metric for commercial applications, like search engines. However, we believe that their adaptation for measuring the performance of an ontology mapping system is somewhat questionable. We cannot elaborate fully on our reservations regarding the use of such a metric in this short paper, but we highlight the main points of our objections: (a) precision is regarded as hard to implement and reveals the usefulness of a retrieved document (or hit in a hitlist) for a search engine. We can't judge the usefulness of a found alignment by comparing it with the reference alignment; (b) neither precision nor recall take into account the possible applications of the alignments found. In all the past EON (and this year OAEI) contests, a set of pre-defined alignments were used as a standard against which all found alignment were compared. This does not say anything about the usefulness of the found alignments, or even of they are complete as the pre-defined ones can be erroneous. Further to these comments, we would also like to add that the assignment of numerical values in the range 0.0 to 1.0 does not reveal their semantic relevance, but purely a brute-force algorithmic way of comparing performance. We also observed a variety of interpretations of precision and recall metrics by the ontology alignment community.

3.5 Proposed new measures

Devising new measures for assessing the found alignments between two ontologies in a universally agreed manner is a difficult task. We do not see a quick solution to this problem, but as ontology engineers we can apply knowledge engineering technologies that encompass as much semantic information as possible; for example, we were surprised that the semantically rich definitions of OWL for declaring class or property equality (and inequality) and the universal construct for declaring similarity, are hardly used by the community.

We would also like to see ways of introducing "application-driven" alignment metrics where an example application (i.e., a Semantic Web service information lookup engine) will need to access two different ontologies and the alignments found will need to be used in the application in a specific way. Having an application-driven alignment metric, we can experiment with the notion of usefulness of alignment in a real world scenario, rather than doing meaningless number crunching with regard to found and pre-defined alignments. After all, alignment needs to be done in the first place because there is a real world need for it.

4. CONCLUSION

The 2005 OAEI ontology alignment contest was the first one that introduced sizeable ontologies and posed some interesting and challenging problems with respect to performance, scaling and domain exploration. We found it a rewarding experience and we look forward to continue the fruitful exploration of this key field in the emergent Semantic Web.

5. REFERENCES

[1] W.W. Cohen, P. Ravikumar, and S.E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IJCAI 2003 IIWeb Workshop*, pages 73–78, 2003.

[2] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the ACM SIGMOD*

International Conference on Management of Data, pages 301–312. ACM Press, 2003.

[3] C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[4] M. Ehrig and S. Staab. QOM - Quick Ontology Mapping. In *Proceedings of the 3rd International Semantic Web Confernece (ISWC'04), LNCS 3298, Hiroshima, Japan*, page 683–697, 2004.

[5] C. Golbreich, S. Zhang, and O. Bodenreider. Migrating the FMA from Protege to OWL. Technical report, jul 2005. In notes of the 8th International Protege Conference, Madrid, Spain.

[6] C. Rosse and JL. Mejino. A Reference Ontology for Bioinformatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36:478–500, 2003.

[7] S. Zhang, P. Mork, and O. Bodenreider. Lessons learned from aligning two representations of anatomy. In *in Proceedings of the KR 2004 Workshop on Formal Biomedical Knowledge Representation, Whistler, BC, Canada*, pages 102–108, 2004.

6. RAW RESULTS

All of our results are included in a tabular format in table 6.3. These results have been the best of the CMS combinations with different matcher. We report on those in section 3.1. So, for example, alignments for case #103 were produced using CMS Matcher A, whereas alignments for case 225 were produced using CMS Matchers A+B+C. A list of all this combibnation can be found in table 3.2.

6.1 Link to the system and parameters file

Access to the Web-based interface of the CMS system is provided via `www.aktors.org/crosi/cms`. We note that the system is not available in the community for free distribution yet, due to the legalities of the IPR for the CROSI project.

6.2 Link to the set of provided alignments (in align format)

The results of all three cases (BibTex, Web directories, Medical) are available for download from the CROSI web site at `www.aktors.org/crosi/eon05contest/results`.

6.3 Matrix of results

#	Name	Prec.	Rec.	Time (s)
101	Reference alignment	N/A	N/A	N/A
102	Irrelevant ontology	N/A	N/A	108
103	Language generalization	1.0	0.788	88
104	Language restriction	1.0	0.788	159
201	No names	1.0	0.189	70
202	No names, no comments	N/A	N/A	
203	No comments	1.0	0.697	147
204	Naming conventions	1.0	0.605	153
205	Synonyms	1.0	0.230	85
206	Translation	1.0	0.255	82
207		1.0	0.264	88
208		1.0	0.473	149
209		1.0	0.103	84
210		0.818	0.246	74
221	No specialisation	1.0	0.788	129
222	Flatenned hierarchy	1.0	0.724	169
223	Expanded hierarchy	0.962	0.758	316
224	No instance	1.0	0.788	151
225	No restrictions	0.788	0.788	85
228	No properties	0.788	0.788	76
230	Flattened classes	1.0	0.760	161
231	Expanded classes	1.0	0.788	145
232		1.0	0.788	118
233		0.838	0.788	70
236		0.788	0.788	77
237		1.0	0.724	156
238		0.961	0.757	315
239		0.766	0.793	220
240		0.757	0.757	221
241		0.838	0.788	70
246		0.766	0.793	70
247		0.757	0.757	221
301	Real: BibTeX/MIT	1.0	0.363	30
302	Real: BibTeX/UMBC	1.0	0.348	31
303	Real: Karlsruhe	1.0	0.474	328
304	Real: INRIA	0.85	0.566	131

Acknowledgements

This work is supported under the Capturing, Representing, and Operationalising Semantic Integration (CROSI) project which is sponsored by Hewlett Packard Laboratories at Bristol, UK. The first author is also supported by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC) project which is sponsored by the UK EPSRC under Grant number GR/N15764/01.

Falcon-AO: Aligning Ontologies with Falcon

Ningsheng Jian, Wei Hu, Gong Cheng, Yuzhong Qu

Department of Computer Science and Engineering
Southeast University
Nanjing 210096, P. R. China

{nsjian, whu, gcheng, yzqu}@seu.edu.cn

ABSTRACT

Falcon-AO is an automatic tool for aligning ontologies. There are two matchers integrated in Falcon-AO: one is a matcher based on linguistic matching for ontologies, called LMO; the other is a matcher based on graph matching for ontologies, called GMO. In Falcon-AO, GMO takes the alignments generated by LMO as external input and outputs additional alignments. Reliable alignments are gained through LMO as well as GMO according to the concept of reliability. The reliability is obtained by observing the linguistic comparability and structural comparability of the two ontologies being compared. We have performed Falcon-AO on tests provided by OAEI 2005 campaign and got some preliminary results. In this paper, we describe the architecture and techniques of Falcon-AO in brief and present our results in more details. Finally, comments about test cases and lessons learnt from the campaign will be presented.

Categories and Subject Descriptors

D.2.12 [**Software**]: Interoperability; I.2.6 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods; I.5.3 [**Pattern Recognition**]: Clustering—*Similarity measures*

General Terms

Experimentation, Measurement

Keywords

Semantic Web, Ontology Alignment, Mapping, Matching, Similarity Measurement

1. PRESENTATION OF THE SYSTEM

As an infrastructure for semantic web applications, Falcon is a vision of our research group. It will provide enabling technologies for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is still under development in our group. As a component of Falcon, Falcon-AO is an automatic tool for aligning ontologies. It is dedicated to aligning web ontologies expressed in OWL DL [5]. There are two matchers integrated in current version of Falcon-AO (version 0.3). One is a matcher based on linguistic matching for ontologies, called LMO, and the other one is a matcher based on graph matching for ontologies, called GMO.

1.1 Linguistic Matching for Ontologies

As is known, linguistic matching plays an important role in matching process. Generally, linguistic similarity between two entities relies on their names, labels, comments and some other descriptions.

LMO combines two different approaches to gain linguistic similarities: one is based on lexical comparison; the other is based on statistic analysis.

In lexical comparison, we calculate the edit distance [4] between names of two entities and use the following function to capture the string similarity (denoted by *SS*):

$$SS = 1/e^{\frac{ed}{|s1.len+s2.len-ed|}} \qquad (1)$$

Where *ed* denotes the edit distance between *s1* and *s2*; *s1.len* and *s2.len* denote the length of the input strings *s1* and *s2*, respectively.

