Comparing Concepts in Differentiated Ontologies

Peter C. Weinstein[1]
Appnet, Inc.
650 Avis Dr., Suite 100
Ann Arbor, MI 48108
peter.weinstein@appnet.com

William P. Birmingham
Artificial Intelligence Laboratory
University of Michigan
1101 Beal Ave., Ann Arbor, MI 48109
wpb@eecs.umich.edu

Abstract

Concepts in *differentiated ontologies* inherit definitional structure from concepts in shared ontologies. Shared, inherited structure provides a common ground that supports measures of “description compatibility.” These algorithms are the primary contribution of this paper. The description-compatibility measures compare concepts to predict semantic compatibility, the probability that an instance of a recommendation will satisfy a request. The description-compatibility measures cross a spectrum regarding their knowledge of the semantics of roles in concept definitions. Some of the measures identify and analyze correspondences among elements of the definitions, and are thus a form of analogical reasoning. We use simulations to evaluate the description-compatibility measures in detail. Description compatibility can be used to rank alternative query translations, and to guide search for capabilities across communities that subscribe to differentiated ontologies.

1. INTRODUCTION

The same fundamental technical capability is critical to the success of research in several important areas, including queries across multiple databases (Mena, Illarramendi et al. 1999), reuse of knowledge components (Benjamins, Plaza et al. 1998), and dynamic composition of agent teams to satisfy user requests (Weinstein, Birmingham et al. 1999). Each of these problems requires matching descriptions of needs to descriptions of available resources, where the descriptions use terminology subject to semantic heterogeneity. We define terminology as *semantically heterogeneous* when it includes symbols with inconsistent denotations (the same symbol references different sets of objects), redundant denotations (different symbols reference identical sets of objects), or both. Typically, systems must deal with terms whose denotations overlap.

In this paper, we describe algorithms that compare concept definitions to predict their semantic compatibility, which we define as the probability that an instance of a recommendation will satisfy the request. These “description compatibility” algorithms are made possible by our assumption of *differentiated ontologies.* In these structures:

- concepts are defined by their relations to other concepts using description logic or another formalism that supports inference of subsumption, and
- local concepts inherit definitional structure from concepts in shared ontologies.

[1] This work was conducted by the author as part of his dissertation research at the University of Michigan.

These restrictions reflect our expectations that diverse system participants can adhere to standard terminology on a general level, if given the freedom to develop specialized meanings in local communities.

The measures of description compatibility vary in the complexity of their computation, and the knowledge that they require about the domain. Several measures utilize *matching:* the identification of maximal one-to-one correspondences between elements of the compared definitions. Matchings enable analysis of similarities and differences between the concepts to predict their semantic compatibility. This reasoning is not logically sound, because in any particular situation differences between the concepts may or may not matter.

We study the accuracy of our compatibility measures with simulations. These include ontologies generated to describe finite universes of artificial objects. In real-world applications, it is difficult to quantify the meaning of a concept. In our simulations, however, a concept's meaning is its denotation (a set of objects), and the semantic compatibility of two concepts is a function of the intersection of their denotations. The simulations thus provide an authoritative standard for evaluating the accuracy of the description compatibility measures. Our results are statistical analyses of empirical data generated by the simulations.

1.1. An Example

The following hypothetical example is from the domain of digital libraries (Weinstein, Birmingham et al. 1999). Consider a request where the user asks her interface agent for a compact disc containing Beethoven's opera *Fidelio*. Her agent formulates a query using symbols from its local terminology, which is oriented towards the needs of libraries and includes the terms Uniform-Title, Publishing-Format, and Audio-Recording (see Figure 1). The user agent then submits the query to a remote agent that manages a collection oriented towards music consumers. This collection agent uses different terminology, including CompactDisc, to express a meaning that overlaps the intended meaning of the request. The question is whether the collection agent can recognize and evaluate the compatibility of the request and the concept of CompactDisc, perhaps leading to a recommendation of the particular instance of CompactDisc that satisfies the query.

Request concept:	Uniform-Title(?x, Fidelio) and Publishing-Format(?x, Audio-Recording)
Recommendation Concept:	CompactDisc(?x)
Recommendation Instance:	cd000123 where CompactDisc(cd000123) and Has-Title(cd000123, Fidelio)

Figure 1: Request and recommendation subject to semantic heterogeneity

Figure 2 shows a graph of the request concept. Most of the structure in the figure comes from the ontological definitions of concepts, such as Audio-Recording rather than the query itself.

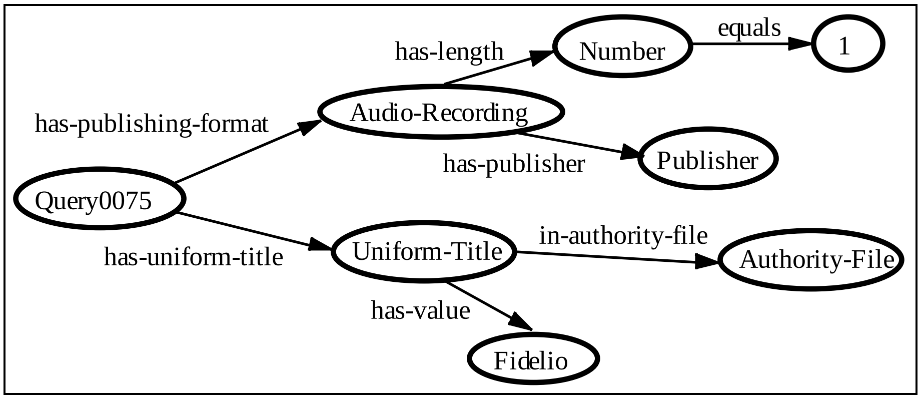


Figure 2: A request concept as a graph

Assuming differentiated ontologies mitigates syntactic and semantic heterogeneity, to a degree that depends on the degree of differentiation. Figure 3 shows a graph of the candidate recommendation. There is a correspondence between Music-CD in the recommendation and Query0075 in the request, between CompactDisc and Audio-Recording, and Title and Uniform-Title, respectively. The concepts in each of these correspondences inherit structure from shared concepts that are not shown in the figure; in each pair, however, one or both of the matched concepts contains roles that do not correspond to roles in the other concept.

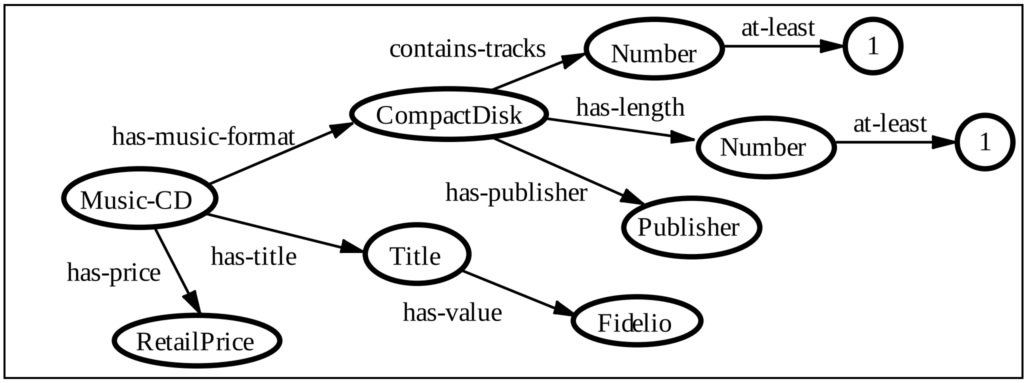


Figure 3: A recommendation candidate in a differentiated ontology

Figure 4 illustrates the similarities and differences between Audio-Recording and CompactDisc. In this figure, unlabeled edges with open arrows indicate subsumption, black ovals indicate shared definitional structure, medium grey identifies concepts in the request ontology, and light grey identifies concepts in the recommendation ontology. Publishing-Format is a shared inherited concept, with both the request and recommendation inheriting the associated role <has-publisher Publisher>. The two <has-length Number> roles are not shared—they have been added separately to local definitions in each of the differentiated ontologies—but, they are compatible with each other. The role <contains-tracks Number>, finally, is unique to the recommendation.

