

# Knowledge Management: Concepts, Methodologies, Tools, and Applications

Murray E. Jennex  
*San Diego State University, USA*



**INFORMATION SCIENCE REFERENCE**

Hershey • New York

Assistant Executive Editor: Meg Stocking  
Acquisitions Editor: Kristin Klinger  
Development Editor: Kristin Roth  
Senior Managing Editor: Jennifer Neidig  
Managing Editor: Sara Reed  
Typesetter: Amanda Appicello  
Cover Design: Lisa Tosheff  
Printed at: Yurchak Printing Inc.

Published in the United States of America by  
Information Science Reference (an imprint of IGI Global)  
701 E. Chocolate Avenue, Suite 200  
Hershey PA 17033  
Tel: 717-533-8845  
Fax: 717-533-8661  
E-mail: [cust@igi-pub.com](mailto:cust@igi-pub.com)  
Web site: <http://www.igi-pub.com/reference>

and in the United Kingdom by  
Information Science Reference (an imprint of IGI Global)  
3 Henrietta Street  
Covent Garden  
London WC2E 8LU  
Tel: 44 20 7240 0856  
Fax: 44 20 7379 0609  
Web site: <http://www.eurospanonline.com>

Copyright © 2008 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher.

Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

# Chapter 3.4

## Description Logic–Based Resource Retrieval

**Simona Colucci**

*Politecnico di Bari, Italy*

**Tommaso Di Noia**

*Politecnico di Bari, Italy*

**Eugenio Di Sciascio**

*Politecnico di Bari, Italy*

**Francesco M. Donini**

*Università della Tuscia, Italy*

**Marina Mongiello**

*Politecnico di Bari, Italy*

### INTRODUCTION

Resource retrieval addresses the problem of finding best matches to a request among available resources, with both the request and the resources described with respect to a shared interpretation of the knowledge domain the resource belongs to. The problem of resource matching and retrieval arises in several scenarios, among them, personnel recruitment and job assignment, dating agencies, but also generic electronic marketplaces, Web services discovery and composition, resource matching in the Grid. All these scenarios share

a common purpose: given a request, find among available descriptions those best fulfilling it, or at “worse,” when nothing better exists, those that fulfill at least some of the requirements.

Exact, or full, matches are usually rare and the true matchmaking process is aimed at providing one or more “promising” matches to be explored. Non-exact matches should take into account both missing information—details that could be positively assessed in a second phase—and conflicting information—details that could leverage negotiation if the proposed match is worth enough pursuing.

Because of its intangibility, it is now a widely shared opinion that knowledge has to be modeled to make unambiguous the interpretation of any information domain. This disambiguation process is usually obtained through an ontology, that is, a specification of a representational vocabulary for a shared domain of discourse—definitions of classes, relations, functions, and other objects (Gruber, 1993).

Once a knowledge domain has been modeled, and several different resources have been described using such a model, issues that need to be faced for efficient knowledge management are: What, if any, kind of retrieval is possible on these resources? How could we benefit both of the model and formalisms used to build the model, in order to perform a “smart” search of described resources matching a request? The above questions focus on important aspects of knowledge-based retrieval:

- formalisms used to model a knowledge domain
- retrieval services that fully use the expressiveness of the formalism to infer new knowledge from the model in order to perform a knowledge-based search

Knowledge domain is modeled with a formalism, whose expressiveness is used in the retrieval process to infer not elicited information from the model. In such a context, choosing this formalism strongly affects the complexity, as well as success probability, of the retrieval process.

In recent years description logics (DLs) have been investigated by both the academic and industrial world as a formalism for knowledge representation. Modeling an information domain through the formalism of a DL allows one to employ reasoning services provided by DLs to perform a knowledge-based search. Knowledge domains are formalized in ontologies, which resource descriptions refer to. The use of ontologies allows elicited descriptions to be stored so

that information can be inferred from them to retrieve a resource.

The remainder of this article is structured as follows: Background work is revised, including DL basics with associated reasoning services and previous approaches to resource retrieval, including non-logic- and logic-based alternatives. Then, we introduce semantic-based resource retrieval, first highlighting new non-standard inference services and then showing how they can be used for “smart” resource retrieval. Finally, we propose some future trends and draw a conclusion.