In statistic analysis, we use the algorithm of VSM [6] (Vector Space Model) in our implementation. Given a collection of documents, we denote N the number of unique terms in the collection. In VSM, we represent each document as a vector in an N-dimensional space. The components of the vector are the term weights assigned to that document by the term weighting function for each of the N unique terms. Clearly, most of these are going to be 0, since only a few of the N terms actually appear in any given document. In our scenario, we construct a virtual document for each of the ontology entities (classes, properties and instances). The virtual document of an entity consists of "bag of terms" extracted from the entity's names, labels and comments as well as the ones from all neighbors of this entity. The term weighting functions are defined as follows:

$$TermWeighting = TF * IDF \quad (2)$$

$$TF = \frac{t}{T} \quad (3)$$

$$IDF = \frac{1}{2} * (1 + log_2 \frac{D}{d}) \quad (4)$$

In equation (3), t denotes the number of times where one term occurs in a given document and T denotes the maximum number of times. In equation (4), D denotes the number of documents in collection and d denotes the number of documents where the given term occurs at least once.

We can gain the cosine similarity between documents (denoted by DS) by taking the vectors' dot product:

$$DS = N \cdot N^t \quad (5)$$

It is worthy of note that there are several preparing steps before calculating term weights, such as splitting words, stemming and removing stop words.

The two methods described above will both take effect in ontology matching. In our implementation, we combine them together, and use the following equation to calculate the final linguistic similarity. Please note that the parameters in the equation comes from our experience:

$$LinguisticSimilarity = 0.8 * DS + 0.2 * SS \quad (6)$$

Currently, we do not use any lexicons in LMO and it is certain that the use of lexicons may bring some benefits for matching. We plan to take into account using some lexicons in later versions.

1.2 Graph Matching for Ontologies

Another important component in Falcon-AO is GMO, which is based on a graph matching approach for ontologies. It uses directed bipartite graphs to represent ontologies and measures the structural similarity between graphs by a new measurement. Details of the approach are described in another paper [3] also presented in the K-Cap 2005 Workshop on Integrating Ontologies [1].

The main idea of GMO is as follows. Similarity of two entities from two ontologies comes from the accumulation of similarities of involved statements (triples) taking the two entities as the same role (subject, predicate, object) in the triples, while the similarity of two statements comes from the accumulation of similarities of involved entities of the same role in the two statements being compared.

Usually, GMO takes a set of matched entity pairs, which are typically found previously by other approaches, as external mapping input in the matching process, and outputs additional matched entity pairs by comparing the structural similarity.

Our previous experiments showed that GMO were irreplaceable when there was little gain from lexical comparison. In addition, GMO can be integrated with other matchers. While using GMO approach to align ontologies, there should be another component to evaluate reliability of alignments generated by GMO.

1.3 Linguistic vs. Structural Comparability

Given two ontologies to be aligned, GMO always tries to find all the possible matched entity pairs. However, how to evaluate the reliability of these matched entity pairs is still a problem. As mentioned above, another component is needed to select more reliable matched entity pairs by using other information. In Falcon-AO, we use a simple approach to observe the reliability of matched entity pairs output by GMO, and select more reliable matched entity pairs to the users. The approach is based on the measure of linguistic comparability (LC) and structural comparability (SC) of two ontologies to be aligned.

Given two ontologies O_1, O_2 to be aligned, the linguistic comparability *(LC)* of O_1 and O_2 is defined as follows:

$$LC = \frac{M}{\sqrt{N_{O_1} * N_{O_2}}} \quad (7)$$

Where M denotes the number of entity pairs with similarity larger than c and c is an experience value; N_{O_1} and N_{O_2} represent the number of named entities in O_1 and O_2, respectively.

The structural comparability is determined through comparing the occurrences of built-in properties used in the two ontologies to be aligned. The built-in properties are RDF [2], RDFS [1] and OWL [5] built-in vocabularies used as properties in triples (e.g. rdf:type, rdfs:subClassOf and owl:onProperty).

[1] http://km.aifb.uni-karlsruhe.de/ws/intont2005

We use VSM method to observe the structural comparability. The vectors V_1, V_2 represent the frequency of built-in properties used in O_1 and O_2 and the element v_{ij} denotes the number of occurrence of built-in property p_j in O_i. The structural comparability of O_1 and O_2 is the cosine similarity [7] of V_1 and V_2:

$$SC = \frac{V_1 \cdot V_2}{\|V_1\| \, \|V_2\|} = \frac{\sum_{j=1}^{n} v_{1j} * v_{2j}}{\sqrt{\sum_{j=1}^{n} v_{1j} * v_{1j}} \sqrt{\sum_{j=1}^{n} v_{2j} * v_{2j}}} \quad (8)$$

1.4 Implementation

LMO and GMO are integrated in Falcon-AO. Alignments output by Falcon-AO come from the integration of alignments generated by LMO and GMO. The architecture of Falcon-AO is shown in Figure. 1.

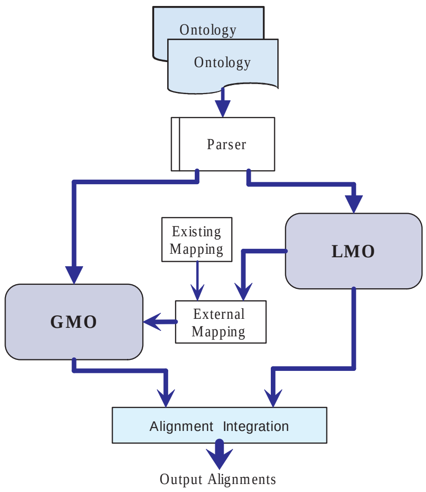


Figure 1: System Architecture

Due to heterogeneous ways in expressing semantics and the inference capability brought from ontology languages, two ontologies being matched may need to be coordinated by removing some redundant axioms from it or adding some inferred axioms. So coordination actions should be taken before using GMO approach. We have integrated several coordination rules in Falcon-AO. Our *Parser* component based on Jena [2] has the functionality of coordinating ontology models.

As is known, given external mapping as input, GMO can find additional mapping. The external mapping is made of two parts: one is the existing mapping preassigned by the system; the other comes from another matcher. The existing mapping is the mapping between built-in vocabularies of web ontology languages, datatypes, data literals and URIs used in both ontologies. And in Falcon-AO we take the alignments generated by LMO as the other part of external mapping. Entities involved in the alignments generated by LMO are set to be external entities and GMO will just output mapping between internal entities.

When the alignments generated by LMO and GMO are obtained, Falcon-AO will integrate these alignments by observing the linguistic comparability and structural comparability, following the rules below:

1. We take that linguistic similarity is somewhat more reliable than structural similarity, and that the alignments generated by LMO are always accepted by Falcon-AO.
2. When the linguistic comparability is high and the structural comparability is low, only alignments generated by GMO with high similarity are reliable and accepted by Falcon-AO.
3. If the linguistic comparability is low, all of the alignments generated by GMO are accepted by Falcon-AO. In this case, there is no enough information to measure these alignments and we can only assume that they are reliable.

Falcon-AO is implemented in Java. The implemented process can be outlined as follows:

1. Input two ontologies and parse them.
2. Run LMO and obtain matched entity pairs.
3. Calculate linguistic comparability and structural comparability.
4. In the case that linguistic comparability is below a very low threshold (e.g. 0.01) and the structural comparability of them is also low, we take that these ontologies are quite different and Falcon-AO exits with no alignment.
5. Set external entities of the ontologies according to the matched entity pairs generated by LMO.
6. Input matched entity pairs generated by LMO into GMO and form external mapping for GMO. In the current version of Falcon-AO, all the individuals of ontologies are specified as external entities and their similarities are computed by LMO.
7. Run GMO and obtain matched entity pairs.
8. Integrate the alignments generated by LMO and GMO following the rules described above.
9. Exit with alignments as output.

[2]http://jena.sourceforge.net/

1.5 Adaptations Made for the Contest

For anatomy test, FMA [3] ontology and OpenGALEN [4] ontology are not OWL DL. In order to make effective use of descriptions of entities, we have manually found some annotation properties and inputted them into LMO. With the help of these annotation properties, Falcon-AO can find about 500 more matched entity pairs in addition to other 2000 matched entity pairs found by a simple version of LMO.

2. RESULTS

In this section we will present the results of alignment experiments on OAEI 2005 campaign. All the alignments output by Falcon-AO are based on the same parameters.