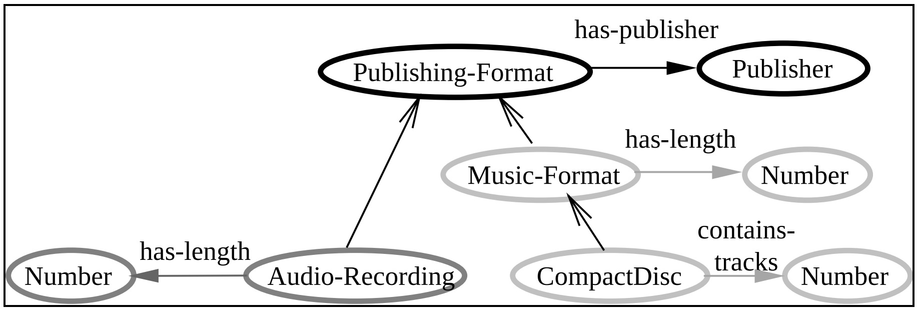


Figure 4: Shared and local definitional structure

1.2. Related Research

Mappings identify correspondences between items in two terminologies. Most research in this area assumes that mappings are developed manually, and focuses on how to represent them. Mappings may be represented as conditional rules (Chang and Garcia-Molina 1998), functions (Chawathe, Garcia-Molina et al. 1994; Baldonado, Chang et al. 1997), logic (Guha 1992; Catarci and Lenzerini 1993), or as sets of tables and procedures (Weinstein and Birmingham 1998). Declarative representations are advantageous: the Protégé system (Park, Gennari et al. 1998) organizes mappings themselves in an ontology, which helps users compose applications from ontologically-described components.

Unfortunately, manual specification of mappings is extremely expensive for real-world problems (Heiler 1995), partly because typical knowledge representations do not fully capture the intended semantics of the data (Ram and Ramesh 1999), and because of the complexity of semantic heterogeneity in multi-databases (Batini, Lenzerini et al. 1986; Kim and Seo 1991; Kashyap and Sheth 1996) and ontologies (Visser, Jones et al. 1997). Most often, mapping is between local terminologies and a global terminology, to avoid requiring a quadratic number of mappings. It is often quite difficult, however, to specify global terminologies that are adequate for all intended uses (Hull 1997).

Many research projects assume knowledge input in various representations from which mappings can be deduced, rather than assuming mappings themselves (Bayardo, Bohrer et al. 1996; Levy, Rajaraman et al. 1996; Knoblock and Ambite 1997; Benjamins et al. 1998). A number of systems assist human database experts (Larson, Navathe et al. 1989; Ram and Ramesh 1995; Gangemi, Pisanelli et al. 1998) or programmers (Papakonstantinou, Gupta et al. 1995; Milo and Zohar 1998) to integrate schemas.

There are preliminary attempts to generate mappings automatically. Li (Li 1995) identifies similarities between attributes from two schemas using neural networks. Campbell and Shapiro describe an agent that mediates between agents that subscribe to different ontologies (Campbell and Shapiro 1995). Bright et. al. (Bright, Hurson et al. 1994) use a thesaurus and a measure of “semantic-distance” based on path distance (an additive version of our “least-common subsumer” measure).

In a few papers, the quality of correspondences across schemas or ontologies is itself a focus. Kashyap and Sheth (Kashyap and Sheth 1996) define the “semantic proximity” of two concepts as a tuple encompassing contexts, value domains and mappings, and database states. The resulting analysis yields a hierarchy of types of semantic proximity, including equivalence, relationship, relevance, resemblance, and in a separate branch, incompatibility.

Lehmann and Cohn (Lehmann and Cohn 1994) require that concept definitions (“eggs”) include secondary, more specialized definitions for typical instances (“yolks”), and assume that the set relation between any two definitions can be identified as either equivalence, containment, overlap, or disjointness. Then, the eggs and yolks of two concepts can overlap in 42 different configurations, which the authors rank by the reliability of the mapping.

OBSERVER (Mena, Illarramendi et al. 1999) combines intensional and extensional analysis to calculate lower and upper bounds for the precision and recall of queries that are translated across ontologies on the basis of manually identified subsumption relations. For example, if a term is known to subsume its translation then Precision = 1. Bounds that are not known are calculated using concept extensions in local databases, without making any assumption about the degree to which the database contents overlap. Under these realistic conditions, unfortunately, the estimated intervals tend to be very broad when no intensional relationship between the term and its translation is known. In these cases, for example, the lower bound for Precision is often zero, while the upper bound is one.

In comparison to existing research, we make assumptions on several levels that let us develop and study algorithms that automatically predict semantic compatibility with an unprecedented degree of attention and detail. First, requiring formal ontological concept definitions means that we are working with knowledge representation intended to fully capture the relevant semantics of the data. Rather than working with unadorned symbols, which forces manual specification of mappings, we have definitions that can be represented as graphs and manipulated computationally.

Second, requiring differentiated ontologies establishes a common ground—shared inherited structure—that provides a basis for comparing concepts. Other approaches require shared instances (Lehmann and Cohn 1994), shared concepts (Campbell and Shapiro 1995), identifiable interschema or interontology relations (the various forms of subsumption) (Bright, Hurson et al. 1994; Ram and Ramesh 1995; Mena, Illarramendi et al. 1999), shared attribute characteristics (Larson, Navathe et al. 1989), or manually developed mappings across schemas or ontologies in a variety of representations. The Onions methodology (Gangemi, Pisanelli et al. 1998) integrates local ontologies by inheriting from shared generic ontologies, but does not automatically compare concepts. The advantages of differentiated ontologies derive from the power of inheritance for organizing complex domains.

Third, we evaluate the description-compatibility measures with simulations that simplify the structure of the world, and make all object instances accessible to all concepts as if the world were one great database. There are other approaches for generating feedback to evaluate description compatibility, but none are fully satisfactory. If concept meaning is defined by extensions in local databases under realistic assumptions (Mena, Illarramendi et al. 1999), the resulting feedback is so vague as to be vacuous. Alternatively, one might study agent interoperation in particular task contexts, and claim that appropriately coordinated behavior indicates effective communication. The deficiency of this approach is the required commitment to very particular scenarios. The disadvantage of our simulations is lack of proof regarding the generalizability of the empirical results. Therefore, the simulations do not supercede other approaches for evaluating description compatibility, but they do provide an excellent environment for developing and studying algorithms that can subsequently be tested under more realistic conditions.

Our measures of description compatibility can be used to identify mappings automatically. Mapping is direct, there is no global terminology. The quality of our mappings is quantified as a single real number, rather than ordinally in a quality hierarchy or as an interval. The accuracy of individual description-compatibility estimates is not guaranteed, however, except probabilistically. Our mappings are, in this sense, imprecise. Thus, we require that agents (or other system elements) be able to learn which mappings tend to be successful in particular contexts. We believe that this requirement, although task-specific in nature, will prove to be more amenable to automation than specification of precise mappings.

1.3. Overview

Conceptually, the problem of description compatibility involves interaction among three elements, as illustrated in Figure 5. The ontologies model the world, and are the basis for calculating description compatibility. The world determines semantic compatibility, and thus the accuracy of description compatibility. Each of the relationships in the figure are covered by one of the following sections. Section 2 specifies the assumptions embodied in our differentiated ontologies, involving the structure of the world, the ontological language, and the required relationship between the world and the ontological model of the world. Section 3 describes a selection of description-compatibility measures, including the algorithm used to identify matchings. Section 4 presents the simulation experiments. Space limitations force the presentation in Sections 2, 3, and 4 to be sketchy; the reader is referred to Weinstein (Weinstein 1999) for details. Section 5 discusses this research, and Section 6 gives conclusions.

