Algorithm, Implementation and Application of the SIM-DL Similarity Server

Krzysztof Janowicz, Carsten Keßler, Mirco Schwarz, Marc Wilkes, Ilija Panov, Martin Espeter, and Boris Bäumer

Institute for Geoinformatics, University of Muenster, Germany
janowicz|carsten.kessler|mirco.schwarz|marc.wilkes|i.panov|m.espeter|boris.baeumer
@uni-muenster.de

Abstract. Semantic similarity measurement gained attention as a methodology for ontology-based information retrieval within GIScience over the last years. Several theories explain how to determine the similarity between entities, concepts or spatial scenes, while concrete implementations and applications are still missing. In addition, most existing similarity theories use their own representation language while the majority of geo-ontologies is annotated using the Web Ontology Language (OWL). This paper presents a context and blocking aware semantic similarity theory for the description logic $\mathcal{ALCHQ}$ as well as its prototypical implementation within the open source SIM-DL similarity server. An application scenario is introduced showing how the Alexandria Digital Library Gazetteer can benefit from similarity in terms of improved search and annotation capabilities. Directions for further work are discussed.

1 Introduction and Motivation

Semantic similarity measurement has become a major research topic within geographic information science during the last years, aiming at improved methods for information retrieval and integration of heterogeneous spatial data sources. The utilization of findings on similarity measurement from psychology [1] promises user interfaces and search results with an improved cognitive plausibility. However, existing similarity theories aiming at the geospatial domain [2–4] mostly lack compatibility with current widespread knowledge representation languages such as the Web Ontology Language (OWL). The similarity theories require the knowledge to be present in specific formats, ignoring the applicability to existing (geo-)ontologies. To overcome this gap between semantic similarity theories on the one hand, and existing ontologies on the other hand, we present the description logic (DL) based SIM-DL theory [5].

The relevance of a similarity framework, however, is not only depending on its applicability to existing knowledge representations, but also on its adaptation to technical prerequisites. The DIG[1] interface has been established as a standard interface for communication between applications such as ontology editors and

[1] Description Logic Implementation Group, http://dig.sourceforge.net/

reasoners. To ensure compatibility with this de-facto standard, we extend the DIG interface by a group of similarity functions. The open source SIM-DL server is introduced as a reference implementation of the SIM-DIG interface.

Existing similarity theories for the geospatial domain have been evaluated in specially designed application scenarios, without implementation in real-world applications. Beyond the SIM-DL theory and server, we also present a gazetteer application to demonstrate the benefits of similarity based applications. Current gazetteers are mostly based on semi-formal feature[2] type thesauri, defining feature types in terms of a hierarchy with a restricted number of relations. We present a novel Web interface for the Alexandria Digital Library gazetteer that makes use of the SIM-DL server and retrieves its information from a feature type ontology to provide an intuitive work flow and enhanced support for novice users.

The remainder of this paper is organized as follows: we first present related work on similarity measurement and description logics, and then introduce an extended version of the SIM-DL theory [5] and framework. The server prototype is discussed, followed by a description of the application scenario and an outlook on future work.

2 Related Work

This section gives a brief overview of related work concerning semantic similarity and introduces the description logic $\mathcal{ALCHQ}$ and its normalization. Only such aspects which are necessary for the understanding of the SIM-DL similarity theory and implementation are described; for further details see [6].

2.1 Semantic Similarity Measurement

The notion of similarity originated in psychology and was established to determine why and how entities are grouped into categories, and why some categories are comparable to each other while others are not [1, 7]. The main challenge with respect to *semantic* similarity measurement is the comparison of meanings as opposed to purely structural comparison. A language has to be specified to express the nature of entities and metrics are needed to determine how (conceptually) close the compared entities are. While entities can be expressed in terms of attributes, the representation of entity types is more complex. Depending on the expressivity of the representation language, types are specified as sets of features, dimensions in a multidimensional space, or formal restrictions specified on sets using various kinds of description logics. While some representation languages have an underlying formal semantics (e.g. model theory), the grounding of several representation languages remains on the level of an informal description. Because similarity is measured between entity types which are representations

[2] It is important to distinguish between *geographic* features as organized in gazetteers, and the features—i.e. properties, parts and functions—used for concept comparison in certain similarity theories (see section 2.1).

of concepts in human minds, similarity depends on what is said (in terms of computational representation) about these types. This again is connected to the chosen representation language, leading to the fact that most similarity measures cannot be compared. Beside the question of representation, context is another major challenge for similarity assessments. In many cases meaningful notions of similarity cannot be determined without defining in respect to what similarity is measured [8, 7, 9].

Similarity has been widely applied within GIScience over the past few years. Based on Tversky's feature model [10], Rodríguez and Egenhofer [2] developed an extended model called Matching Distance Similarity Measure (MDSM) that supports a basic context theory, automatically determined weights, and asymmetry. Raubal and Schwering [3, 4] used conceptual spaces [11] to implement models based on distance measures within geometric space, while Janowicz and Raubal [12] combined model theoretic and geometric aspects to determine similarity based on affordances. Several measures [13, 14, 5] were developed to close the gap between (geo-)ontologies described by various kinds of description logics, and similarity theories that had not been able to handle the expressivity of such languages. Other similarity theories [15, 16] have been developed to determine the similarity between spatial scenes. The ConceptVISTA[3] ontology management and visualization toolkit uses similarity for concept comparison.

2.2 Description Logics and DIG Interface

Description Logics are a family of knowledge representation languages used to model concepts and entities in a knowledge base. Such a knowledge base consists of a TBox containing the terminology, i.e. the vocabulary describing a given domain, and an ABox storing assertions (about named entities). Description logics distinguish two kinds of symbols, logical and non-logical symbols. The former have a pre-defined meaning grounded in set theory, while the latter are domain specific. Logical symbols are either[4] constructors ($\sqcap, \sqcup, \exists, \forall, \leq, \geq$) used to compose complex concepts out of primitive ones or connectives such as equality ($\equiv$) or inclusion ($\sqsubseteq$). Same as for first order logic, the formal semantics of description logics is given by its interpretation. An interpretation $\Im$ is defined as a tuple $\langle \triangle^{\mathcal{I}}, \mathcal{I} \rangle$. $\triangle^{\mathcal{I}}$ denotes a non-empty set called the domain of interpretation, whereas $\mathcal{I}$ describes the interpretation function mapping from non-logical symbols to elements and (binary) relations over $\triangle^{\mathcal{I}}$. The subset $A^{\mathcal{I}}$ of $\triangle^{\mathcal{I}}$ associated with a concept A is also called its extension. Within this paper the term description or specification of a concept denotes the statements (phrased using the DL language; see Table 1) used to represent a concept in our mind, not its extension.