2.1 Systematic Benchmark Test

We divide all the benchmark tests [5] into five groups: test 101-104, test 201-210, test 221-247, test 248-266 and test 301-304. We will report the results of alignment experiments on these five groups in correspondence. The full results on all tests are listed in **section** 6.3.

2.1.1 Test 101–104

In tests 101, 103 and 104, the source ontologies contain classes and properties with exactly the same names as those in the reference ontologies. LMO can easily get all the matched entity pairs, and GMO takes little effect.

In test 102, the linguistic comparability of the two ontologies is nearly zero and the structural comparability is low as well. So it could be concluded that the two ontologies to be aligned are quite different. Falcon-AO exits with no alignment.

The average performance on test 101-104 is shown below:

	Precision	Recall	F-Measure	Time
Average	1.0	1.0	1.0	5s

2.1.2 Test 201–210

We find that each pair of ontologies of these ten tests has high structural comparability, which means that each pair of the ontologies to be aligned is quite similar in structure. Our previous experiments showed that GMO performed well on these tests even without any additional external mapping input. In most tests, LMO just finds a small part of all the matched entity pairs, the rest are generated by GMO. Since GMO runs slower than LMO, it takes Falcon-AO more time to find all matched entity pairs.

[3]http://sig.biostr.washington.edu/projects/fm/
[4]http://www.opengalen.org/
[5]http://oaei.inrialpes.fr/2005/benchmarks/

For test 201, where each of the local name of class and property is replaced by a random one, LMO can still find some matched classes and properties due to the sameness of their labels or comments. With these matched entity pairs as feed, GMO performs well.

In test 202, names of classes and properties are disturbed and their comments are suppressed. LMO can only find little mapping. Meanwhile, Falcon-AO still performs not bad by running GMO. In this test, we find that it is hard to distinguish many properties purely by the structure of the ontology, since they have the same domains and ranges, and never used in other part of the ontologies. Falcon-AO failed to find correct mapping of these properties, which makes the result not so well as test 201.

In test 203, LMO is able to find all the matched entity pairs. Therefore, it just takes Falcon-AO several seconds to find all alignments.

For tests 204 and 208 with naming conventions, both the linguistic comparability and structural comparability are high. The outputs of the integration of LMO and GMO are well.

For the synonym tests 205 and 209, due to the fact that no thesaurus is used in our tool, LMO performs not so well. There are some errors in the outputs of LMO. With these errors feed to GMO, GMO failed to perform well. As a result, the outputs of Falcon-AO may be weaker than the outputs of using GMO independently.

In tests 206, 207 and 210, ontologies to be aligned are expressed in different languages. Falcon-AO does not have a specific matcher that uses a dictionary for word translation. However, because of their high structural comparability, GMO in Falcon-AO performs not bad on these tests.

The average performance on test 201-210 is described below:

	Precision	Recall	F-Measure	Time
Average	0.96	0.95	0.95	63s

2.1.3 Test 221–247

In these tests, the linguistic comparability of each pair of ontologies to be aligned is very high. Most of the alignments are found by LMO and GMO takes little effect. So, it only takes Falcon-AO a few seconds to align them.

As is shown below, the average performance on these tests are perfect.

	Precision	Recall	F-Measure	Time
Average	0.99	1.0	0.99	4s

2.1.4 Test 248–266

These fifteen tests are the most difficult ones in all benchmark tests, since both their linguistic comparability and structural comparability are low. In the case that the linguistic comparability between two given ontologies is very low, Falcon-AO would not call any matchers. However, in these tests, there are still some individuals with the same names, which increase the linguistic comparability. So Falcon-AO will still run GMO integrated with LMO.

Since the ontology pairs to be aligned are quite different both in linguistics and in structure, our outputs are not good (with average F-Measure 0.63). Indeed, in some cases, it is really hard to determine the exact mapping. For these tests, the time for aligning relies on the size of two ontologies.

	Precision	Recall	F-Measure	Time
Average	0.71	0.60	0.63	60s

2.1.5 Real Ontologies Test 301–304

In these tests, each pair of ontologies has high linguistic comparability but low structural comparability. This indicates that the outputs of Falcon-AO mainly come from the outputs of LMO. Alignments with high similarity generated by GMO matcher are also reliable and these matched entity pairs should also be output by Falcon-AO. The average performance on these four tests is presented below:

	Precision	Recall	F-Measure	Time
Average	0.93	0.81	0.86	20s

2.2 Blind Tests

Blind tests consist of two groups: directory test [6] and anatomy test [7], and they are all real world cases.

Directory

We have got the alignment results on directory test by using the same set of parameters as the ones for benchmark test.

Anatomy

Falcon-AO detects that the FMA ontology and OpenGALEN ontology in anatomy test are so large that our GMO could not process them. Therefore, our alignment result of anatomy test comes only from a simple version of LMO.

[6]http://oaei.inrialpes.fr/2005/directory/

[7]http://oaei.inrialpes.fr/2005/anatomy/

3. GENERAL COMMENTS

In this section, we will summarize some features of Falcon-AO and the improvement in our future work, some comments about test cases will also be presented.

3.1 Comments on the Results

Our Falcon-AO performs well on benchmark tests 101-104, 201-210 and 221-247, and the results of test 301-304 are moderate, but on test 248-266, Falcon-AO doesn't perform so well. According to the results on these test cases, we can see the strengths and weaknesses of Falcon-AO:

Strengths

According to the experimental results, Falcon-AO performs well when the structures of the ontologies to be aligned are similar to each other or there is much lexical similarity between the two ontologies. Particularly, Falcon-AO performs well when the two ontologies have very little lexical similarity but high structural comparability.

Weaknesses

When there is little common vocabulary between the ontologies and in the meanwhile the structures of the ontologies are quite different, Falcon-AO can hardly find the exact mapping. Furthermore, GMO could not process very large ontologies, which means that while aligning very large ontologies, Falcon-AO cannot use their structural information.

3.2 Improvement of Falcon-AO

From the experiments we have learnt some lessons and plan to make improvements in the later versions. The following three improvements should be taken into account.

1. While expressing the same thing, people may use synonyms and even different languages. Therefore, it is necessary to use lexicons to match ontologies.
2. The current version of Falcon-AO did not support many-to-many mapping. The functionality of finding many-to-many mapping will be included in the later version of Falcon-AO.
3. Currently, the measure of linguistic comparability and structural comparability of ontologies are still simple and an improvement will be considered.

3.3 Comments on the Test Cases

The proposed test cases covered a large portion of discrepancies occurring of ontologies while aligning ontologies. Doing experiments on these test cases is helpful to improving the alignment algorithm and system. However, there are few tests on real world ontologies in benchmark tests.

4. CONCLUSION

While aligning real ontologies, linguistic matching plays an important role in matching process. Therefore, we integrate our GMO with LMO in Falcon-AO. From the experiments, we found that, Falcon-AO performed well on most of benchmark tests. It is also worthy of note that most of benchmark tests came from artificially altered ontologies, and more real world ontologies are expected to be included in benchmark tests.

Acknowledgments

This work is supported in part by National Key Basic Research and Development Program of China under Grant 2003CB317004, and in part by the NSF of Jiangsu Province, China, under Grant BK2003001. We are grateful to other members in our team for their discussion and comments that helped strengthen our work. We also would like to thank OAEI 2005 campaign for providing test cases.

5. REFERENCES

[1] D. Brickley and R. Guha (eds). RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation, 10 February 2004. Latest version available at http://www.w3.org/TR/rdf-schema/*

[2] P. Hayes (ed.). RDF Semantics. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/rdf-mt/*

[3] W. Hu, N. Jian, Y. Qu and Y. Wang. GMO: A Graph Matching for Ontologies. *Submitted to K-Cap workshop on Integrating Ontologies (2005). Now is available at http://xobjects.seu.edu.cn/project/falcon/GMO.pdf*

[4] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics - Doklady 10 (1966) 707-710*

[5] P. Patel-Schneider, P. Hayes, I. Horrocks (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. *W3C Recommendation 10 February 2004. Latest version is available at http://www.w3.org/TR/owl-semantics/*

[6] V. Raghavan and S. Wong. A Critical Analysis of Vector Space Model for Information Retrieval. *Journal of the American Society for Information Science 37(5) (1986) 279-287.*

[7] G. Salton. Automatic Text Processing. *Addison-Wesley Longman Publishing Co., Inc. (1989) 613-620.*

6. RAW RESULTS

Information about our project can be found at http://xobjects.seu.edu.cn/project/falcon/falcon.html, and our tool is available now.