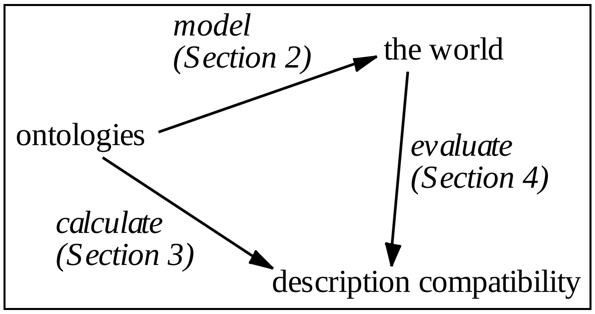


Figure 5: A conceptual model of the problem of description compatibility

2. ASSUMPTIONS

This section states assumptions about the ontologies and the world that are required to specify the measures of description compatibility. For reasons of space and readability, the presentation is informal; see Weinstein (Weinstein 1999) for formal notation. To implement the simulations requires additional restrictions, as described in Section 4.

2.1. Worlds

We model worlds as sets of objects that have relations to other objects. An object as it is represented in the ontologies is an "instance". Instances are vectors of attribute values, with associated relations to other instances. The attributes can be thought of as features that describe the structure or behavior of an object. Instances have the following fundamental characteristics:

- Values for different sets of attributes.
- Attribute values are correlated with certain others.
- Values constrain relations to other instances.

To illustrate this final characteristic, say that we are concerned with agent services in digital libraries (Weinstein and Birmingham 1998), and OperaVid0123 is an instance of a service that provides access to operatic videos. A certain attribute value of OperaVid0123 identifies it as a live-video access service. Then we expect

OperaVid0123 to have values for certain other attributes that describe different aspects of the service: for example, compression-format, delivery-rate, and so on. These values may be supplied by a relation to another instance.

2.2. Language

Description logic is appropriate for our simulations because of its formal model-theoretic semantics, and its ability to automatically classify definitions with subsumption inference. Many description logics have been developed, used, and studied from various perspectives (Woods and Schmolze 1992; Borgida and Patel-Schneider 1994; Cohen and Hirsh 1994; Koller, Levy et al. 1997; Horrocks 1998). We initially used Loom (MacGregor 1991) to implement the simulations, but finding it far too slow for this purpose, decided to implement a customized description logic designed for speed.

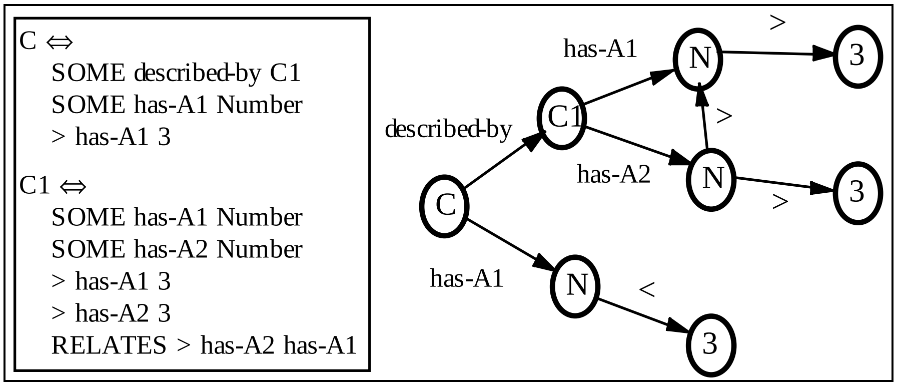
Our language, which we call Basic Existential Description Logic (BEDL), is a subset of Loom and of CLASSIC (Brachman, McGuinness et al. 1991). To use all of Loom or CLASSIC would have made our simulations too complex; also, we found that Loom was too slow for our purposes. We designed BEDL in accordance with our experience developing ontologies in complex domains (Weinstein and Birmingham 1998), and expectations for the sensitivities of the measures of description compatibility. Hence, we use existentially quantified relations rather than universals, because they have an intuitive semantics that make them desirable for practical work. We also include multiple inheritance among concepts, single inheritance among relations, and relations between role fillers that produce undirected cycles within concepts represented as graphs. We exclude less useful features (such as filler cardinality constraints) that would add complexity to the simulations without reason to expect an important impact on measures of description compatibility. BEDL is not itself meant to be a contribution of this work.

Figure 6 presents the syntax for non-primitive BEDL concepts. Primitive concepts, in comparison, are defined with necessary (⇒) but not sufficient (⇐) conditions. Also, BEDL concept definitions may not have type restrictions that reference the concept being defined, i.e., they may not be recursive. The RELATES clause specifies that the fillers of the second and third arguments are related by the relation specified in the first argument.

```
concept ⇔ concept-expr
concept-expr := concept-name |
                ( AND concept-expr*
                      ( SOME relation-expr concept-expr )*
                      ( { < | > | ≤ | ≥ | = } relation-expr number )*
                      ( RELATES relation-expr relation-expr relation-expr )* )
```

Figure 6: Syntax for defined concepts in BEDL

Concepts may also be represented as “description graphs” (Borgida and Patel-Schneider 1994; Cohen and Hirsh 1994). The graph in Figure 7 represents the definitions to the left, which are in the style of the abstract concepts used in the simulations. The vertices of the graph are concepts; the “root” vertex is associated with the concept being defined, and other vertexes are associated with type restrictions. Relations are associated with edges, and “roles” includes a relation and a type restriction.



```
C ⇔
   SOME described-by C1
   SOME has-A1 Number
   > has-A1 3
C1 ⇔
   SOME has-A1 Number
   SOME has-A2 Number
   > has-A1 3
   > has-A2 3
   RELATES > has-A2 has-A1
```

Figure 7: A concept graph defining concept C

Objects are instances of a concept if they are related to other objects such that every element of the concept graph has a corresponding element in the graph of objects. This does not necessarily require subgraph isomorphism, since a single element of the object graph can satisfy multiple constraints in the concept graph.

2.3. Modeling the World

To calculate description compatibility using an ontological structure requires that there be a minimal correspondence between the ontologies and the world. Relations between objects (“world relations”) must correspond to primitive ontology relations. Also, if the description compatibility measures are to have any predictive value at all, proximity of relations in the ontologies should correspond to covariance of attributes in the world. For example, in Figure 3 of Section 1, has-length and contains-tracks are relations in sibling roles of the concept CompactDisc, and their filler values might be expected to covary, whereas has-price is not a close relative to these relations and is thus modeled as being independent, or at least less strongly covariate.

Finally, there are two differences between differentiated ontologies, and a single ontology stored in distributed locations. First, differentiated ontologies may have naming conflicts. While often misleading, shared names are potentially a source of information that our description-compatibility measures do not exploit. Second, there may be relations across differentiated ontologies that are entailed, but not proved. In this sense, the differentiated ontologies are not integrated.

3. MEASURES OF DESCRIPTION COMPATIBILITY

This section presents a few of the most interesting description-compatibility measures, selected from dozens that we have experimented with. We categorize the measures into three groups. The “filter” functions are inexpensive, universally applicable, and are appropriate for identifying initial sets of candidate recommendations. The “matching-based” functions build and evaluate one-to-one correspondences between elements of concept definitions represented as graphs. The “probabilistic” functions require domain-specific knowledge of the joint distribution of primitives.To calculate these measures in real-world systems would require using Bayesian networks or other techniques to model the joint distribution.

The filter, matching-based, and probabilistic measures cross a spectrum with respect to the degree to which knowledge is required of the semantics of roles within concept definitions. Roughly, the filter measures do not involve role semantics; the matching-based measures use knowledge of roles considered independently; and the probabilistic measures exploit knowledge of the interactions among roles.

The description-compatibility measures are designed to predict semantic compatibility. In the simulations, we operationalize semantic compatibility as the probability that a random instance (j) of service recommendation (D) will satisfy a request (C) under logical interpretation (*I*) (see Figure 8):

$$\begin{aligned}\mathrm{SEMR}(C, D) &= \Pr(j \in C^{I} \mid j \in D^{I}) \qquad \text{(Eq 1)}\\ &= \Pr(j \in C^{I} \cap D^{I}) / \Pr(j \in D^{I})\\ &= |C^{I} \cap D^{I}| \;/\; |D^{I}| \,.\end{aligned}$$

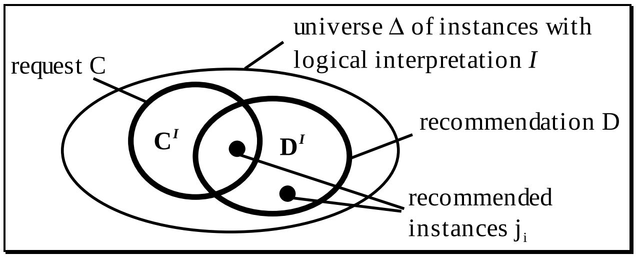


Figure 8: A semantic standard for description compatibility

“SEMR” is mnemonic for “SEMantic compatibility in the Recommendation scenario”. SEMR is an asymmetric measure analogous to the Information Retrieval concept of precision. In some practical contexts, a symmetric measure of semantic compatibility may be more appropriate, where the denominator is the union of the request and recommendation. The labels for all of the description-compatibility measures defined below include an “R” at the end to indicate that they are suitable for predicting SEMR.