$\mathcal{ALCHQ}$ used as representation language for the SIM-DL similarity measure is an expressive description logic that supports intersection, union, full existential quantification, value restriction, full negation and qualified number restrictions to inductively construct complex concepts out of primitive ones and roles (binary

[3] http://www.geovista.psu.edu/ConceptVISTA

[4] Leaving punctuation and numbers aside.

Table 1. Syntax and semantics of $\mathcal{ALCHQ}$.

Syntax	Semantics	Name
$\top$	$\Delta^{\mathcal{I}}$	Top
$\bot$	$\emptyset$	Bottom
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	Atomic concept
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	Atomic role
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	(Full) negation
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$	Concept equality
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	Concept inclusion
$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$	Role equality
$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	Role inclusion
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	Concept intersection
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	Concept union
$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$	Value restriction
$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	Existential quantification
$\leq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \vert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\vert \leq n\}$	Qualified max. number restriction
$\geq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \vert\{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\vert \geq n\}$	Qualified min. number restriction

predicates). In the following sections the letters A and B are used to represent atomic concepts, R and S for roles and C and D for complex (composed) concepts. X and Y are used for general statements about similarity and alignment that hold for both, concepts and roles. Additional background information about $\mathcal{ALCHQ}$ and related description logics is discussed in [6].

The Web Ontology Language (OWL) comes in different flavors: OWL-Lite is based on the description logic $\mathcal{SHIF}$, while OWL-DL corresponds to $\mathcal{SHOIN}(\mathcal{D})$. The extended new version OWL 1.1 matches the expressivity of $\mathcal{SROIQ}(\mathcal{D})$. For this paper we have chosen the description logic $\mathcal{ALCHQ}$ because it is close enough to OWL-DL, leaving aspects that are not relevant for similarity aside. $\mathcal{ALCHQ}$ even supports qualified number restrictions which are part of OWL 1.1. The main difference between $\mathcal{ALCHQ}$ and the OWL logics is the missing support for several role axioms such as role inclusion in $\mathcal{ALCHQ}$ (a similarity measure for role intersection was discussed in [5]), role transitivity and inverse roles on the one hand as well as nominals and datatype properties on the other hand. While it is hard to find a meaningful notion of similarity for role axioms such as transitivity, the similarity between nominals (and simple datatypes) boils down to instance similarity.

The DIG interface is an API specification for reasoning in DL systems [17]. The DIG 1.1 specification provides an interface for reasoning services based on the $\mathcal{SHOIN}(\mathcal{D})$ language. The specification provides an XML-encoded HTTP interface. Clients communicate with a server via HTTP `POST`, with requests and responses encoded based on the underlying DIG XML Schema[5]. DIG distinguishes between different types of messages and operations. The reasoner's *identification* message is comparable to OGC's `getCapabilities` requests: the server responds which language and services it supports. This is especially important because of the variety of DL languages, i.e. not every DIG server will support all constructs that are part of the specification (the basic constructs are

[5] The DIG XML Schema can be found at: http://dl-web.man.ac.uk/dig/2003/02/

compulsory, however). The *management* operation creates or releases a knowledge base (KB) that is further identified with an unique URI. *Tells* operations insert assertions into the reasoner's KB, while *Asks* operations allow the client to perform reasoning tasks on the KB (see [17] for details).

3 Similarity Framework and Theory

By studying several similarity theories (including feature, geometric and model driven approaches) we found generic patterns which jointly form a framework for measuring similarity between concepts (see also [5, 18]). This section describes the framework and applies it to determine similarity between concepts specified in $\mathcal{ALCHQ}$.

The framework consists of the following five steps. Their concrete implementation depends on the semantic similarity theory on the one hand and the underlying representation language on the other hand.

1. Selection of query (search) and target concepts.
2. Transformation of concepts to canonical form.
3. Definition of an alignment matrix for concept descriptors.
4. Application of constructor specific similarity functions to selected pairs.
5. Determination of normalized overall similarity.

For reasons of readability all equations forming the SIM-DL measure (steps 4 and 5) have been moved to the appendix.

3.1 Query and Target Concepts

Before measuring similarity it needs to be determined which concepts from the examined ontology should be compared. Depending on the application scenario and theory, the query (search) concept C_s can be part of the ontology or phrased using a shared vocabulary [5, 19]. The target concepts $\{C_t\}$ are selected by hand or determined by the context of the query. Such a context specifies the domain of application either by explicitly selecting the compared-to concepts or implicitly by defining a context concept C_c. In the latter case the target concepts are all concepts subsumed by C_c. Same as for the matching distance similarity measure defined by Rodriguez and Egenhofer [2], SIM-DL defines the set of target concepts as $\{C_t | C_t \sqsubseteq C_c\}$. All similarity functions (see section 3.4) are defined with respect to this context.

3.2 Canonical Normal Form

Before similarity can be computed, the compared concepts have to be rephrased to a canonical normal form to reduce potential syntactic influence. The procedure can be further distinguished into a normalization step and the application of rewriting rules. Both steps mostly depend on the underlying representation language and their importance increases with the expressivity of the used language.

In case of geometric representations a canonical normal form can be achieved through mappings between reference spaces if they approximate the same quality space (see [20]).