## **BACKGROUND**

### **Description Logics Basics**

Description, or terminological, logics (Baader, Calvanese, Mc Guinness, Nardi, & Patel-Schneider, 2002; Donini, Lenzerini, Nardi, & Schaerf, 1996) are a family of logic formalisms for knowledge representation. All DLs are endowed of a syntax and a model-theoretic semantics. The basic syntax elements of DLs are: concept names, role names, individuals. Intuitively, concepts stand for sets of objects, and roles link objects belonging to different concepts. Individuals are special named elements of the sets of objects concepts represent.

We give a more formal definition of the outlined basic elements by introducing the concept of semantic interpretation.

**Definition 1:** A semantic interpretation is a pair  $I=(\Delta, \cdot I)$  made up of a domain  $\Delta$  and an interpretation function  $\cdot I$ , which maps every concept to a subset of  $\Delta$ , every role to a subset of  $\Delta \times \Delta$ , and every individual to an element of  $\Delta$ .

Usually, a so-called Unique Name Assumption (UNA) is made which ensures different individu-

als to be mapped to different elements of  $\Delta$ , i.e.,  $aI \neq bI$  for individuals  $a \neq b$ .

Every DL allows one to combine basic elements using constructors to form concept and role expressions. Each DL has its distinguished set of constructors, though all of them provide the conjunction of concepts, usually denoted as  $\sqcap$ . Among the distinguishing concept expressions constructors we enumerate disjunction  $\sqcup$  of concepts and complement  $\neg$  to close concept expressions under Boolean operations.

Role expressions can be obtained by combining roles with concepts using existential role quantification and universal role quantification. Other constructs may involve counting, as number restrictions.

Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations (Calvanese, De Giacomo, & Lenzerini, 1998). Nevertheless, it is a well-known result that usually leads to an explosion in computational complexity of inference services (Brachman & Levesque, 1984). Hence, a trade-off is needed between expressivity and expected performance of reasoning services.

Once expressions have been built, they are given semantics by defining the interpretation function over each construct. Concept conjunction is interpreted as set intersection, and the other Boolean connectives also have the usual set-theoretic interpretation. The interpretation of constructs involving quantification on roles needs to make domain elements explicit.

Concept expressions can be used in inclusion assertions, and definitions, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain. Definitions are useful to give a meaningful name to particular combinations. Sets of such inclusions are called TBox (terminological box). A TBox, which basically amounts to an ontology, represents a formal, shared, and objective intensional knowledge on a domain. Individuals can be asserted to belong to a concept using membership assertions in an ABox.

An ABox is the extensional knowledge of the domain that can be described based on the TBox. The semantics of inclusions and definitions is based on set containment: An interpretation  $I$  satisfies an inclusion  $C \sqsubseteq D$  if  $CI \sqsubseteq DI$ , and it satisfies a definition  $C = D$  when  $CI = DI$ . A model of a TBox  $T$  is an interpretation satisfying all inclusions and definitions of  $T$ . DL-based systems are equipped with reasoning services: logical problems whose solution can make explicit knowledge that was implicit in the assertions.

DL-based systems usually provide at least two basic reasoning services for  $T$ :

- Concept Satisfiability: Given a TBox  $T$  and a concept  $C$ , does there exist at least one model of  $T$  assigning a non-empty extension to  $C$ ?
- Subsumption: Given a TBox  $T$  and two concepts  $C$  and  $D$ , is  $C$  more general than  $D$  in any model of  $T$ ?

The previous services can be seen, from a knowledge management perspective, in a more informal way:

- Concept Satisfiability: Given an ontology ( $T$ ) modeling the domain we are investigating on and a description ( $C$ ) of a resource referring to the ontology: Is the information modeled in the description consistent with the one in the ontology?
- Subsumption: Given an ontology ( $T$ ) modeling the domain we are investigating on and two resources described by expressions ( $C$ ,  $D$ ) referring to the information modeled in the ontology: Is the information about a resource more general than the one related to the other one?