3.1. Filter Measures

The “Least Common Subsumer” measure of description compatibility (LCSR) is a function of path distance to the most proximate concept that subsumes both the request and recommendation. Let $L_1,...L_n$ be the set of concepts that subsume C and D. Let $|p(C, L_i)|$ be the number of links necessary to transverse the graph of inheritance links from C to L_i. Then

$$\mathrm{LCSR}(C, D) = \gamma^{\min_i (|p(C, L_i)|)} \quad , \qquad \text{(Eq 2)}$$

where γ is an "attenuation factor” in the interval (0, 1) that models the expected percentage of instances of a parent concept that are also members of a child concept. Path-distance measures such as LCSR have a long history in artificial intelligence, but they are fragile because they are sensitive to the degree of detail in the ontological structure. For example, classifying new concepts can change LCSR estimates although the compared concepts have not changed.

The remaining filter measures leverage the assumption that local concepts in differentiated ontologies inherit definitional structure from concepts in shared ontologies. Like LCSR, these measures use inheritance links to identify concepts that are the ancestors of both the request and recommendation. Unlike LCSR, they also utilize the definitions of the source and target concepts. These filter measures are thus far more robust than path-distance measures.

The Shared Edges (SHER) measure is the proportion of edges in the request concept’s description graph that inherit from edges in concepts that are shared, and that subsume both the request and the recommendation. Let $L_1,...L_n \in P'$ be the set of most specific concepts that subsume the request C and recommendation D, i.e., there is no i, j such that L_i subsumes L_j. Then, create a new *shared-inherited concept:*

$$A \equiv (L_1 \text{ AND ... AND } L_n) \qquad \text{(Eq 3)}$$

that merges the roles of L_i. Then

$$\text{SHER}(C, D) = |E(G(A))| / |E(G(C))| \qquad \text{(Eq 4)}$$

where |E(G(X))| is the number of edges in concept X's description graph.

The next measures differ from SHER only by their focus on selected subsets of edges. The Shared Rooted roles (SHRR) measure considers only "rooted" edges: those whose first vertex is the root of the description graph. These edges are associated with roles in the concept's definition, rather than within the definitions of its type restrictions. To calculate SHRR, construct the shared-inherited concept (A) as for SHER. Then, let |RR(A)| be the number of rooted roles in A, and let |RR(C)| be the number of rooted roles in the request concept C. Then:

$$\text{SHRR}(C, D) = |RR(A)| / |RR(C)| \;. \qquad \text{(Eq 5)}$$

Finally, the Shared Value-Constraining Edges (SHVR) measure focuses exclusively on edges that directly constrain attribute values. These edges have the greatest (independent) impact on the extensions of the concepts. Value-constraining edges are leaves in the description graphs of concepts (keeping in mind that concept graphs are not necessarily trees). The number of value-constraining edges is thus highly correlated with the number of other types of edges, such as those that relate multi-attribute objects, since these other edges constitute the trunks and branches that lead to the value-constraining edges. Inheriting value-constraining edges occurs only in the context of these other edges, which constrain the meaning of the value-constraining edges. To calculate SHVR, create the shared-inherited concept A as above. Then:

$$\text{SHVR}(C, D) = |VC(G(A))| / |VC(G(C))| \qquad \text{(Eq 6)}$$

where |VC(G(X))| is the number of value-constraining edges in concept X's description graph.

3.2. Matching-based Measures

Matchings place elements of two concept definitions into one-to-one correspondence. They provide a means to analyze similarities and differences between two definitions. This section describes how we construct matchings, how we analyze matchings, and how we use them to predict semantic compatibility.

3.2.1. Matching

We represent matchings as graphs of matches, where a *match* is a pairing of a concept (or relation, or edge) from the request graph with a concept (relation, or edge) from the recommendation graph. The correspondence must be one-to-one, but is typically not total: the problem is to match subgraphs of the request and recommendation so as to maximize the total compatibility captured by the matching.

This problem, which we call Rooted Weighted-Edge-Match Graph Matching, is NP-complete (Weinstein 1999). We use a greedy algorithm to produce high-quality matchings. This algorithm starts with the root match, and repeatedly extends the matching by adding the most-compatible edge match whose edges are incident on a match currently in the matching. If the new edge conflicts with other reachable edges, the matching is cloned, and each inconsistent edge is added to one of the copies. Tractability is maintained by limiting the number of alternative matchings: when the numberof alternatives exceeds a threshold, the matchings are ranked, and those with the lowest ranks are dropped. If the threshold is not exceeded then the solution is guaranteed to be optimal.

Edge compatibility is the product of the concept compatibility of the vertex match not already in the matching, and the compatibility of the relations linking the matched vertices. For concept compatibility, we can use any of the measures described in this section. For relation compatibility, we use an LCS measure that prevents matches across different clusters, thus refining the model of clusters embodied in the ontology.

It is possible to convert matchings to concepts. These “matching-derived concepts” include all definitional structure shared by the compared concepts.

3.2.2. Matching Analysis

It is useful to categorize matches within matchings into several groups:

- Inherited matches. These matches include, exclusively, definitional structure inherited from shared concepts that subsume the request and recommendation. The edge compatibility of inherited matches equals 1.0.
- Specialized matches. In these matches, at least one of the concepts or relations in the edge match has been specialized in a local ontology. The edge compatibility of specialized matches is less than 1.0.
- Serendipitous matches. The concepts in these matches do not inherit from shared concepts that subsume the request and recommendation; they represent definitional structure that was added locally in each of the differentiated ontologies.This correspondence may be inherited from other shared concepts, but this is not necessarily so (unless we consider virtual concepts; i.e., concepts that could potentially be defined and classified).

Inherited matches represent pure similarity. Specialized and serendipitous matches include a mixture of similarity and difference that requires some extra processing to tease apart. We also identify two other kinds of differences:

- Non-correspondence: definitional structure that does not correspond to structure in the other concept: i.e., edges in the request that are unmatched.
- Conflict: when constraints are incident on matched vertices that demand values in the request and recommendation, respectively, that tend to be mutually exclusive. For example, it is almost impossible to find healthy humans that are over six feet tall and that weigh less than 100 pounds. Identifying conflicts requires knowledge about the domain, and involves reasoning about the semantics of the involved constraints that may be arbitrarily sophisticated. By definition, conflict reduces the intersection of concept denotations: we might expect, therefore, that it is a potent predictor of semantic compatibility.

3.2.3. Selected Measures

There are many ways to use matchings to predict semantic compatibility. The following measures illustrate some of the possibilities.

The Sum of Match Scores Penalized (SMSPR) measure combines several elements. First, it sums the compatibility scores of all edge matches. Second, it reduces this total score by a factor that exponentiates a constant (smaller than one) for each unmatched value-constraining edge in the request. The assumption here is that each of these unmatched edges reduces the request’s extension by a constant multiplicative factor. Third, it reduces the score by a factor that penalizes for conflicts. Thus:

$$\mathrm{SMSPR}(C, D) = \Sigma_{e \in E(M(C,D))} \lambda(e) * \upsilon^{|VC(G(C))| - |VC(M(C, D))|} * Q(M(C, D)) \quad \text{(Eq 7)}$$

where C is the request concept, D is the recommendation, λ is the edge-compatibility function, VC(G(C)) is the set of value-constraining edges in the description graph G for concept C, VC(M(C, D)) is the set of value-constraining edge matches in the matching between C and D, and $\upsilon \in (0, 1)$ is a constant selected empirically.