6.1 Link to the System and Parameters File

Falcon-AO can be found at http://xobjects.seu.edu.cn/project/falcon/download.html.

6.2 Link to the Set of Provided Alignments

Results presented in this paper are available at http://xobjects.seu.edu.cn/project/falcon/results/falcon.zip.

6.3 Matrix of Results

Runtime Environment: Tests were run on a PC running Windows XP with an Intel Pentium 4 2.4 GHz processor and 512M memory.

No.	Precision	Recall	Time
101	1.0	1.0	4s
102	NaN	NaN	6s
103	1.0	1.0	4s
104	1.0	1.0	4s
201	0.98	0.98	105s
202	0.87	0.87	140s
203	1.0	1.0	4s
204	1.0	1.0	22s
205	0.88	0.87	55s
206	1.0	0.99	51s
207	1.0	0.99	51s
208	1.0	1.0	34s
209	0.86	0.86	102s
210	0.97	0.96	68s
221	1.0	1.0	4s
222	1.0	1.0	4s
223	1.0	1.0	4s
224	1.0	1.0	4s
225	1.0	1.0	4s
228	1.0	1.0	3s
230	0.94	1.0	4s
231	1.0	1.0	4s
232	1.0	1.0	4s
233	1.0	1.0	3s
236	1.0	1.0	3s
237	1.0	1.0	4s
238	0.99	0.99	4s
239	0.97	1.0	3s
240	0.97	1.0	4s
241	1.0	1.0	3s
246	0.97	1.0	3s
247	0.94	0.97	3s
248	0.84	0.82	100s
249	0.86	0.86	114s
250	0.77	0.70	7s
251	0.69	0.69	166s
252	0.67	0.67	119s
253	0.86	0.85	80s
254	1.0	0.27	4s
257	0.70	0.64	4s
258	0.70	0.70	162s
259	0.68	0.68	113s
260	0.52	0.48	7s
261	0.50	0.48	8s
262	0.89	0.24	4s
265	0.48	0.45	7s
266	0.50	0.48	8s
301	0.96	0.80	18s
302	0.97	0.67	3s
303	0.80	0.82	39s
304	0.97	0.96	18s

oMAP: Results of the Ontology Alignment Contest*

Umberto Straccia
ISTI-CNR, Via G. Moruzzi 1
56124 Pisa, Italy
straccia@isti.cnr.it

Raphaël Troncy
CWI Amsterdam, P.O. Box 94079
1090 GB Amsterdam, The Netherlands
raphael.troncy@cwi.nl

ABSTRACT
This paper summarizes the results of the *oMAP* system for the 2005 Campaign tests of the Ontology Alignment Evaluation Initiative. First, it describes the system and its main components. The results of the experiments for the three tasks follow. Then, we have a short discussion and interpretation of the results. Finally, we sketch some ideas to improve our system and provide the link to our current results.

1. PRESENTATION OF THE SYSTEM
oMAP [12] is a framework whose goal is to automatically align two OWL ontologies, finding the best mappings (together with their weights) between the entities defined in these ontologies. The final mappings are obtained by using the prediction of different classifiers. For this experiment, we have used terminological and machine learning-based classifiers, plus a new one, based on the structure and the semantics of the OWL axioms.

The *oMAP* implementation allows to align any OWL ontologies, represented in the RDF/XML syntax. Hence, it uses extensively the OWL API [1] and the Alignment API available in JAVA [3].

1.1 State, purpose, general statement
Our approach is inspired by the data exchange problem [4] and borrows from others, like GLUE [2], the idea of using several specialized components for finding the best set of mappings. The framework resumes partially the formalization proposed in [7] and extends the sPLMAP (*Schema Probabilistic Learning Mappings*) system to cope with the ontology alignment problem.

Theoretically, an ontology *mapping* is a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are respectively the source and target ontologies, and Σ is a finite set of *mapping constraints* of the form:

$$\alpha_{i,j}\ T_j \quad \leftarrow \quad S_i$$

where S_i and T_j are respectively the source and target entities. The intended meaning of this rule is that the entity S_i of the source ontology is mapped onto the entity T_j of the target ontology, and the confident measure associated with this mapping is $\alpha_{i,j}$. Note that a source entity may be mapped onto several target entities and conversely. But, we do not require that we have a mapping for every target entity.

Aligning two ontologies in *oMap* consists of three steps:

1. We form a possible Σ, and estimate its quality based on the quality measures for its mapping rules;
2. For each mapping rule $T_j \leftarrow S_i$, we estimate its quality $\alpha_{i,j}$, which also depends on the Σ it belongs to, i.e. $\alpha_{i,j} = w(S_i, T_j, \Sigma)$;
3. As we cannot compute all possible Σ (there are exponentially many) and then choose the best one, we rather build iteratively our final set of mappings Σ using heuristics.

Similar to GLUE [2], we estimate the weight $w(S_i, T_j, \Sigma)$ of a mapping $T_j \leftarrow S_i$ by using different classifiers $CL_1, \ldots, CL_n$. Each classifier CL_k computes a weight $w(S_i, T_j, CL_k)$, which is the classifier's approximation of the rule $T_j \leftarrow S_i$. For each target entity T_j, CL_k provides a rank of the plausible source entities S_{i_k}. Then we rely on a priority list on the classifiers, $CL_1 \prec CL_2 \prec \ldots \prec CL_n$ and proceed as follows: for a given target entity T_j, select the top-ranked mapping of CL_1 if the weight is non-zero. Otherwise, select the top-ranked mapping provided by CL_2 if non-zero, and so on.

In the next section, we briefly present the classifiers that are currently used in our framework. It is worth not-

*This work was carried out during the tenure of an ERCIM fellowship at CNR.

ing that some of them consider the terminological part of the ontologies only, while others are based on their instances (i.e. the values of the individuals). Finally, we end this section by introducing a new classifier that fully uses the structure and the semantics of ontology definitions and axioms.

1.2 Specific techniques used

The terminological classifiers work on the name of the entities (class or property) defined in the ontologies. In OWL, each resource is identified by a URI, and can have some annotation properties attached. Among others, the `rdfs:label` property may be used to provide a human-readable version of a resource's name. Furthermore, multilingual labels are supported using the language tagging facility of RDF literals. In the following, we consider that the name of an entity is given by the value of the `rdfs:label` property or by the URI fragment if this property is not specified. The typical terminological classifiers we used in *oMAP* compare the name of the entities, their stem (using the Porter stemming algorithm [9]), compute some similarity measures between the entity names (once downcased) such that the Levenshtein distance[6] (or edit distance), or compute similarity measure between the entity names using the WordNet®[1] relational dictionary.

Additionally, an ontology often contains some individuals. It is then possible to use machine learning-based classifiers to predict the weight of a mapping between two entities. The instances of an OWL ontology can be gathered using the following rules: we consider (i) the label for the named individuals, (ii) the data value for the datatype properties and (iii) the type for the anonymous individuals and the range of the object properties. For example, using the abstract syntax of [5], let us consider the following individuals :

```
Individual (x1 type (Conference)
            value (label "Int Conf on Knowledge Capture")
            value (location x2))
Individual (x2 type (Address)
            value (city "Banff") value (country "Canada"))
```

Then, the text gathered u_1 for the named individual x_1 will be (`"Int Conf on Knowledge Capture", "Address"`) and u_2 for the anonymous individual x_2 (`"Address", "Banff", "Canada"`). Typical and well-known classifiers used in machine learning such as Naive Bayes and kNN [11] have then been implemented in *oMAP* using these data.

Finally, a new classifier is able to use the semantics of the OWL definitions while being guided by their syntax. This *structural classifier* is fully described in [12]. It is used in the framework *a posteriori*. Indeed, we rely on the classifier preference relation $CL_{Name} \prec CL_{Stem} \prec CL_{EditDistance} \prec CL_{NaiveBayes}$. According to this preference relation, a set Σ' of mappings is determined. This set is given as input to the structural classifier. Then the structural classifier tries out all alternative ways to extend Σ' by adding some $T_j \leftarrow S_i$ if no mapping related to T_j is present in Σ'.

[1]WordNet: `http://wordnet.princeton.edu/`.

1.3 Adaptations made for the contest

All the classifiers detailed previously have been implemented to be compatible with the alignment API[2], thus easing their chaining. Therefore, our *oMAP* framework benefits from all the evaluation facilities for comparing our approach with other methods.