Both Subsumption and Satisfiability are adequate in all those knowledge management contexts where a yes/no answer is enough. For example, given a resource and a request represented

respectively by a concept  $O$  and a concept  $R$ , using Concept Satisfiability we are able to determine whether they are compatible, that is,  $O$  models information that is not in conflict with the one modeled by  $R$ . This task can be performed checking the satisfiability of the concept  $O \sqcap R$ .

On the other hand, Subsumption can be used to verify, for example, if a resource described by  $O$  satisfies a request  $R$ , namely, if the relation  $O \sqsubseteq R$  holds, then  $O$  is more specific than  $R$  and it contains at least all the requested features.

For ABoxes, other standard inference services have been defined. Among the various devised, we point out:

- Instance checking: An assertion  $\alpha$  is entailed by an Abox  $A$ , if every interpretation that satisfies  $A$  also satisfies  $\alpha$ .
- Retrieval problem: Given an Abox  $A$  and a concept  $C$ , find all individuals  $\alpha$  such that  $A$  entails  $C(\alpha)$ .
- Realization problem: Given an individual  $\alpha$  and a set of concepts, find the most specific concept  $C$  from the set such that  $A$  entails  $C(\alpha)$ .

Together with standard inference problems, non-standard ones have been proposed and investigated. The least common subsumer (lcs), most specific concept (msc), unification, matching and concept rewriting have been thoroughly presented by Baader et al. (2002). The application field for lcs and msc is the construction of DL knowledge bases using a bottom-up approach instead of the usual top-down one (Baader & Turhan, 2002). The unification and matching services are useful for large knowledge bases maintenance, allowing knowledge engineers to catch equivalence or subsumption relationships among concept expressions (Baader & Turhan, 2002). With concept rewriting the readability of large concept descriptions can be increased, by using concepts defined in an ontology.

Although the general approach proposed in this article does not depend on a particular DL, it has been fully devised for a particular DL, namely the ALN (Attributive Language with Number Restrictions). Constructs allowed in an ALN DL are:

- $\top$  Universal Concept: All the objects in the domain
- $\perp$  Bottom Concept: The empty set
- $A$  Atomic Concepts: All the objects belonging to the set represented by  $A$
- $\neg A$  Atomic negation: All the objects not belonging to the set represented by  $A$
- $C \sqcap D$  Intersection: The objects belonging both to  $C$  and  $D$
- $\forall R.C$  Universal restriction: All the objects participating to the  $R$  relation whose range are all the objects belonging to  $C$
- $\exists R$  Unqualified existential restriction: There exists at least one object participating in the relation  $R$ . Notice that  $\exists R \equiv (\geq 1 R)$
- $(\geq n R) \mid (\leq n R) \mid (= n R)$  Unqualified number restrictions: Respectively, the minimum, the maximum, and the exact number of objects participating in the relation  $R$ . We write  $(= n R)$  for  $(\geq n R) \sqcap (\leq n R)$

We adopt a simple-TBox, that is, in all the axioms (for both inclusion and definition) the left side is represented by a concept name, and there is only one axiom for each atomic concept.

Ontologies using this logic can be easily modeled using languages for the Semantic Web. These languages have been conceived to allow for representation of machine-understandable, unambiguous, description of Web content through the creation of domain ontologies, and aim at increasing openness and interoperability in the Web environment. The strong relation between DLs and the introduced languages for the Semantic Web also is evident in the definition of the OWL language. In fact, there are three different sub-languages for OWL:

## **Description Logic-Based Resource Retrieval**

- OWL-Lite: It allows class hierarchy and simple constraints on relation between classes.
- OWL-DL: Based on description logics theoretical studies, it allows a great expressiveness keeping computational soundness and completeness.
- OWL-Full: Using such a language, there is a huge syntactic flexibility and expressiveness. This freedom is paid in terms of no computational guarantee.

The ALN DL is basically a subset of OWL-DL.

## **Approaches to Resource Retrieval**

We start with a description of various approaches to resource retrieval, highlighting limitations of non-logical approaches, then discussing the general knowledge representation principles that a logical approach may yield.