We calculate penalty factors for conflicting constraints as:

$$Q(M(C, D)) = \Pi_{v2(e) \in Vq(M(C, D))} (1 - \lambda(e)) \qquad \text{(Eq 8)}$$

where $V_q(M(C, D))$ is the set of node matches in the matching M(C, D) where there is a contradiction among the numeric constraints incident on the match, v2(e) is the second concept match in an edge match, e, and $\lambda(e)$ is the compatibility of the edge match. Figure 9 motivates this equation by illustrating a conflict where the values of attributes A and B covary positively. The boxes represent matches; the numeric constraints are not matched because '>' is incompatible with '<'. If the relation match was perfect, then the conflict would be a contradiction. As it is, the degree of the conflict depends on the covariance of A and B, a quantity that is estimated by the compatibility score of the relations has-attribute-A and has-attribute-B.

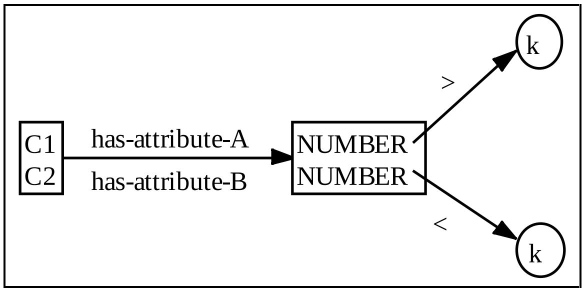


Figure 9: Conflicting constraints incident on a match

The Counts of Similarities and Differences Penalized (CSDPR) measure includes all of the analytical categories described in Section 0. It is constructed in the spirit of statistical prediction as a linear regression, except that the conflict factor, which is non-linear and non-linearizable, is applied in a multiplicative manner rather than additively:

$$\text{CMSPR} = (\beta_0 + \beta_1(\text{NMI}) + \beta_2(\text{NMS}) + \beta_3(\text{NMR}) + \beta_4(\text{NUNMV})^{\alpha}) \qquad \text{(Eq 9)}$$
$$* \; Q(M(C, D))$$

where NMI is the number of inherited matches, NMS counts specialized matches, NMR counts serendipitous matches, NUNMV counts unmatched value-constraining edges in the request, and α adjusts for the non-linear effect of unmatched edges. The constant β_0 is required to ensure non-negativity.

Finally, here is a relatively simple measure that avoids the need to estimate parameters, the Shared Rooted-roles Penalized (SHRPR) measure. This measure combines the approximation of shared-inherited structure and unmatched edges captured by the SHRR measure with conflict identification:

$$\text{SHRPR}(C, D) = \text{SHRR}(C, D) \; * \; Q(M(C, D)) \qquad \text{(Eq 10)}$$

3.3. Probabilistic Measures

Semantic compatibility is a probabilistic notion: the goal is to predict P(C|D), where C and D are concepts defined in differentiated ontologies. The basic idea behind the probabilistic description-compatibility measures is to emulate, as closely as possible, the equation for semantic compatibility. This is possible if we assume more information—enough to model the semantic interaction of roles within concept definitions, or, more precisely, the joint distribution of primitive concepts or relations (some promising research on integrating Bayesian networks and frame-based representations is now available (Koller, Levy et al. 1997; Pfeffer, Koller et al. 1999)). Such a model can be used to estimate *membership probability:*

$$\hat{P}r(C) \approx Pr(j \in C^I) \qquad \text{(Eq 11)}$$

where C is a concept, j is an arbitrary object, and ***I*** is a logical interpretation.

The measures described in this section combine membership probabilities in different ways. The Probabilistic Conjoined Intersection (PCIR) measure is a faithful analog of the semantic compatibility measure SEMR (compare to Equation 1):

$$PCIR(C, D) = \hat{P}r(C \text{ AND } D) / \hat{P}r(D) \qquad \text{(Eq 12)}$$

where (C AND D) is a concept defined as the conjunction of C and D. Unfortunately, PCIR is realistic only for source and target concepts in the same ontology, because it is highly unlikely that any model of the joint distribution of primitives would be available for concepts and relations defined in differentiated ontologies.

Therefore, we revisit the approaches of the filter and matching-based measures, and focus on shared definitional structure and structure that can be put into correspondence, respectively. The Probabilistic Shared Structure measure (PSHR) is defined as:

$$PSHR(C, D) = \hat{P}r(C) / \hat{P}r(A) \qquad \text{(Eq 13)}$$

where A is the shared-inherited concept defined in Equation 3. The Venn diagram in Figure 10 illustrates this calculation. Intuitively, the more shared structure inherited by C and D, the more closely A^I will encircle C^I and D^I; and in the limiting case where $C^I = D^I$, $A^I = C^I$ will also be true. More precisely, the equation for PSHR derives from an assumption that C and D are conditionally independent given A, an assumption that is reasonable if we believe that A embodies everything we can know about the relationship between C and D.

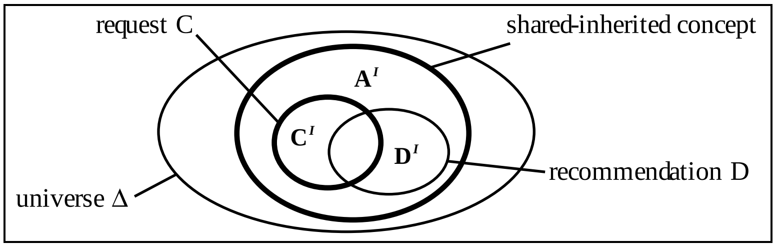


Figure 10: The semantic basis of PSHR

We define the Probabilistic Matching-Based Penalized (PMBPR) measure as:

$$PMBPR(C, D) = \hat{P}r(C) * Q(M(C, D)) / \hat{P}r(MC(C, D)) \qquad \text{(Eq 14)}$$

where Q(M(C, D)) is the conflicts penalty factor for matching M(C, D), and MC(C, D) is the concept derived from the matching M(C, D). PMBPR is different from PSHR in two respects. It includes the conflict identification factor, Q, and uses the matching-derived concept rather than the shared-inherited concept, thus accounting for serendipitous as well as inherited similarity.

4. EXPERIMENTS

The simulation experiments depend on many parameters and other implementation decisions. These affect the generation of mapping data at each stage, including the creation of objects, ontology generation, and calculation of description-compatibility for a set of concept comparisons. Due to space limitiations, in this paper we present results for a single configuration (“Experiment 1”) and a few variations on that experiment. We focus, however, only on fundamental results that persist across multiple experiments.See Weinstein (Weinstein 1999) for more coverage.

4.1. Ontology Generation

The first step in running a simulation experiment is to generate a world of objects. This necessarily involves additional commitments besides those described in Section 0. The simulated worlds must have some structure that the ontologies can model, and the description compatibility measures can exploit to produce interesting behavior.

Objects in Experiment 1 each have the structure pictured in Figure 11. They consist of five clusters with three attributes each, of which three clusters have values. Attribute values are distributed normally, with positive covariance within clusters, and none across clusters. As explained in Section 0, values should constrain relations among objects. We model this property of object values by interpreting values as cluster indices; these constrain links among objects by requiring that objects linked to have attributes in the indexed cluster.

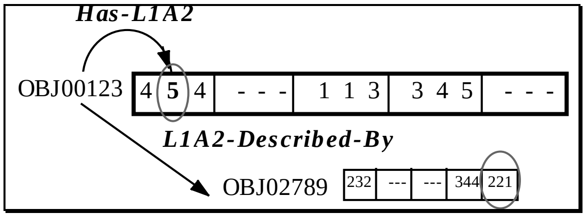


Figure 11: Object relations to values and to other objects

A single, simple idea, in conjunction with the syntax of BEDL, determines our process for automatically generating concepts to describe objects. We define new concepts so as to distinguish selected objects from other objects (see Figure 12).