For the purpose of this contest, all our classifiers have been tested alone and then combined. We have then made no specific adaptations since we are still investigating how the classifiers should be combined to improve the overall quality of *oMAP*.

2. RESULTS

The tests proposed by the 2005 campaign of the Ontology Alignment Evaluation Initiative is composed of three tasks. Below, we describe the results of the *oMAP* system for these three tasks as well as the problems we have encountered.

2.1 Task1: benchmarks

The *benchmarks tests* are systematic benchmarks series produced for identifying the areas in which each alignment algorithm is strong and weak. Taking back the tests of the 2004 contest [13] and extending them, there are based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided. The full table results for this task is given in the section 6.3.

The overall score of *oMAP* for this task is quite good (see the table below).

Tests	Prec.	Rec.
1xx	0.96	1.00
2xx	0.80	0.63
3xx	0.93	0.64
H-Mean	0.83	0.66

However, *oMAP* has poor performance for the tests 25x and very bad performance for the tests 26x. Actually, the terminological and machine-learning based classifiers give wrong input to our structural classifier, since most of the data used in these classifiers have been changed in these tests. The structural classifier is then not able to counterbalance this effect and give also wrong alignments. It is the typical case where the

[2]`http://co4.inrialpes.fr/align/`.

other classifiers should be turn off and the structural classifier should work alone.

2.2 Task2: directory

The *directory real world case* consists of aligning web sites directory. It is more than two thousand elementary tests. These tests are blind in the sense that the expected alignments are not known in advance. *oMAP* success to compute the alignments for all of them in a total time of about 11 minutes on a normal PC laptop.

2.3 Task3: anatomy

The *anatomy real world case* covers the domain of body anatomy and consists of two big ontologies with an approximate size of several 10k classes and several dozen of relations. This test was clearly the hardest one and our *oMAP* system has not been able to begin the computation of the alignment.

The main problem is the size of the FMA ontology since the XML parser cannot load the full ontology and crash for *out of memory* problem. However, we notice also that this ontology contains some small mistakes:

- the entities, such that `&xsd; &rdfs;` ... , are not defined;
- a datatype property contains an error since the value of its `rdf:ID` attribute contains a space which is forbidden. The correct definition of this property should be:

```
<owl:DatatypeProperty
    rdf:ID="has_inherent_3-D_shape"
    rdfs:label="has inherent 3-D shape">
        <rdfs:domain rdf:resource="&rdfs;Resource"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
```

Once these mistakes have been corrected, the FMA ontology can be validated with an RDF parser, but the parser included in the alignment API is not able to deal with, thus preventing the beginning of the alignment computation by *oMAP*.

3. GENERAL COMMENTS

3.1 Comments on the results

As we have seen in the previous section, the *oMAP* framework is based on numerous classifiers. Each of them try to predict some mappings between the ontology entities and these predictions are then combined. Some classifiers are strongly based on the labels attached to the entities (terminological). Therefore, they performed especially well when labels were preserved. The machine learning-based classifiers can use the individuals of the ontologies if they contain. The main improvement of our approach is then the structural classifier which is able to align two ontologies solely on their semantics, and without the presence of individuals or even labels.

Finally, the combination of all these classifiers rely on many different features and thus balance the influence of individual features. This mixed approach tend to success on every case (either if the labels are preserved or not) even if we dispose yet of a large progression margin.

The main weakness of *oMAP* is clearly its computation time. Like we have seen previously, our approach begins to form some possible Σ sets, for evaluating the weight of each mapping rules they contain. The generation of *all* possible Σ sets becomes quickly a critical issue since this number can be huge (exponentially many) [12]. We have addressed this problem by implementing some approximation. The most efficient for reducing the space search is a local maximum heuristic. When forming a Σ set, we consider firstly a class from the first ontology, and gather all the entities (classes and properties) involved in its closure definition. We do the same for each classes of the second ontology and we evaluate all these small Σ sets for retaining the best one. We iterate this process over all the classes. Additional criteria allow us to guarantee the convergence of our approach (i.e. the order of the classes considered has no significance).

3.2 Discussions on the way to improve the proposed system

As future work, we see some appealing points. Additional classifiers using more terminological resources can be included in the framework, and are currently under implementation, while the effectiveness of the machine learning part could be improved using other measures like the KL-distance. While to fit new classifiers into our model is straightforward theoretically, practically finding out the most appropriate one or a combination of them is quite more difficult. In the future, more variants should then be developed and evaluated to improve the overall quality of *oMAP*. Furthermore, the appropriateness of each classifier could be learned via regression.

3.3 Comments on the test cases

It is always difficult to create good test cases. The benchmarks tests should cover the widest range of discrepancies occurring when having two ontologies. For the cases where a lot of features were explicitly changed at the same time, *oMAP* is clearly less good, but such a mess is unlikely to occur in real cases. On the contrary, in the real world scenario, as presented by the last four test cases, *oMAP* performs quite good.

4. CONCLUSION

As the number of Semantic Web applications is growing rapidly, many individual ontologies are created. The development of automated tools for ontology alignment will be of crucial importance. In this paper, we have presented the results for the 2005 campaign of the Ontology Alignment Evaluation Initiative of our formal framework for ontology Matching, which for ease we call *oMap*. oMap uses different classifiers to estimate the quality of a mapping. Novel is the classifier which uses the structure of the OWL constructs and thus the semantics of the entities defined in the ontologies. We have implemented the whole framework and we continue to evaluate it on independent benchmark tests such that the ones provided by this contest.

5. REFERENCES

[1] Bechhofer, S., Volz, R., Lord, P.W.: Cooking the Semantic Web with the OWL API. In *2nd International Semantic Web Conference (ISWC'03)*, Sanibel Island, Florida, USA, 2003, 659–675.

[2] Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to Match Ontologies on the Semantic Web. The VLDB Journal, **12**(4), 2003, 303–319.

[3] Euzenat, J.: An API for ontology alignment. In *3rd International Semantic Web Conference (ISWC'04)*, Hiroshima, Japan, 2004, 698–712.

[4] Fagin, R., Kolaitis, P.G., Miler, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. In *9th International Conference on Database Theory (ICDT'03)*, Sienna, Italy, 2003, 207–224.

[5] Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics, **1**(1), 2003, 7–26.

[6] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii Nauk SSSR, **163**(4):845–848, 1965 (Russian). English translation in Soviet Physics Doklady, **10**(8):707–710, 1966.

[7] Nottelmann, H., Straccia, U.: sPLMap: A probabilistic approach to schema matching. In *27th European Conference on Information Retrieval (ECIR'05)*, Santiago de Compostela, Spain, 2005, 81–95.

[8] OWL, Web Ontology Language Reference Version 1.0. W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/owl-ref/`

[9] Porter, M.F.: An algorithm for suffix stripping. Program, **14**(3), 1980, 130–137.

[10] RDF, Ressource Description Framework Primer W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/rdf-primer/`

[11] Sebastiani, F.: Machine learning in automated text categorization. ACM Comuting Surveys, **34**(1), 2002, 1–47.

[12] Straccia, U., Troncy, R.: oMAP: Combining Classifiers for Aligning Automatically OWL Ontologies. In *6th International Conference on Web Information Systems Engineering (WISE'05)*, New York City, New York, USA, 2005.

[13] Sure, Y., Corcho, O., Euzenat, J., Hughes, T. editors. *3rd International Workshop on Evaluation of Ontology-based Tools (EON'04)*, Hiroshima, Japan, 2004.

6. RAW RESULTS

6.1 Link to the system and parameters file

The *oMAP* system is available at: `http://homepages.cwi.nl/~troncy/`.

It can be run with the command:

```
java -jar omap.jar -i %method% -r %renderer%
     -o %resultFile% %sourceOnto% %targetOnto%
```

where:

- *method* is: `it.cnr.isti.OMapAlignment`;
- *renderer* is: `fr.inrialpes.exmo.align.impl.renderer.RDFRendererVisitor2`;
- *resultFile* is the name of the result file;
- *sourceOnto* and *targetOnto* are the absolute URIs of the source and the target ontologies to align.

6.2 Link to the set of provided alignments

All the alignments computed for this campaign are available in the alignment format [3] at: `http://homepages.cwi.nl/~troncy/OAEI/`.