In case of model driven measures based on description logics, the procedure is more complex. For $\mathcal{ALCHQ}$ we have developed the following disjunctive normal form (DNF): A concept description C is in normal form iff $C = \top$, $C = \bot$ or $C = C_1 \sqcup ... \sqcup C_n$ and each $C_i (i = 1, ...n)$ is of the form:

$$\begin{aligned} C := \bigsqcap_{A \in primitive(C_i)} A \sqcap \bigsqcap_{R \in N_R} \Bigg(& \bigsqcap_{C' \in exists_R(C_i)} (\exists R.C') \sqcap \forall R.forall_R(C_i) \\ \sqcap & \bigsqcap_{C' \in min_R(C_i)} (\geq |min_R(C_i)| R.C') \sqcap \bigsqcap_{C' \in max_R(C_i)} (\leq |max_R(C_i)| R.C') \Bigg) \end{aligned} \tag{1}$$

The set *primitive*(C) represents all (negated) primitives (and $\bot$) at the top-level of C. N_R is the set of available roles, and $exists_R(C)$, $min_R(C_i)$ and $max_R(C_i)$ denote the sets of all C' for which there exists $\exists R.C'$ (respectively min/max restrictions) at the top-level of C. $forall_R(C_i)$ denotes the intersection of concepts $(C1 \sqcap ... \sqcap C_n)$ derived by merging all value restrictions for the role$R$ $(\forall R.C_i)$ on the top level of C. $|min_R(C_i)|$ and $|max_R(C_i)|$ represent the minimum and maximum cardinalities for the role R on the top-level of C. Note that the concepts $forall_R(C_i)$ and C' are again in $\mathcal{ALCHQ}$ normal form.

To ensure that the SIM-DL measure is not influenced by the syntactic form, rewriting rules (see also [21, 22]) have to be applied in order to get a canonical representation of the compared concepts. On the one hand these rewriting rules map between equivalent expressions such as $(\forall R.\bot)$ and $(\leq 0R.\top)$. On the other hand they ensure that only such descriptions are used within concept specifications which (by definition) have an impact on the cardinality of the regarded sets. For instance $(\geq 1R.C) \sqcap (\geq 2R.C)$ is mapped to $(\geq 2R.C)$, while $(... \sqcap \top)$ can be skipped without changing the extension of the specified concept.

3.3 Alignment Matrix and Blocking

While section 3.1 describes how concepts $(C_s, C_{t_1}...C_{t_n})$ are selected, an alignment matrix [5, 23] is necessary to determine which parts of their descriptions are compared. Most theories assume similarity to be a binary relation, hence the alignment matrix creates tuples $sim(X_s, Y_{t_n})$ for all possible combinations of the Cartesian product $C_s \times C_{t_n}$. While C_s and C_{t_n} denote two compared-to concepts, X_s and Y_{t_n} are parts of their descriptions (e.g. concrete number restrictions).

In case of feature based representations such as used for MDSM [2], the alignment matrix is reduced to a 0/1 matching. If two parts of compared concept descriptions have the same label, they count as common features, if not they are distinguishing features. The impact of these features on the overall similarity depends on sub/super relations between the compared concepts (see section 3.5). Note that MDSM distinguishes between three feature types: functions, parts and attributes. Features are only compared if they belong to the same feature type.

Geometric approaches which take relations into account, choose such tuples for later comparison where the target relation is a subtype of the source relation.

For SIM-DL the alignment matrix is defined as follows. If two concepts are compared, an alignment matrix M_1 with all possible combinations of their parts is created. Once similarity for each tuple is calculated (see section 3.4), those tuples with the highest similarity values are chosen for further computation. Note that each X_s respectively Y_{t_n} is only selected once and similarity can only be calculated if both elements of the tuple are based on the same constructor. For instance, a value restriction is never compared to a quantification. For each selected tuple the normalization factor (see section 3.5) is increased by 1.

To handle circular definitions[6] such as $C \equiv ... \sqcap (\forall R.C)$ the matrix (and the similarity functions) need to implement a blocking mechanism as known from tableaux algorithms for subsumption reasoning in DL. For instance, consider the tuple $sim(C, D)$ from the matrix M_1 used to compare a search and target concept (where C is defined as above and $D \equiv ... \sqcap (\forall R.D)$). In order to calculate the similarity between C and D, an alignment matrix M_2 that contains tuples for all possible combinations of the Cartesian product $C \times D$ is created. Since the definition of concept C (and D) is circular, all tuples from M_2 containing $(\forall R.C)$ (and $(\forall R.D)$) will end up in a loop (creating infinite alignment matrices). Instead such tuples are set as *blocked*. All similarity values for tuples in the matrix M_2 are calculated leaving the blocked tuples aside. The result is an approximated similarity between C and D. Using this value, the blocked tuples can now be computed and M_2 (and finally M_1) can be re-calculated without loops. This tuple-wise blocking often appears in case of negation. If only one part of the tuple is blocked (e.g. if $(\forall R.D)$ is replaced by $(\forall R.E)$) the process continues unfolding E and building matrices until no expression to be compared to $(\forall R.C)$ is left, or its filler is either $\top$ or primitive. As similarity can be computed for this tuple, the value is now used one level (matrix) higher and so on until $sim(C, D)$ can be determined. This kind of blocking is called expression-wise here.

3.4 Similarity Functions and Neighborhoods

After choosing the compared-to concepts and aligning their descriptions, similarity is measured for each selected tuple $sim(X_s, Y_{t_n})$. Depending on the constructors used for X_s and Y_t different similarity functions have to be applied.

In case of MDSM, features are distinguished into different types during the alignment process, however, the same similarity measure (a weighted and asymmetric feature ratio function) can be applied to all of them. Geometric approaches allow for several functions either based on different metrics (such as Euclidian or city-block) and, if they support relations, distinguish between similarity (inverse distance) within a conceptual space and network-based similarity measures for relations.

[6] The problem of circularity also affects other similarity measures, but was not taken into account so far.