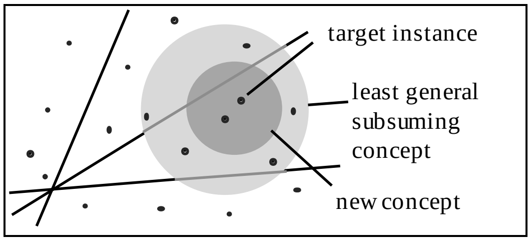


Figure 12: New concepts distinguish objects from others

For each concept, the generation process includes the following steps:

1. Select an instance as a target.
2. Pick a least-general concept that includes the target to be the “parent”.
3. Define the new concept by specializing the parent definition with one of the following methods:
    - Multiple inheritance (conjoin two or more parents).
    - Add a value (or type) relation role.
    - Further restrict a value (or type) filler.
    - Add a constraint between value (or type) fillers.

4. If the method chosen in step 3 does not distinguish the target object from enough of the parent's instances, try again with another parent concept. If the method fails for all possible parents try another method, and so on.
5. Check all of the parent's instances to recognize instances of the new concept.

We generate concepts in a tree of ontologies, where each ontology on a branch favors using an increasingly restricted subset of attributes to specialize its concepts. Subsequent to concept generation, we delete many concepts from the interior of the ontologies (the parameter that controls deletion is the "thinning percentage"). This yields a structure with greater variation, intensionally and extensionally, between concepts and their parents.

Experiment 1 includes 1327 concepts in 25 differentiated ontologies. Concepts have an average of 1.9 value-relation roles, 1.6 type-relation roles, 11.7 edges total (excluding inverses) in their description graphs, and 2.9 ancestors on their longest path to the generic (topmost) concept. To reduce noise, we accumulated membership and concept intersection counts for 500,000 instances. We generated over 20,000 concept comparisons for 490 agent searches for services, including source and target concepts from 42 pairs of ontologies. The SHRR measure is used to estimate concept compatibility when building matchings; value-constraining edges are those with numeric relations; and all comparisons pass an initial filter that checks for some shared-inherited structure.

4.2. Results

This section describes the results of the simulation experiments on two levels. First, on the level of individual concept comparisons, we examine the statistical association of description compatibility and semantic compatibility. We include analysis of the contribution of matching to prediction accuracy. Second, we analyze the results on the level of searches: sets of comparisons where the request concept is the same. The question is how good the various description-compatibility measures are at identifying the best candidate out of those available; in other words, whether description compatibility can be used to increase the efficiency of search.

When analyzing simulations, the role of statistical significance is diminished, since the number of observations is large and can be made arbitrarily larger. All correlations reported in this section are statistically significant at a very high level of confidence.

Table 1 shows the association between each description-compatibility measure and SEMR. Ordinary Pearson's correlations are highly sensitive to extreme data points and tend to be erratic across our experiments, so we use Spearman's ranked correlations to analyze the data instead, although the coefficients tend to be substantially lower.

Table 1: Associations of description and semantic compatibility measures

N=21,815 Category	Method	Ranked Correlation
Path-distance	LCSR	.164
Shared-inherited structure	SHER	.218
	SHRR	.380
	SHVR	.308
Matching-based	SHRPR	.552
	SMSPR	.535
	CSDPR	.548
Probabilistic	PSHR	.559
	PMBPR	.705
	PCIR	.963

Relative to each other, the accuracy of the description-compatibility measures is roughly what one might expect. The path-based LCSR measure is the weakest. Without any "thinning", LCSR does about as well as the other filter measures, but degrades as thinning increases. The filter measures based on shared inherited structure are stronger than LCSR. The SHVR measure, which counts only value-constraining edges, is significantly stronger than SHER, which counts all types of edges. The SHRR measure, which uses only rooted edges and is thus the easiest to calculate, is more accurate with a ranked correlation of 0.38. The non-probabilistic matching-based measures achieve association that is clearly stronger, and are all essentially equal to each other with corrrelations around 0.54. The probabilistic matching-based measure is quite strong, and the (unrealistic) conjoined intersection measure is best of all.

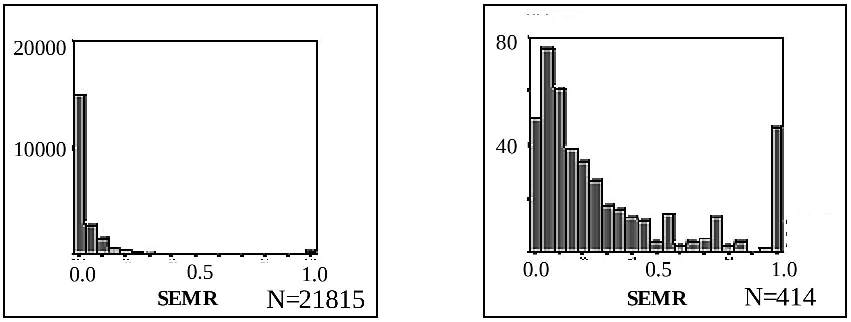
The shared-inherited structure measures show that more shared structure predicts greater semantic compatibility. There is also evidence for a second-order effect: if samples of comparisons are filtered by a test for shared inherited structure, there is a weak but noticeable increase in the predictive accuracy of the compatibility measures. Most of this increase, however, probably derives from the increase in variance of SEMR when subject to filtering. Usually, filtering decreases variance. Here, however, SEMR is bounded below by zero, and its mean increases with filtering, leading to increased variance as well.

The linear regressions used to estimate the parameters for the CSDPR measure (on different mapping data to avoid overfitting) also provide a way to judge the relative importance of the analytical elements of matchings for predicting semantic compatibility. Unmatched edges have the biggest impact, with a standardized coefficient of -.434. Inherited matches have less than half the impact, with beta=.203. According to the linear model, the contribution of identifying conflicts is about equal to that of inherited matches (in the other direction). The linear model clearly underestimates the impact of conflicts in our data, however, as is apparent from inspection of the regression, and the large improvement in accuracy when non-matching-based measures are multiplied by the penalty factor.

Many concept compatibility estimates are required in the process of constructing matchings. It would be desirable to use the relatively inexpensive filter measures for this purpose if the final results are not significantly harmed. Fortunately, comparisons using matches built with the LCSR, SHRR, SMSPR, and PCIR measures all achieve accuracy that is essentially equal in accuracy. Although the matchings frequently differ in the detail of their structure, all of the measures are successful in identifying the matches that matter most for predicting semantic compatibility.

The most dramatic case for the usefulness of description compatibility can be made with analyses on the level of searches: sets of comparisons each for a particular request. For each search, we rank comparisons by their SEMR value to obtain an ordered set of "candidate" services. We also rank comparisons by each of the compatibility measures to obtain ordered sets of "recommended" services. There are several ways to rank comparisons that have equal values, and unfortunately, the method used can have a substantial impact on the reported results. Therefore, we rank comparisons in several ways and combine the results as needed to answer questions of practical interest.

Experiment 1 includes 490 searches. Of these, we drop searches with less than 10 comparisons (due to initial filtering), leaving 414 searches averaging 52 comparisons per search. Figure 13a shows that the large majority of comparisons are quite poor, with very small semantic compatibility. The semantic compatibility for the best candidates is much improved, of course, as shown in Figure 13b. The average semantic overlap of the best candidates, however, is still only 0.3.



(a) all comparisons (b) best candidates in each search

Figure 13: Histograms of candidate quality

Figure 14 considers the question appropriate for situations where the quality of the top-ranked recommendation is what matters: it shows the probability that the first recommendation will have the highest semantic compatibility, or be within the top half of candidates. The first recommendation is selected randomly from among candidates with the best score (the prevalence of ties in semantic compatibility causes the location of the data points along the x-axis). For the matching-based measures SMSPR and CSDPR, the first recommendation is usually pretty good; almost half of the time it identifies a candidate tied for best, and almost two-thirds of the time the top recommendation is in the top ten percent of candidates. The probabilistic matching-based measure PMBPR does even better. The other measures are less strong, partly because they are less discriminating (many recommendations tend to be tied for first).