6.3 Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	0.96	1.00	20ms
102	Irrelevant ontology	0.00	NaN	
103	Language generalization	0.96	1.00	20ms
104	Language restriction	0.96	1.00	20ms
201	No names	0.88	0.38	20ms
202	No names, no comments	0.85	0.24	20ms
203	No comments	0.96	1.00	
204	Naming conventions	0.95	0.89	40ms
205	Synonyms	0.81	0.63	40ms
206	Translation	0.89	0.49	40ms
207		0.89	0.49	40ms
208		0.96	0.90	30ms
209		0.73	0.54	40ms
210		0.90	0.39	40ms
221	No specialisation	0.96	1.00	20ms
222	Flatenned hierarchy	0.96	1.00	20ms
223	Expanded hierarchy	0.96	1.00	20ms
224	No instance	0.96	1.00	20ms
225	No restrictions	0.96	1.00	20ms
228	No properties	0.92	1.00	20ms
230	Flattened classes	0.91	1.00	20ms
231	Expanded classes	0.96	1.00	20ms
232		0.96	1.00	20ms
233		0.92	1.00	20ms
236		0.92	1.00	20ms
237		0.95	1.00	20ms
238		0.96	1.00	20ms
239		0.85	1.00	20ms
240		0.87	1.00	20ms
241		0.92	1.00	20ms
246		0.85	1.00	20ms
247		0.87	1.00	20ms
248		0.85	0.24	50ms
249		0.85	0.23	50ms
250		0.05	0.06	50ms
251		0.85	0.25	50ms
252		0.85	0.24	50ms
253		0.85	0.23	50ms
254		0.06	0.06	50ms
257		0.00	0.00	50ms
258		0.85	0.25	50ms
259		0.85	0.24	50ms
261		0.03	0.03	50ms
262		0.00	0.00	50ms
265		0.00	0.00	50ms
266		0.00	0.00	50ms
301	Real: BibTeX/MIT	0.94	0.25	40ms
302	Real: BibTeX/UMBC	1.00	0.58	40ms
303	Real: Karlsruhe	0.90	0.79	40ms
304	Real: INRIA	0.91	0.91	40ms

OLA in the OAEI 2005 alignment contest

Jérôme Euzenat
INRIA Rhône-Alpes
Montbonnot, France
euzenat-at-inrialpes.fr

Philippe Guégan
DIRO, Université de Montréal
Montr eal (Qc), Canada
Guegan-at-iro.umontreal.ca

Petko Valtchev
DIRO, Université de Montréal
Montr eal (Qc), Canada
Petko.Valtchev-at-umontreal.ca

ABSTRACT

Among the variety of alignment approaches (e.g., using machine learning, subsumption computation, formal concept analysis, etc.) similarity-based ones rely on a quantitative assessment of pair-wise likeness between entities. Our own alignment tool, OLA, features a similarity model rooted in principles such as: completeness on the ontology language features, weighting of different feature contributions and mutual influence between related ontology entities. The resulting similarities are recursively defined hence their values are calculated by a step-wise, fixed-point-bound approximation process. For the OAEI 2005 contest, OLA was provided with an additional mechanism for weight determination that increases the autonomy of the system.

1. PRESENTATION OF THE SYSTEM

OLA (for *O*WL-*L*ite *A*lignment) is an open-source tool jointly developed by teams at University of Montréal and INRIA Rhône Alpes. It features similarity-based alignment and a set of auxiliary services supporting the manipulation of alignment results [5, 6].

1.1 General purpose statement

The primary goal behind the OLA tool design is to perform alignment of ontologies expressed in OWL, with a short-term emphasis on OWL-Lite and long-term one on OWL-DL. However, its GUI component, VisOn[1] allows for many other services involving alignments (in the sense of [4]) to be accessed.

1.1.1 Functional specifications

From a mere algorithm for automated alignment construction, OLA has grown for the last year to an environment for alignment manipulation. Indeed, in its current version, the system offers, via its GUI component VisOn, the following services:

- parsing and visualization of OWL-Lite and OWL-DL ontologies,
- computation of similarities between entities from two ontologies,
- extraction of alignments from a pair of ontologies, provided with a set of similarity matrices, one per category of ontology entities (see below),
- manual construction of alignments by composing entity pairs from two ontologies,
- use of an existing (partial) alignment as a seed for automated alignment construction (alignment completion),
- alignment visualization,
- comparison of two alignments.

In the remainder, the focus will be limited to the automated alignment construction with OLA.

1.1.2 Principles of the alignment in OLA

The following fundamental principles underly the design of the three key mechanisms in OLA, internal representation of the ontology, similarity computation and alignment extraction, that are involved in the global ontology alignment process:

All-encompassing comparison : We tend to believe that all the available knowledge about a pair of ontology entities should be taken into account when aligning. This does not exclude the possibility of ignoring particular aspects, i.e., OWL instances in case of OWL class comparison. However such a choice should be deliberately made by the tool user, here through appropriate weight assignment, or, if performed by an automated mechanisms, should reflect some particularity, either of the entire ontology (e.g., global absence of instances in both ontologies) or of the pair of entities at hand (e.g., local absence of instances in the pair of classes to be compared).

[1] see `http://www.iro.umontreal.ca/∼owlola/`

Highest automation level : Although we recognize that the entire alignment process often needs to be set on a semi-automated basis, we nevertheless argue in favor of a completely automated process for "draft" alignment generation. Thus, we see the OLA user providing a minimal set of parameters at the initial steps of the process whereas the tool will suggest one or more candidate alignments at the end, without any other human intervention.

Category-dependent comparison : Following the syntactic structure of the OWL language, entities are divided into categories, e.g., *classes*, *objects*, *properties*, *relations*, and only entities of the same category are compared. Moreover, the entities of a category are compared using similarity functions of the same basic shape. The respective category functions comprise the same factors and the same weights. They are further customized for each pair of category entities by projecting them over the actual feature space of the entities (which may be far smaller than the complete space of the category).

Comparability of similarity results : To enable comparison of similarity scores between different alignment tasks but also for some computational reasons, a set of useful properties is insured for the similarity functions: *normalization*, *positiveness*, *maximalness*[2], and *symmetry*[3].

1.1.3 Current limitations

- Although it would be of certain value for alignment, OLA currently offers no inference mechanisms that could help complete the entity descriptions. In particular, inheritance is not used to expand entities, mostly out of efficiency considerations.

- Although neighborhoods play crucial role in the similarity definition, two neighbor entities are not necessarily affecting each other's respective similarities to a pair of other entities. As only descriptive knowledge is taken into account, given two such entities, say e_1 and e_2, for e_2 to appear in a similarity expression for e_1, it should be considered as part of the description of the latter. For instance, a data type is not seen as being described by a property whose range the datatype represents. Consequently, datatypes are compared in an ontology-independent manner.

- Category borders are not similarity-permeable: Only entities from the same category are compared for similarity and hence for alignment.

1.2 Specific techniques used

OLA features an alignment process that splits into three basic steps: constructing the intermediate representation of the compared ontologies as labeled graphs, computing the similarity of each pair of same-category entities from the respective ontology graphs, extracting an alignment from the similarity matrices for each category.

1.2.1 OL-Graph construction

OL-Graphs are graph structures that provide an easy-to-process inner representation of OWL ontologies. An OL-Graph is a labeled graph where vertices correspond to OWL entities and edges to inter-entity relationships. As described in [6], the set of different vertex categories is: class (C), object (O), relation (R), property (P), property instance (A), datatype (D), datavalue (V), property restriction labels (L). Furthermore, we distinguish between datatype relations (R_{dt}) and object relations (R_o), and between datatype properties (P_{dt}) and object ones (P_o).

The OL-Graph model allows the following relationships among entities to be expressed:

- *specialization* between classes or relations (denoted $\mathcal{S}$),
- *instanciation* (denoted $\mathcal{I}$) between objects and classes, property instances and properties, values and datatypes,
- *attribution* (denoted $\mathcal{A}$) between classes and properties, objects and property instances;
- *restriction* (denoted $\mathcal{R}$) expressing the restriction on a property in a class,
- *valuation* (denoted $\mathcal{U}$) of a property in an object.

The OL-Graph of an ontology is built after the ontology is parsed[4]. The process of OL-Graph construction is described in [8].