Because the $\mathcal{ALCHQ}$ knowledge representation language allows for more expressive conceptualizations, SIM-DL has to offer a similarity function for each constructor. The measurement process always starts at the union level (see $\mathcal{ALCHQ}$ canonical normal form; section 3.2) with the sim_u function. Each concept on this level is itself formed by intersection and similarity between such concepts is measured by sim_i[7]. Each concept of this intersection is either a primitive (sim_p), an existential quantification (sim_e), a value restriction (sim_f) or a qualified number restriction (sim_{min}, respectively sim_{max}). In addition to role hierarchies (sim_r) SIM-DL supports temporal and topological neighborhoods (sim_n) to calculate similarity between roles. This allows to determine the similarity between tuples such as ($\exists inside.Lake$, $\exists overlap.Lake$); see [5] for more details. All necessary similarity functions are listed in the appendix.

3.5 Overall Similarity

The overall similarity determines the similarity between compared concepts C_s and C_t based on the similarities for all considered tuples $sim(X_s, Y_{t_n})$. In most examined theories this step was a summation function, normalized to values between 0 and 1.

For MSDM the overall similarity is the weighted sum of the similarity determined between functions, parts and attributes. While the weighting indicates the relative importance of each feature type, at the same time it acts as the normalization factor ($\sum \omega = 1$)[2]. In case of geometric approaches the overall similarity is given by the normalized (via z-transformation) sum of compared dimensions.

For SIM-DL each similarity function discussed in section 3.4 takes care of its normalization using the number of compared tuples. Each similarity function returns a value between 0 and 1 to the function (on a higher level) it was called by.

4 Similarity Server and Interfaces

This section gives a brief overview of the architecture of the DIG-based semantic similarity server. A plug-in for the Protégé Ontology Editor will be described. The SIM-DL server and the plug-in are still under development, but already available as an open-source cross-platform project at Sourceforge.net. The current beta version[8] supports subsumption reasoning and similarity measurement up to $\mathcal{ALCHQ}$, support for more expressive description logics is under development.

[7] Of course primitives, restrictions and quantifications can already appear on union level without violating the measurement process (see appendix).

[8] The current release can be downloaded at http://sim-dl.sourceforge.net/.

4.1 Architecture

The SIM-DL server is based on an embedded Jetty HTTP server[9]. Incoming requests via XML-over-HTTP are processed by a request handler who interprets DIG operations and starts the similarity and reasoning engines. The reasoner implements a tableaux algorithm to determine TBox subsumption based on ABox satisfiability, while the similarity engine is based on the presented SIM-DL framework and theory. Both components implement their own normalization and blocking methodes. Each similarity request involves interaction with the reasoning component to determine target concepts out of the context. The reasoner is also used for some similarity functions such as sim_p. In this paper, we propose the Protégé[10] plugin and gazetteer Web interface (see section 5) as clients; however, every DIG compatible client software can be used.

4.2 SIM-DIG Interface

A short introduction to the DIG interface was given in section 2.2. The interface has to be extended to enable similarity measurement between concepts. First, the *Ask* syntax has to be extended by a similarity query which defines a search concept (C_s) and a context concept (C_c). The search concept is compared to all subclasses of the context concept. Table 2 shows the supported queries as well as our extension.

Table 2. Supported Ask language, *similarity extensions* and query syntax.

Request Category	Tag Syntax
Satisfiability	`<satisfiable>C</satisfiable>`
Concept Hierarchy	`<parents>C</parents>`
	`<children>C</children>`
	`<ancestors>C</ancestors>`
	`<descendants>C</descendants>`
	`<equivalents>C</equivalents>`
Similarity Queries	`<ccsimilarity>CS CC</ccsimilarity>`

The result of a similarity query contains a set of concepts where each concept has a value indicating the similarity to the source concept. Since the existing response operators do not allow for assigning a value to a concept, the response syntax has to be extended, too. Table 3 shows the supported response operators and, additionally, the syntax extension that permits similarity queries.

4.3 Protégé Plug-in

To enable the use of reasoning services there is a need for suitable graphical user interfaces. This holds for standard reasoning tasks, such as subsumption reason-

[9] http://jetty.mortbay.org/

[10] http://protege.stanford.edu/

Table 3. Supported Ask language, *similarity extensions* and response syntax.

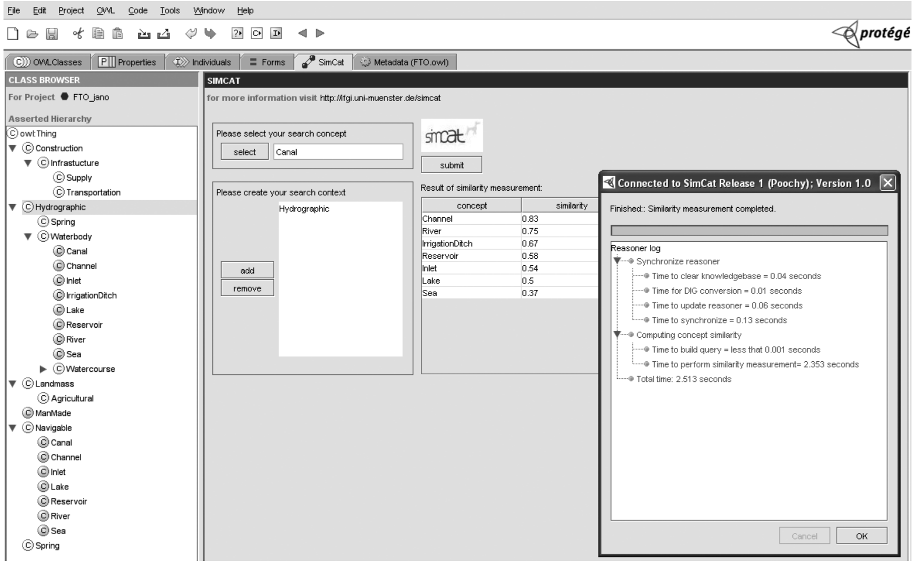
Response Category	Response Syntax	Request Category
Boolean	<true/> <false/>	Satisfiability
Concept Set	<conceptSet> <synonyms>S11...S1N</synonyms> <synonyms>SM1...SMN</synonyms> </conceptSet>	Concept Hierarchy
Similarity Set	<conceptSet> <catom name=S1> <simValue>s1</simValue> </catom> <catom name=SN> <simValue>sN</simValue> </catom> </conceptSet>	*Similarity Query*