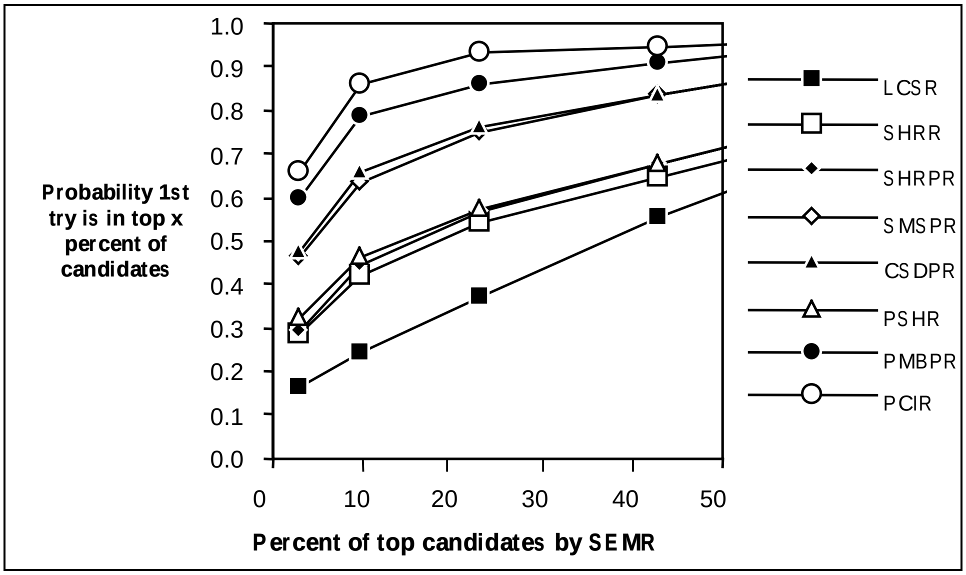


Figure 14: Quality of first recommendation

Figure 15 addresses situations where the system can try multiple candidates. It answers the question of how many recommendations one should expect to try before finding the best candidate. Here, many measures do quite well. The CSDPR measure has a better than even chance of recommending the best candidate with a recommendation tied for first, and over a 70 percent chance that the best candidate is within the top 15 percent of recommendations. Again, PMBPR does even better, and the filter measures do less well. Even the easily calculated shared-inherited measure SHRR, however, may be useful in many situations, as two-thirds of the time the best candidate is within the top 15 percent of recommendations (all of which are tied for first).

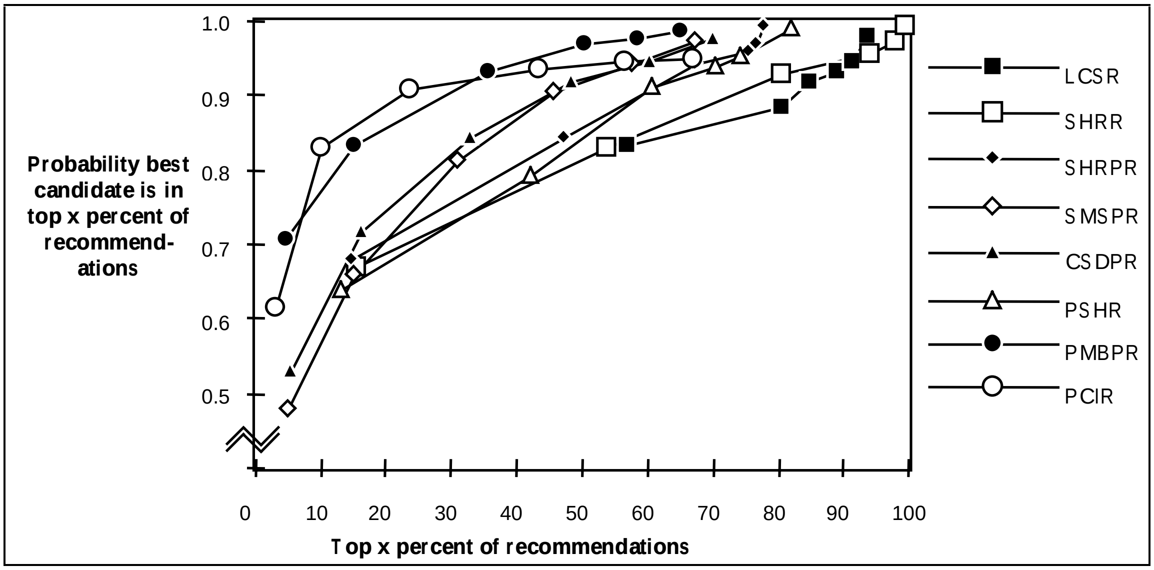


Figure 15: Cumulative probability of recommending the best candidate

5. DISCUSSION

The simulations provide an effective environment for developing algorithms for description compatibility and studying their behavior, at the cost of greatly simplifying semantic heterogeneity as it would occur in practice, even subject to the requirements of differentiated ontologies. Unfortunately, it is difficult to be precise about the degree to which the simulations diverge from reality, since differentiated ontologies have not yet been purposely developed as such (although numerous ontologies exist and one might be able to find some that appear to be differentiating).

We believe that the specific definition of BEDL is not critical for our findings. Basic Description Logic (BDL) (Borgida and Patel-Schneider 1994) may be considered a well-studied subset of description logic. It includes universal roles and cardinality constraints, but not inheritance among relations. Interestingly, BDL ontologies would model the simulation worlds in a way equivalent to the BEDL simulations, and would thus produce equivalent results regarding description compatibility. Essentially, cardinality constraints would be represented in a manner structurally equivalent to the current numeric constraints, with covariance among cardinalities equivalent to the current covariance among values.

We are more concerned that our world objects are too orderly. To attempt a realistically varied domain, however, would yield simulations that were uninterpretable. Because there are numerous ways to add complexity to the simulations, and because it is desirable to keep them simple, we added complexity only where there is a specific reason to anticipate an effect on the accuracy of the description-compatibility measures. Typically, we observed behavior that was less sensitive than we expected. For example, covariance of attribute values does contribute to the predictive benefit of matching, but its effect is minor because 80% of matchings do not include matches of different attributes.

Our greatest concern is that the simulations include unrealistic assumptions that remain unrecognized. In particular, we do not compare concept definitions that include unshared atomic primitives. Unshared atomic primitives violate the basic assumptions of differentiated ontologies (that formal definitions capture the intended semantics of data, and share inherited structure), but these assumptions apply on the level of the concepts being compared rather than to every concept within the definitions of the compared concepts. Meanwhile, the semantics of our definitions are grounded in numeric constraints whose meaning is shared, as embodied by a relation-compatibility function that calculates the intersection of values that can satisfy the compared constraints. To test whether we are taking “unfair” advantage of shared semantics that in practical contexts may not be shared, we ran a version of Experiment 1 where all matches not based on shared-inherited structure were excluded from the matchings. The resulting decreases in the accuracy of the matching-based description-compatibility measures were not significant.

Ultimately, proof of semantic interoperability must be provided by working systems. Description compatibility measures will predict the probability of satisfactory agent service, or of a useful query response. The accuracy of these measures must then be evaluated by comparison to pragmatic measures: feedback following actual use, including analysis of the frequency of false negatives and false positives. Ideally, machine induction will be used in these contexts to learn situations where description-compatibilty estimates err. We expect that large errors will usually be due to unidentified conflicts, which can be remedied by improving the functions that reason about matchings.

6. CONCLUSIONS

Differentiated ontologies support communication in distributed systems subject to semantic heterogeneity. Concepts are formally defined in relation to other concepts, such that concepts in local ontologies inherit definitional structure from concepts in ontologies that are shared. A variety of algorithms can compare concept definitions to predict their semantic compatibility.

The results of simulations that evaluate these algorithms show that they are effective. The description-compatibility measures predict semantic compatibility with accuracy consistent with their requirements in terms of input and computation. Several measures quantify the proportion of definitional structure inherited from shared concepts. As such, their correlations with the semantic measure (SEMR) are succinct summaries of the leverage provided by differentiated ontologies, and, that correlation is moderately strong (SHRR has a ranked correlation of 0.38 in Experiment 1, while the relatively unstable unranked correlation is 0.64). Matchings enable analysis of similarities and differences between the compared concepts, yielding measures that are significantly stronger, with ranked correlations over 0.5. Most of the improvement for matching-based measures derives from identification of conflicts between the request and recommendation definitions. The probabilistic measures require knowledge of the domain and the semantics of relations in concept definitions. With this additional input, it is possible to achieve very accurate predictions of semantic overlap: the probabilistic matching-based measure PMBPR, for example, enjoys ranked correlations of 0.7 or better. The performance of all of our algorithms, except for the path-based LCSR measure, is independent of the “thinning percentage”, which determines the homogeneity of extension reduction when descending in the ontological structure towards increasingly specialized concepts. This confirms our basic intuition that shared inherited definition structure supports estimation of description compatibility.