1.2.2 Similarity model

The similarity functions used in OLA are designed in a category-specific manner and cover all the available descriptive knowledge about an entity pair. Thus, given a category X of OL-Graph nodes, the similarity of two nodes from X depends on:

- the similarities of the terms used to designate them, i.e., URIs, labels, names, etc.,
- the similarity of the pairs of neighbor nodes in the respective OL-Graphs that are linked by edges expressing the same relationships (e.g., class node similarity depends on similarity of superclasses, of property restrictions and of member objects),
- the similarity of other local descriptive features depending on the specific category (e.g., cardinality intervals, property types)

[2]With normalization, this amounts to forcing scores of 1 for identical entities within identical ontologies

[3]The price to pay for symmetry is the impossibility of detecting subsumption by this purely numerical procedure.

[4]So far, we use the OWL API [1].

Funct.	Node	Factor	Measure
Sim_O	$o \in O$	$\lambda(o)$	sim_L
		$a \in A,\ (o,a) \in \mathcal{A}$	$MSim_A$
Sim_A	$a \in A$	$r \in R,\ (a,r) \in \mathcal{R}$	Sim_R
		$o \in O,\ (a,o) \in \mathcal{U}$	$MSim_O$
		$v \in V,\ (a,v) \in \mathcal{U}$	$MSim_V$
Sim_V	$v \in V$	value literal	type dependent
Sim_C	$c \in C$	$\lambda(c)$	sim_L
		$p \in P,\ (c,p) \in \mathcal{A}$	$MSim_P$
		$c' \in C,\ (c,c') \in \mathcal{S}$	$MSim_C$
sim_D	$d \in D$	$\lambda(r)$	XML-Schema
Sim_R	$r \in R$	$\lambda(r)$	sim_L
		$c \in C,\ (r, \mathtt{domain}, c) \in \mathcal{R}$	$MSim_C$
		$c \in C,\ (r, \mathtt{range}, c) \in \mathcal{R}$	$MSim_C$
		$d \in D,\ (r, \mathtt{range}, d) \in \mathcal{R}$	Sim_D
		$r' \in R,\ (r,r') \in \mathcal{S}$	$MSim_R$
Sim_P	$p \in P$	$r \in R,\ (p,r') \in \mathcal{S}$	Sim_R
		$c \in C,\ (p, \mathtt{all}, c) \in \mathcal{R}$	$MSim_C$
		$n \in \{0,1,\infty\},\ (p, \mathtt{card}, n) \in \mathcal{R}$	equality

Table 1: Similarity function decomposition (`card` = `cardinality` and `all` = `allValuesFrom`).

Datatype and datavalue similarities are external to our model and therefore they are either user-provided or measured by a standard function (e.g., string identity of values and datatype names/URIs).

Formally, given a category X together with the set of relationships it is involved in, $\mathcal{N}(X)$, the similarity measure $Sim_X : X^2 \to [0,1]$ is defined as follows:

$$Sim_X(x,x') = \sum_{\mathcal{F} \in \mathcal{N}(X)} \pi_{\mathcal{F}}^X MSim_Y(\mathcal{F}(x), \mathcal{F}(x')).$$

The function is normalized, i.e., the weights $\pi_{\mathcal{F}}^X$ sum to a unit, $\sum_{\mathcal{F} \in \mathcal{N}(X)} \pi_{\mathcal{F}}^X = 1$. for the computability The set functions $MSim_Y$ compare two sets of nodes of the same category (see [6] for details). Table 1 illustrates the set of similarities in our model.

Following the lessons learned with our participation in the EON 2004 alignment contest [?], we have adapted the above measure to fit cases where particular pair of entities is described only by a small subset of the entire set of category descriptors. Thus, a descriptive factor is ignored for similarity computation whenever neither of the compared entities possesses a neighbor with the underlying link label (e.g., no instances for a pair of compared classes). In this case, not only its weight is set to 0, but also the weights of the remaining "active" factors are increased correspondingly. To scale that principle up to the entire set of descriptive factors, the following simple mechanism has been realized in OLA: In order to keep both normalization and equity in similarity values, the weights of all non-null factors for a given entity pair are divided through their sum. Thus, for a category X, the similarity measure $Sim_X^+ : X^2 \to [0,1]$ becomes:

$$Sim_X^+(x,x') = \frac{Sim_X(x,x')}{\sum_{\mathcal{F} \in \mathcal{N}^+(x,x')} \pi_{\mathcal{F}}}$$

where $\mathcal{N}^+(x,x')$ is the set of all relationships $\mathcal{F}$ for which $\mathcal{F}(x) \cup \mathcal{F}(x') \neq \emptyset$ [5].

OLA relies on various functions for identifiers comparison. Both string distances and lexical distances are used. Lexical distances rely on an exploration of WordNet 2.0 [7] with a quantitative assessment of the "relatedness" between two, possibly multi-word, terms. More specifically, the degree of relatedness between two WordNet entries is computed as the ratio between the depth, in graph-theoretic sense, of the most specific common hypernym and the average of both term depths. The computation of multi-word term similarity consists in first splitting the terms into a set of tokens each and then comparing all possible pairs of tokens from opposite sets using the above depth-based principle. The global term similarity is then computed as a similarity-based matching between both sets (see above).

As circular dependencies are impossible to avoid with the above definitions, the computing of the similarity values requires non-standard mechanisms. Following [2, 9], an equation system is composed out of the similarity definitions where variables correspond to similarities of node pairs while coefficients come from weights. The process of iterative, fixed-point-bound resolution of that system, as well as the related convergence and determinism issues are described in [6].

1.3 Implementation

OLA is implemented in JAVA. Its architecture follows the one of the Alignment API and the recent implementation that was described in [4]. OLA relies on the OWL API [1] for parsing OWL files. An entire subsystem is dedicated to the onstruction of OL-Graphs on top of the parsed ontologies. A set of further components that offer similarity computation services: substring distances, edit distances, Hamming distance, WordNet interface (via the JWNL library [3]), etc., that were originally designed for OLA are now part of the Alignment API. The VISON GUI component offers a uniform interface to all services provided by Alignment API and OLA. In particular, it visualizes both the input data, i.e., the OL-Graphs, and the final result, i.e., the alignment file, of the global process.

1.4 Adaptations made for the contest

Along the preparation of the AOEI 2005 contest, a row of changes have been made to the system in order to make it fit the complexity of the alignment discovery task. The most striking one is the introduction of a weight-computing mechanism that eliminates the necessity for the tool user to provide initial weights and hence makes a significant step towards full automation of the alignment process.

[5] That is, there exists at least one y such that $(x,y) \in \mathcal{F}$ or at least one y' such that $(x',y') \in \mathcal{F}$.

1.4.1 Weight computing mechanism

As it is far from obvious for novice users how to weigh the different similarity factors, we initiated work on incorporating a weight computing mechanism within the system. The intended mechanism is both intuitive and effective so that alignment practitioners with various skill levels could find a match for their knowledge and experience. So far, we used a simple heuristic method that, according to the obtained results, performs reasonably well. The basic idea of the method consists in distributing the weights among similarity factors in the generic similarity function of a node category according to the relative importance of the corresponding category in the entire ontology. That is to say we use the average number of links of the corresponding type per entity of the category at hand. For instance, the greater the number of super-class links in the ontology, the higher the weight of the super-class factor in the class similarity formula.

1.4.2 Similarity measure for entity names

OLA uses two alternative modes of comparison for entity names (URIs, labels, etc.): a string measure[6] (a default) and a lexical similarity measure that relies on WordNet 2.0 (see above).

The highly sophisticated lexical similarity measure that was used in OLA for the EON competition has been replaced by a simpler but more purposeful one. Indeed, the initial function compared multi-word terms on three separate axes: nouns, verbs and adjectives, as provided by WordNet 2,0. Such comparison seemed appropriate for cases where the meanings of a word fall in more than one part-of-speech category. The inter-word similarities on each axis were aggregated by an independent best-match computations while the three resulting values were further combined to a single one via a weighted sum.

The new measure trades separate matchings on speech-part-wise basis to a single global matching along entry similarities that aggregate all three possible aspects of a word. Thus, the words are compared to each other with all possible meanings and the highest similarity over a single pair of meanings is taken for the words.

For the OAEI competition, as we had to rely on a fixed parameter set for the entire collection of tests, we have chosen to force the use of the string distance. Indeed, it showed better performances while being much more efficient than the WordNet-based computation.

Nevertheless, the improved lexical similarity was not completely discarded: it is currently used as a pre-processing tool that helps decide automatically the distribution of weights among similarity factors.