ing, as well as for similarity reasoning. Today's standard front-end for DL based reasoning is Protégé, a Java based open source ontology editor and knowledge base framework. It is built upon an extensible architecture that provides the possibility to add further functionality via plug-ins. The Protégé OWL plugin is one of the most popular plug-ins that have been developed for the Protégé framework. It enables users to create, explore and modify OWL ontologies supporting OWL-Lite, OWL-DL and OWL-Full [24]. Additionally, it provides DIG-based access to DL reasoners such as Pellet[11]. The combination of DL theory, reasoning services and Protégé as a graphical frontend was a prerequisite for establishing OWL as the standard for creating semantic web applications. A similar combination will be necessary to initiate the spread of DL based similarity measurement. The Protégé OWL API includes several extension points for implementing OWL specific plug-ins. To provide a graphical frontend for accessing the SIM-DL similarity server we developed the SIM-DL plug-in as a GUI-plugin based on Protégé OWL. The possibilty to view and explore the ontologies that are involved in the similarity measurement process is mandatory. This functionality is already provided by Protégé -OWL and reused for the SIM-DL plug-in. Due to the architecture of the similarity server the SIM-DL plug-in has to support the DIG interface. We reused the DIG implementation provided by Protégé OWL and added the SIM-DL specific DIG elements. Figure 1 shows a screenshot of the current state of the plugin.

5 Gazetteer Application Scenario

The use of similarity measurement in current gazetteers is hampered by a lack of formalism in the corresponding feature type thesauri. In the following, we show how subsumption and similarity based user interfaces can improve the gazetteers' functionality and usability, based on a transformation of feature type thesauri into ontologies.

[11] http://pellet.owldl.com



For Project ● FTO_jano

for more information visit http://ifgi.uni-muenster.de/simcat

Please select your search concept: Canal

Please create your search context: Hydrographic

Result of similarity measurement:

concept	similarity
Channel	0.83
River	0.75
IrrigationDitch	0.67
Reservoir	0.58
Inlet	0.54
Lake	0.5
Sea	0.37

Connected to SimCat Release 1 (Poochy); Version 1.0

Finished:: Similarity measurement completed.

```
Reasoner log
  Synchronize reasoner
    Time to clear knowledgebase = 0.04 seconds
    Time for DIG conversion = 0.01 seconds
    Time to update reasoner = 0.06 seconds
    Time to synchronize = 0.13 seconds
  Computing concept similarity
    Time to build query = less that 0.001 seconds
    Time to perform similarity measurement= 2.353 seconds
  Total time: 2.513 seconds
```

Fig. 1. SIM-DL Protégé Plug-in (beta version).

5.1 From ADL FTT to Feature Type Ontology

Georeferencing is the core functionality of gazetteers as place name directories. The distinction between different place (feature) types is enabled by thesauri, which contain semi-formal descriptions of the feature types and can be queried via *type-lookup* functionality. To fully support subsumption and similarity-based reasoning, a transformation of these thesauri into formal ontologies is required. We use the example of the Alexandria Digital Library (ADL) Feature Type Thesaurus (FTT)[12] in the following to demonstrate the required steps and design decisions. The procedure can be transferred to other thesauri as well.

The ADL FTT contains textual definitions for *preferred* terms in the form of *scope notes* (SN); in addition, *non-preferred* terms are listed as pointers to preferred terms via the *Use* (USE) and *Used for* (UF) relations, e.g. *lakes UF lagoons*. Inheritance between preferred terms is marked by the *broader term* (BT) and *narrower term* (NT) relations, which are not directly comparable to the sub- and supertype relations in ontologies [25], so that transitivity cannot be taken for granted. Moreover, there is only one broader term for every term in the ADL FTT (despite the ANSI-NISO 39.19 standard allowing for multiple inheritance). This single inheritance structure forces every term to be a NT of only one of the six top terms; for example, *cities* are only classified as *administrative areas*, but not as *manmade features*. Beyond NT and BT, the *related term* (RT) relation is used to express diverse kinds of relations between terms, so that the semantics of

[12] http://www.alexandria.ucsb.edu/gazetteer/FeatureTypes/ver070302/index.htm

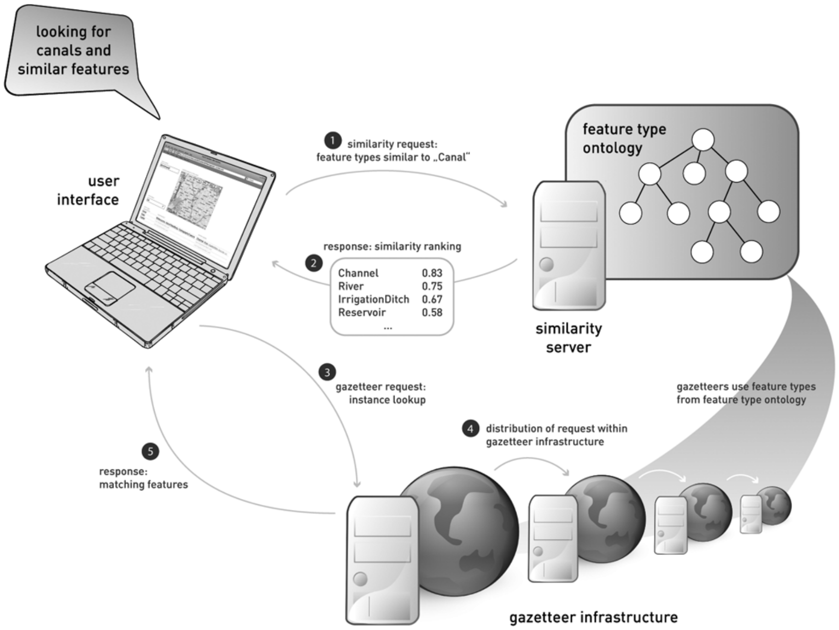
RT remain ambiguous—for example, RT is used to describe the relation of *lakes* to *reservoirs*, i.e. a functional relation, but also to *wetlands*, i.e. a topological relation.