The simulations suggest that description compatibility can significantly increase the efficiency of search in several important contexts, including translation of queries across differentiated schemas, searches for reusable components, and agent searches for service when forming teams to satisfy user requests. These results indicate that differentiated ontologies are worthy of study in practical contexts, where the accuracy of description-compatibility measures can be evaluated with feedback from users engaged in solving problems.

7. ACKNOWLEDGMENTS

Mike Wellman and Ed Durfee provided valuable suggestions for conducting this research, which was supported in part by the NSF/ARPA/NASA Digital Library Initiative under grant CERA IRI-9411287, and partially by the Congregating Agents project, NSF grant IIS-9872057.

8. REFERENCES

Baldonado, M., C.-C. K. Chang, L. Gravano and A. Paepcke (1997). “The Stanford Digital Library Metadata Architecture.” International Journal of Digital Libraries 1(2): 515-521.

Batini, C., M. Lenzerini and S. B. Navathe (1986). “A Comparative Analysis of Methodologies for Database Schema Integration.” ACM Computing Surveys 18(4): 323-364.

Bayardo, R. J. J., W. Bohrer, R. Brice, A. Cichocki, J. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh and D. Woelk (1996). InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. Technical report MCC-INSL-088-96, MCC, Austin, TX.

Benjamins, V. R., E. Plaza, E. Motta, D. Fensel, R. Studer, B. Wielinga, G. Schreiber, Z. Zdrahal and S. Decker (1998). IBROW3: An Intelligent Brokering Service for Knowledge-Component Reuse on the World-Wide Web. Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW'98), Banff, Alberta.

Borgida, A. and P. F. Patel-Schneider (1994). “A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Language.” Journal of Artificial Intelligence Research 1: 277-308.

Brachman, R. J., D. L. McGuinness, P. F. Patel-Scheider, L. A. Resnick and A. Borgida (1991). Living with CLASSIC: When and How to Use a KL-ONE-Like Language. Principles of Semantic Networks. J. F. Sowa. San Mateo, California, Morgan Kaufmann: 401-56.

Bright, M. W., A. R. Hurson and S. Pakzad (1994). “Automated Resolution of Semantic Heterogeneity in Multidatabases.” ACM Transactions on Database Systems 19(2): 212-253.

Campbell, A. E. and S. C. Shapiro (1995). Ontologic Mediation: An Overview. IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal.

Catarci, T. and M. Lenzerini (1993). “Representing and Using Interschema Knowledge in Cooperative Information Systems.” Journal of Intelligent and Cooperative Information Systems 4(2): 375-398.

Chang, C.-C. K. and H. Garcia-Molina (1998). Conjunctive Constraint Mapping for Data Translation. Third ACM Conference on Digital Libraries, Pittsburgh.

Chawathe, S., H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman and J. Widom (1994). The TSIMMIS Project: Integration of Heterogenous Information Sources. IPSJ Conference, Tokyo, Japan.

Cohen, W. W. and H. Hirsh (1994). “The Learnability of Description Logics with Equality Constraints.” Machine Learning 17(2/3): 169-199.

Gangemi, A., D. M. Pisanelli and G. Steve (1998). Some Requirements and Experiences in Integrating Terminological Ontologies in Medicine. Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW'98), Banff, Alberta.

Guha, R. V. (1992). Contexts: A formalization and some applications. Ph D. Thesis, Department of Computer Science, Stanford.

Heiler, S. (1995). “Semantic Interoperability.” ACM Computing Surveys 27(2): 271-273.

Horrocks, I. R. (1998). Using an Expressive Description Logic: FaCT or Fiction? Sixth International Conference on the Principles of Knowledge Representation and Reasoning.

Hull, R. (1997). Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. Symposium on Principles of Database Systems (PODS-97), Tuscon, Arizona USA.

Kashyap, V. and A. Sheth (1996). “Semantic and Schematic Similarities between Database Objects: A Context-based approach.” International Journal on Very Large Data Bases 5(4): 276-304.

Kim, W. and J. Seo (1991). “Classifying Schematic and Data Heterogeneity in Multidatabase Systems.” IEEE Computer 24(12): 12-18.

Knoblock, C. A. and J. L. Ambite (1997). Agents for Information Gathering. Software Agents. J. Bradshaw. Menlo Park, CA, AAAI/MIT Press: 1-27.

Koller, D., A. Levy and A. Pfeffer (1997). P-Classic: A tractable probabilistic description logic. Fourteenth National Conference on Artificial Intelligence (AAAI-97): 390-397,, Providence, RI.

Larson, J. A., S. B. Navathe and R. Elmasri (1989). “A Theory of Attribute Equivalence in Databases with Application to Schema Integration.” IEEE Transactions of Software Engineering 15(4): 449-463.

Lehmann, F. and A. G. Cohn (1994). The EGG/YOLK Reliability Hierarchy: Semantic Data Integration Using Sorts with Prototypes. Third International ACM Conference on Information and Knowledge Management (CIKM-94), New York, ACM Press.

Levy, A. Y., A. Rajaraman and J. J. Ordille (1996). Querying Heterogeneous Information Sources using Source Descriptions. 22nd VLDB Conference, Mumbai (Bombay), India.

Li, W.-S. (1995). Knowledge Gathering and Matching in Heterogeneous Databases. AAAI Spring Symposium on Information Gathering from Distributed, Heterogenous Environments.

MacGregor, R. M. (1991). The Evolving Technology of Classificiation-based Knowledge Representation Systems. Principles of Semantic Networks. J. F. Sowa. San Mateo, California, Morgan Kauffman: 385-400.

Mena, E., A. Illarramendi, V. Kashyap and A. P. Sheth (1999). “OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies.” Distributed and Parallel Databases (to appear).

Milo, T. and S. Zohar (1998). Using Schema Matching to Simplify Heterogeneous Data Translation. Very Large Databases (VLDB).

Papakonstantinou, Y., A. Gupta, H. Garcia-Molina and J. Ullman (1995). A query translation scheme for rapid implementation of wrappers. Fourth International Conference on Deductive and Object-Oriented Databases, National University of Singapore, Singapore.

Park, J. Y., J. H. Gennari and M. A. Musen (1998). Mappings for Reuse in Knowledge-based Systems. Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW'98), Banff, Alberta.

Pfeffer, A., D. Koller, B. Milch and K. T. Takusagawa (1999). SPOOK: A system for probabilistic object-oriented knowledge representation. Uncertainty in Artificial Intelligence.

Ram, S. and V. Ramesh (1995). “A Blackboard-Based Cooperative System for Schema Integration.” IEEE Expert 10(3): 56-62.

Ram, S. and V. Ramesh (1999). Schema Integration: Past, Present, and Future. In Management of Heterogeneous and Autonomous Database Systems. A. Elmagarmid, M. Rusinkiewicz and A. Sheth. San Francisco, California, Morgan Kaufmann: 119-156.

Visser, P. R. S., D. M. Jones, T. J. M. Bench-Capon and M. J. R. Shave (1997). An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability. AAAI-97 Spring Symposium on Ontological Engineering, Palo Alto, California.

Weinstein, P. (1999). Integrating Ontological Metadata: algorithms that predict semantic compatibility. Ph.D. Thesis, Dept. Electrical Engineering and Computer Science, University of Michigan, Ann Arbor.

Weinstein, P. C. and W. P. Birmingham (1998). "Creating Ontological Metadata for Digital Library Content and Services." International Journal on Digital Libraries 2(1): 19-36.

Weinstein, P. C., W. P. Birmingham and E. H. Durfee (1999). "Agent-Based Digital Libraries: Decentralization and Coordination." IEEE Communications Magazine 37(1): 110-115.

Woods, W. A. and J. G. Schmolze (1992). The KL-ONE Family. Semantic Networks in Artificial Intelligence. F. Lehmann, Pergamon Press.