[6]subString distance provided by the Alignment API

1.4.3 Minor adaptations

Following experiences from EON 2004, a set of simple but decisive modifications have been applied in order to prevent the precision leak in the tests. First, the instances have been excluded from the alignments by default, although the possibility is given to the user to reverse this choice. Then, entities external to the ontologies at hand have also been excluded from the alignment (but not from the similarity computation). Finally, one-to-one alignment production has been enforced in OLA to increase the potential recall of the resulting alignment.

2. RESULTS

The comments are grouped by test categories.

2.101 Tests 10X

OLA performed very well on the tests of this group. This seems to be due to the fact that while the language varies along the individual tests of the group, the basic ontology entities involved in the similarity computation remain unchanged with respect to the reference ontology.

2.102 Tests 2XX

The performances of the algorithm seem to suggest that three sub-groups of tests can be distinguished. The first one comprises the tests 21X, 22X, 23X and 24X, with a small number of exceptions where the performance have been:

- Quite good: This is the case of tests 201, 202, with random class names. The random names were putting a strain on the ability of the algorithm to propagate similarity along the network of node pairs. Obviously, our technique needs some improvements on that point.
- Satisfactory: In the case of tests 248, 249, there is a combination of missing (or random) names with one other missing factor. For tests 248, 249, the missing factors are hierarchy (sub-class links) and instances, respectively. Both play important role in similarity computation of classes, whenever these are stripped of their names as is the case with these two ontologies. Hence the sharp drop in precision and recall with respect to the preceding tests.
- Weak: The notorious failure here have been the tests 205, 209, which are the only ones to use of synonymous names in the ontology entities (with respect to the intial ontology). As WordNet has been plugged-out of the similarity computation, these results are not surprising.

The second groups is made of the tests 25X. Here OLA performances varied substantially: from extremely poor

(254) to satisfactory (252, 259).

The last five ontologies of the group, the 26X ones, have proven to represent a serious obstacle for OLA. The performances of the system here were poor to very poor.

2.103 Tests 30X

The real-world ontologies of the group 30X made OLA perform in an unimpressive way. We believe that this is due to the fact that string similarity was systematically used as identifier comparison means. Indeed, tentative runs with WordNet as basis for name similarity yielded way more precise alignments on that group. Unfortunately, they also brought down the overall statistics from the entire test set such as mean precision and mean recall. Hence the choice of the WordNet-based lexical similarity for a default name comparison means has been dropped.

3. GENERAL COMMENTS

3.1 Comments on the results

The results show a substantial progress has been made since the EON 2004 alignment contest. With respect to the performances of OLA at that forum, we made a big leap amounting to about 25% in both mean precision and mean recall.

Nevertheless, we see that a vast space for improvement lays ahead of our project. The weaknesses of the current similarity mechanisms can be summarized as follows. First, the tuning of the algorithm is still a rigid process. Indeed, while the weights can now be computed following a specific footprint of the ontology, a mechanism for the choice of a particular name similarity on the same basis has yet to be defined.

Second, although we take into account the biggest possible amount of knowledge about entities, there are sources of similarity that have been ignored so far, in particular entity comments.

3.2 Discussions on the way to improve the proposed system

Besides expanding the lexical processing to comments in entities and providing a flexible decision mechanism for the choice of the default name similarity, a possible improvement of the system will be the integration of a learning module for weight estimation. As for similarity, the biggest challenge here is to define the representation of the input data, i.e., the descriptors of the entries for the learning algorithm.

Another research track would be the definition of an optimal matching algorithm. In fact, the current procedures are sub-optimal in the sense that they only chose local optima for each aligned entity. Consequently, as strict 1:1 matchings are to be produced, a single bad choice could easily generate a chain of wrong alignment decisions and thus negatively impact the performances of the tool.

3.3 Comments on the experiment

Two months during summer period is definitely too short to run shuch an experiment.

4. CONCLUSION

In its latest version, OLA has proven a more robust tool for alignment than it was a year before. While the difficulties with real-world ontologies persist, the progress on noisy ones has been substantial.

The next key topic of the research around OLA will be the automation of the weight computation for a specific pair of ontologies.

5. RAW RESULTS

5.1 Link to the set of provided alignments

A .zip archive of all the contest results is available at the following URL:

`http://www.iro.umontreal.ca/~owlola/OAEI.html`

5.2 Link to the system and parameters file

A similar archive with the parameters and the .jar files used in the contest-related experiments is available at the following URL:

`http://www.iro.umontreal.ca/~owlola/OAEI.html`

5.3 Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	1	1	
103	Language generalization	1	1	
104	Language restriction	1	1	
201	No names	0.71	0.62	
202	No names & no comments	0.66	0.56	
203	No comments	1	1	
204	Naming conventions	0.94	0.94	
205	Synonyms	0.43	0.42	
206	Translation	0.94	0.93	
207		0.95	0.94	
208		0.94	0.94	
209		0.43	0.42	
210		0.95	0.94	
221	No specialisation	1	1	
222	Flatenned hierarchy	1	1	
223	Expanded hierarchy	1	1	
224	No instance	1	1	
225	No restrictions	1	1	
228	No properties	1	1	
230	Flattened classes	0.95	0.97	
231		1	1	
232		1	1	
233		1	1	
236		1	1	
237		0.97	0.98	
238		0.99	0.99	
239		0.97	1	
240		0.97	1	
241		1	1	
246		0.97	1	
247		0.97	1	
248		0.59	0.46	
249		0.59	0.46	
250		0.3	0.24	
251		0.42	0.3	
252		0.59	0.52	
253		0.56	0.41	
254		0.04	0.03	
257		0.25	0.21	
258		0.49	0.35	
259		0.58	0.47	
260		0.26	0.17	
261		0.14	0.09	
262		0.2	0.06	
265		0.22	0.14	
266		0.14	0.09	
301	Real: BibTeX/MIT	0.42	0.38	
302	Real: BibTeX/UMBC	0.37	0.33	
303	Real: Karlsruhe	0.41	0.49	
304	Real: INRIA	0.74	0.66	

6. REFERENCES

[1] Sean Bechhofer, Raphael Voltz, and Phillip Lord. Cooking the semantic web with the OWL API. In *Proc. 2nd International Semantic Web Conference (ISWC), Sanibel Island (FL US)*, 2003.

[2] Gilles Bisson. Learning in FOL with similarity measure. In *Proc. 10th AAAI, San-Jose (CA US)*, pages 82–87, 1992.

[3] John Didion. The Java WordNet Library, http://jwordnet.sourceforge.net/, 2004.

[4] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd ISWC*, pages 698–712, Hiroshima (JP), 2004.

[5] Jérôme Euzenat and Petko Valtchev. An integrative proximity measure for ontology alignment. In *Proc. ISWC-2003 workshop on semantic information integration, Sanibel Island (FL US)*, pages 33–38, 2003.

[6] Jérôme Euzenat and Petko Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proc. 15th ECAI*, pages 333–337, Valencia (ES), 2004.

[7] A.G. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[8] Mohamed Touzani. Alignement d'ontologies dans OWL. Master's thesis, University of Montréal, in preparation.

[9] Petko Valtchev. *Construction automatique de taxonomies pour l'aide à la représentation de connaissances par objets.* Thèse d'informatique, Université Grenoble 1, 1999.

Organizing Committee

Benjamin Ashpole, Lockheed Martin Advanced Technology Lab
Marc Ehrig, University of Karlsruhe
Jérôme Euzenat, INRIA Rhône-Alpes
Heiner Stuckenschmidt, Vrije Universiteit Amsterdam

special thanks to Mikalai Yatskevich (University of Trento), who supported us for the evaluation

Program Committee

Dean Allemang (TopQuadrant Inc.)
Jos de Bruijn (DERI Innsbruck)
Oscar Corcho (University of Manchester)
Christine Golbreich (University Rennes 1)
Lewis Hart (Applied Minds)
Todd Hughes (Lockheed Martin Advanced Technology Labs)
Ryutaro Ichise (NII, Tokyo)
Yannis Kalfoglou (University of Southampton)
Deborah McGuinness (Stanford University)
Natasha Noy (Stanford University)
Amit Sheth (University of Georgia and Semagix)
Pavel Shvaiko (University of Trento)
Michael Sintek (DFKI, Kaiserslautern)
Umberto Straccia (CNR)
York Sure (University of Karlsruhe)
Mike Uschold (Boeing Corp.)
Petko Valtchev (DIRO, Université de Montréal)

Acknowledgement

This workshop along with its organizers is partially supported by the Knowledge Web European network of excellence (IST-2004-507482).