It must be pointed out that the structure of the ADL FTT is not wrong or badly designed, since thesauri are developed for different purposes than ontologies. However, there is a lack of formalism and explicit semantics from an ontological point of view, so that an automatic transformation into a feature type ontology is not possible. To manually transform the thesaurus and preserve the original naming and structure, a syntactic and semantic conversion as described in [25] must be performed. The resulting ontology (see figure 1 for an extract) uses the top level concept *Feature*, subsumed by different classes such as *Manmade*, *Hydrographic* or *Transportation*; note that these classes are not disjoints, i.e. the concept *Canal*, for example, subsumes all these feature classes at the same time. Moreover, feature types (or concepts in ontology terminology) can be related to each other with an arbitrary number of hierarchically ordered properties which have to be extracted manually from the RT relations and the scope notes in the thesaurus. For example, we introduce the property *hasConnection*, with sub-properties *hasOrigin* and *hasDestination*, to specify that a canal connects (*hasDestination*) two hydrographic features. This brief insight into the conversion process shows that the generation of a feature type ontology requires a significant effort; in the following, we argue that such a conversion is worthwhile, as gazetteer Web interfaces can greatly benefit from a feature type ontology.

5.2 Towards a Distributed Gazetteer Infrastructure

The long term vision of current gazetteer research is focussing on the development of a distributed local-responsibility service infrastructure instead of a single world gazetteer. Such an infrastructure can be compared to the Domain Name Service (DNS) which maps hostnames on the internet to their IP addresses. Each gazetteer offers lookup for local places within its spatial and thematic scope. If the gazetteer cannot answer a request, it redirects the query to a higher level gazetteer which decides whether it or another gazetteer can resolve the query. The underlying idea is that gazetteers should contain and maintain data of interest for the community running the service. This ensures that the stored data is both accurate and up-to-date.

A distributed gazetteer infrastructure raises several challenges for both the georeferencing and the type-lookup function. For georeferencing, the main challenge is that several names may point to the same place using different footprints, which includes divergences between the referred-to coordinates, but especially between the type of footprint such as point versus polygon representation (see also [26]). In the case of type-lookup, one must ensure that all involved gazetteers share a common understanding of the feature types used. Gazetteers are developed for different thematic scopes and spatial scales, which may require different conceptualizations of the described features. Consequently, a common feature type specification needs to be generic enough to form a top level for all gazetteers and extensible to allow for local type definitions. Figure 2 illustrates the role of the SIM-DL server within the proposed gazetteer infrastructure.


looking for canals and similar features
user interface
1 similarity request: feature types similar to „Canal"
feature type ontology
response: similarity ranking
2
similarity server
3 gazetteer request: instance lookup
gazetteers use feature types from feature type ontology
4 distribution of request within gazetteer infrastructure
5 response: matching features
gazetteer infrastructure

Channel	0.83
River	0.75
IrrigationDitch	0.67
Reservoir	0.58
...	

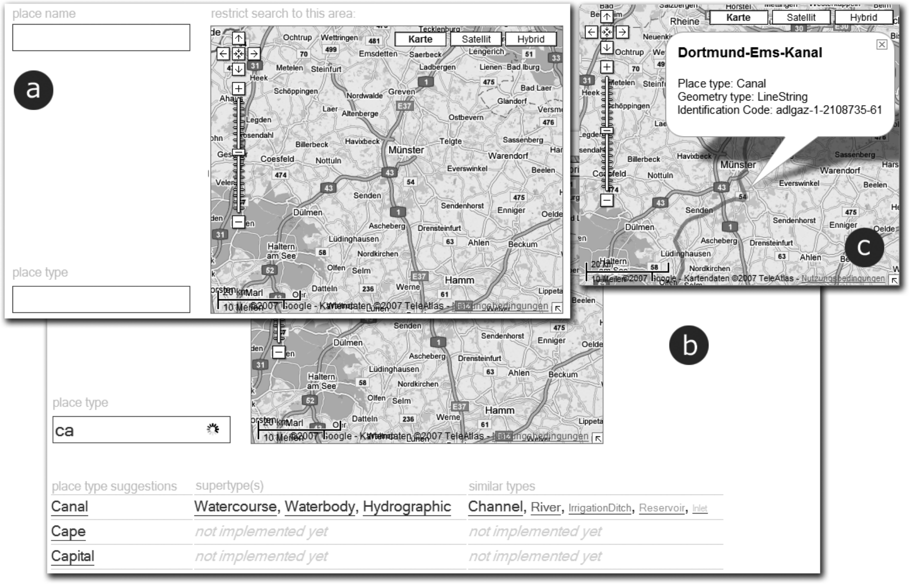
Fig. 2. Similarity-based feature type lookup within the proposed gazetteer infrastructure.

5.3 Similarity-based Gazetteer Web Interface

To efficiently use the ADL gazetteer's Web interface[13], the user needs detailed knowledge of the FTT hierarchy to select the adequate preferred term for what he is looking for. If the user is not aware of the FTT hierarchy, retrieving the desired information is complicated and tedious, as the user must first consult the FTT to find out about the preferred term for his query. To overcome these difficulties, we propose a subsumption and similarity based gazetteer Web interface based on a feature type ontology, as shown in figure 3.

The proposed interface utilizes AJAX technology in a *search-while-you-type* input field: as the user types in the place type he has in mind, results are automatically loaded in the background. The suggested types are based on a syntactic match of the letters already typed in by the user; next to every suggestion, its supertypes and the most similar other types from the ontology are presented,

[13] http://www.alexandria.ucsb.edu/clients/gazetteer/



place type: ca

place type suggestions	supertype(s)	similar types
Canal	Watercourse, Waterbody, Hydrographic	Channel, River, IrrigationDitch, Reservoir, Inlet
Cape	not implemented yet	not implemented yet
Capital	not implemented yet	not implemented yet

Dortmund-Ems-Kanal

Place type: Canal
Geometry type: LineString
Identification Code: adlgaz-1-2108735-61

Fig. 3. Conceptual design for the gazetteer Web interface: search interface with input fields for place name and type, and map for spatial restriction (a); automatic suggestion of place types during user input (b); display of results as map overlays (c).

where the size and color of the type indicate its similarity to the suggested type in the leftmost column. This way, there is no need for the user to know about the underlying feature type hierarchy, as similar types are automatically suggested by the interface. All suggestions are hyperlinked and can be moved to the input field with a single click. Moreover, the interface also allows for spatial restriction by simply zooming the map to the desired extent. The proposed interface thus allows for an intuitive workflow that supports also novice users in the selection of the appropriate feature types for a query. Apart from up-to-date Web technology, this functionality is made possible by the feature type ontology in the background, and by the similarity server accessing it.

6 Conclusions and Further Work

Most existing similarity theories cannot be implemented as parts of semantically enabled information retrieval infrastructures because they do not support the current standards for knowledge representations (such as OWL). In this paper, we introduced an extended version of the SIM-DL theory [5] and its implementation within an open source similarity server. The server is based on an extended DIG interface and can hence interact with existing tools such as reasoners and editors. An application scenario from gazetteer research demonstrated how similarity measurement can be integrated into user interfaces and existing geo-services. In addition one may also think of the SIM-DL server as a web service within a geo-processing chain as realized in spatial data infrastructures. This would enable to query a Web Feature Service for all features of types similar to *Canal*. As (leaving the list view aside) our approach does not include a visualization component, the integration into ConceptVISTA may be an interesting further step. The presented Protégé plug-in allows ontology engineers to integrate similarity into their development process. For instance, similarity can be used to examine whether a constructed ontology reflects the users view (i.e. conceptualizations).

Further work has to focus on similarity measures for even more expressive description logics and especially for taking modal logics into account as discussed by Poole and Smyth [27]. While some parts of the SIM-DL theory have been evaluated by human subject tests (see [18]) or based on previously evaluated work from psychology or computer science, the evaluation of the whole approach is the next step to be done. In addition the proposed gazetteer Web interface has to be tested against existing interfaces to determine to which degree similarity improves interaction. Finally, while this work focuses on comparing the expressions forming the examined concepts, further work should additionally focus on the ABox. Similarity could then be measured in the style of current tableaux algorithms.

7 Appendix

The appendix gives an overview about the involved similarity functions described in section 3.4; for a detailed description see [5]. The sets of tuples selected by the alignment matrix are represented by the letter S followed by an abbreviation for the type of constructor. For instance, SI is the set of concepts on union level of C where each C_i is formed by intersection.

sim_u is the weighted sum of similarities for all tuples (C_i, D_j). The weighting ω $(\sum \omega_{ij} = 1)$ can be either determined by the count of tuples or by analyzing the ontological structure [5]. If the similarity of a particular tuple is 1, $sim_u = 1$.

$$sim_u(C, D) = \sum_{(C_i, D_j) \in SI} \omega_{ij} * sim_i(C_i, D_j) \tag{2}$$

Following the $\mathcal{ALCHQ}$ canonical normal form (see section 3.2), eachC_i (respectively D_j) is an intersection of primitives or concepts formed by restrictions or quantifications. sim_i is the function that determines similarity on this level as normalized sum derived from the similarity functions for the involved constructors. The normalization factor σ is defined as the sum of cardinalities derived from the sets of compared tuples (SP, SE, SF, $SMIN$ and $SMAX$).

$$sim_i(C, D) = \frac{1}{\sigma}\Bigg(\sum_{(A,B)\in SP} sim_p(A, B) + \sum_{(R,S)\in SE} sim_e(exists_R(C), exists_S(D)) + \sum_{(R,S)\in SF} sim_f(forall_R(C), forall_S(D)) + \sum_{(R,S)\in SMIN} sim_m(min_R(C), min_S(D)) + \sum_{(R,S)\in SMAX} sim_m(max_R(C), max_S(D))\Bigg) \tag{3}$$

Primitives have no description that can be compared, hence an information theoretic approach (comparable to the Jaccard coefficient) is used to determine their similarity. Primitives are the more similar, the more complex concepts (within the context) are subsumed by both.

$$sim_p(A, B) = \frac{\mid \{C \mid C \sqsubset A) \sqcap (C \sqsubset B)\} \mid}{\mid \{C \mid C \sqsubset A) \sqcup (C \sqsubset B)\} \mid} \tag{4}$$

sim_e compares concepts formed by existential quantifications. The similarity is the product of role and filler similarity. The second sum (see sim_i) is necessary as there may be more than one existential quantification for the same role.

$$sim_e(exists_R(C), exists_S(D)) = sim_r(R, S) * \sum_{(C'_i, D'_j)\in SE} sim_u(C'_i), D'_j)) \tag{5}$$

sim_f compares concepts formed by value restriction. The similarity is the product of role and filler similarity.

$$sim_f(forall_R(C), forall_S(D)) = sim_r(R, S) * sim_u(forall_R(C), forall_S(D)) \tag{6}$$

The similarity (sim_m) between concepts formed by quantified number restrictions is the product of the similarities determined for the involved roles, fillers and their maximal or minimal occurrence (cardinality). sim_m is used as an abbreviation here, in fact minimum and maximum restrictions are handled separately (i.e. m is replaced by min respectively max). The normalization $m_{RS}(total)$ is the highest maximum (respectively minimum) restriction for R or S within the context. If one cardinality is explicitly set to 0 (while the other is not), $sim_m = 0$.

$$sim_m(mR(C), m_S(D)) = sim_r(R, S) * \left(1 - \frac{\mid m_R(C) - m_S(D) \mid}{m_{RS}(total)}\right) * sim_u(C'_i), D'_j)) \tag{7}$$

The similarity between roles (sim_r) is their normalized distance within the hierarchy. The normalization is depth-dependent to indicate that the distance from node to node decreases with increasing depth of R and S within the hierarchy.

$$sim_r(R, S) = \frac{depth(lub(R, S))}{depth(lub(R, S)) + edge\_distance(R, S)} \tag{8}$$

If roles are not organized within a hierarchy but within a neighborhood, sim_n is used for comparison.

$$sim_n(R, S) = \frac{max\_distance_n - edge\_distance(R, S)}{max\_distance_n} \tag{9}$$

8 Acknowledgments

The comments from three anonymous reviewers as well as from Martin Raubal provided useful suggestions to improve the content and clarity of the paper. This work is funded by the SimCat project granted by the German Research Foundation (DFG Ra1062/2-1).

References

1. Goldstone, R., Son, J.: Similarity. In Holyoak, K., Morrison, R., eds.: Cambridge Handbook of Thinking and Reasoning. Cambridge University Press (2005)
2. Rodríguez, A., Egenhofer, M.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. International Journal of Geographical Information Science **18**(3) (2004) 229–256
3. Raubal, M.: Formalizing conceptual spaces. In Varzi, A., Vieu, L., eds.: Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004). Volume 114 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, NL (2004) 153–164
4. Schwering, A., Raubal, M.: Spatial relations for semantic similarity measurement. In Akoka, J., Liddle, S., Song, I.Y., Bertolotto, M., Comyn-Wattiau, I., vanden Heuvel, W.J., Kolp, M., Trujillo, J., Kop, C., Mayr, H., eds.: Perspectives in Conceptual Modeling: ER 2005 CoMoGIS Workshop, Klagenfurt, Austria. Volume 3770 of Lecture Notes in Computer Science. Springer, Berlin (2005) 259–269
5. Janowicz, K.: Sim-dl: Towards a semantic similarity measurement theory for the description logic $\mathcal{ALCNR}$ in geographic information retrieval. In Meersman, R., Tari, Z., Herrero, P., al., e., eds.: SeBGIS 2006, OTM Workshops 2006. Volume 4278 of Lecture Notes in Computer Science. Springer, Berlin (2006) 1681 – 1692
6. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
7. Medin, D., Goldstone, R., Gentner, D.: Respects for similarity. Psychological Review **100**(2) (1993) 254–278
8. Goodman, N.: Seven strictures on similarity. In Goodman, N., ed.: Problems and projects. Bobbs-Merrill, New York (1972) 437–447
9. Keßler, C.: Similarity measurement in context. In: Sixth International and Interdisciplinary Conference on Modeling and Using Context. Lecture Notes in Artificial Intelligence. Springer (2007) 277–290
10. Tversky, A.: Features of similarity. Psychological Review **84**(4) (1977) 327–352
11. Gaerdenfors, P.: Conceptual Spaces - The Geometry of Thought. Bradford Books, MIT Press, Cambridge, MA (2000)
12. Janowicz, K., Raubal, M.: Affordance-based similarity measurement for entity types. In: 5th Conference on Spatial Information Theory (COSIT 2007). Lecture Notes in Computer Science, Springer (2007, forthcoming)
13. d'Amato, C., Fanizzi, N., Esposito, F.: A dissimilarity measure for $\mathcal{ALC}$ concept descriptions. In: Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France (2006) 1695–1699
14. Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In: Proceedings of the 2005 International Workshop on Description Logics (DL2005). Volume 147 of CEUR Workshop Proceedings. CEUR, Edinburgh, Scotland, UK (2005)

15. Li, B., Fonseca, F.: Tdd - a comprehensive model for qualitative spatial similarity assessment. Spatial Cognition and Computation **6**(1) (2006) 31–62
16. Nedas, K., Egenhofer, M.: Spatial similarity queries with logical operators. In Hadzilacos, T., Manolopoulos, Y., Roddick, J., Theodoridis, Y., eds.: SSTD '03 - Eighth International Symposium on Spatial and Temporal Databases, Santorini, Greece. Volume 2750 of Lecture Notes in Computer Science. (2003) 430–448
17. Bechhofer, S.: The dig description logic interface: Dig/1.1. In: DL2003 Workshop, Rome (2003)
18. Janowicz, K.: Similarity-based retrieval for geospatial semantic web services specified using the web service modeling language (wsml-core). In Scharl, A., Tochtermann, K., eds.: The Geospatial Web - How Geo-Browsers, Social Software and the Web 2.0 are Shaping the Network Society. Lecture Notes in Computer Science. Springer, Berlin (2007)
19. Lutz, M., Klien, E.: Ontology-based retrieval of geographic information. International Journal of Geographical Information Science **20**(3) (March 2006) 233–260
20. Probst, F., Espeter, M.: Spatial dimensionality as classification criterion for qualities. In: International Conference on Formal Ontology in Information Systems, Baltimore, Maryland, IOS Press (2006)
21. Brandt, S., Küsters, R., Turhan, A.Y.: Approximation and difference in description logics. In Fensel, D., Giunchiglia, F., McGuiness, D., Williams, M.A., eds.: International Conference on Principles of Knowledge Representation and Reasoning (KR2002), San Francisco, CA, Morgan Kaufman (2002) 203–214
22. Molitor, R.: Structural Subsumption for $\mathcal{ALN}$. LTCS-Report LTCS-98-03, LuFG Theoretical Computer Science, RWTH Aachen, Germany (1998)
23. Markman, A.: Structural alignment, similarity, and the internal structure of category representations. In: Similarity and Categorization. Oxford University Press., Oxford, UK (2001) 109–130
24. Knublauch, H., Fergerson, R., Noy, N., Musen, M.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. Third International Semantic Web Conference (2004) 229–243
25. van Assem, M., Menken, M., Schreiber, G., Wielemaker, J., Wielinga, B.: A method for converting thesauri to rdf/owl. In: 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan (2004)
26. Janée, G.: Rethinking gazetteers and interoperability. In: International Workshop on Digital Gazetteer Research & Practice (Santa Barbara, California; December 7-9, 2006). (2006)
27. Poole, D., Smyth, C.: Type uncertainty in ontologically-grounded qualitative probabilistic matching. In Godo, L., ed.: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 8th European Conference, (ECSQARU 2005). Volume 3571 of Lecture Notes in Computer Science., Barcelona, Spain, Springer (2005) 763–774