Modeling a resource retrieval framework using standard relational database techniques would require to completely align the attributes of the available and requested resources descriptions, in order to evaluate a match. On the other hand, if requests and offers are simple names or terms, the only possible match would be identity, resulting in an all-or-nothing approach to the retrieval process. Vague query answering, proposed by Motro (1988), was an initial effort to overcome limitations of relational databases, with the aid of weights attributed to several search variables.

Vector-based techniques taken by classical information retrieval can be used as well, thus, reverting the search for a matching request to similarity between weighted vectors of stemmed terms, as proposed in the COINS matchmaker (Kuokka & Harada, 1996) or in LARKS (Sycara, Klusch, & Lu, 2002). Such a formalization for resource descriptions makes retrieval only probabilistic because descriptions lack a document structure, causing strange situations to ensue. Let us consider

for example the following sentences, describing respectively competences required for a job in a company and competences provided by a worker: “engineer, with experience of two years as project manager, not full time employed, available to transfers” and “experienced project manager, full time employed as engineer for two years, not available to transfers.”

The former is a simple example in which two descriptions in obvious conflict may be considered an exact match because of the formalism chosen to represent them. A further approach structures resource descriptions as a set of words. This formalization allows one to evaluate not only identity between sets but also some interesting set-based relations between descriptions, such as inclusion, partial overlap, and cardinality of set difference. Modeling resource descriptions as set of words is anyway too much sensible to the choice of words employed to be successfully used: the fixed terminology misses meaning that relate words. Such a problem can be overcome by giving terms a logical and shared meaning through an ontology (Fensel, van Harmelen, Horrocks, McGuinness, & Patel-Schneider, 2001). Nevertheless, set-based approaches have some properties we believe are fundamental in a resource matching and retrieval process. If we are searching for a resource described through a set of words, we also are interested in sets including the one we search, because they completely fulfill the resource to retrieve. Moreover, even if there are characteristics of the retrieved resource not elicited in the description of the searched resource, an exact match is still possible because absent information have not to be considered negative. The two statements may be summarized in the following property:

- Property 1 [Open-world descriptions]: The absence of a characteristic in the description of a resource to be retrieved should not be interpreted as a constraint of absence. Instead, it should be considered as a characteristic

that could be either refined later or left open if it is irrelevant for the user searching for the resource.

The set-based match evaluation is non-symmetric: If we search for a resource A, whose describing set of words is included in a set characterizing resource B, we may consider B a resource perfectly satisfying the request for A. On the other hand, if we use the description of B for the search, A also may satisfy the request only partially, as some of the terms describing B may be not included in the A set. We formalize this behaviour as follows:

- Property 2 [Non-symmetric evaluation]: Given two resource descriptions A and B, a resource retrieval system may give different rankings depending on whether it is searching A using B description as query, or B using A as query.

From now on, we assume that resource descriptions, requested and offered, are expressed in a DL. This approach includes the sets-of-keywords one, since a set of keywords also can be considered as a conjunction of concept names. We also assume that a common ontology is established, as a TBox in DL.

With reference to recent related work on logic-based matching and retrieval of resources, approaches are concentrated on electronic marketplaces, where resources are supplies and demands, and Web services discovery, where resources are e-services to be discovered and composed. Finin, Fritzson, McKay, and McEntire (1994) and Kuokka and Harada (1996) introduced matchmaking based on KQML, as an approach whereby potential producers/consumers could provide descriptions of their products/needs to be later unified by a matchmaker engine to identify potential matches. A rule-based approach using the knowledge interchange format (KIF) (Gen-

esereth, 1991), the SHADE prototype (Kuokka & Harada, 1996), or a free-text comparison (the COINS prototype) (Kuokka & Harada, 1996) were used. Approaches similar to the previous ones were deployed in SIMS (Arens, Knoblock, & Shen, 1996), which used KQML and LOOM as description language and InfoSleuth (Jacobs & Shea, 1995), which adopted KIF and the deductive database language LDL++. LOOM also is at the basis of the matching algorithm addressed by Gil and Ramachandran (2001).

Sycara et al. (2002) and Paolucci, Kawamura, Payne, and Sycara (2002) proposed the LARKS language, specifically designed for agent advertisement. The matching process is a mixture of classical IR analysis of text and semantic match via Q-subsumption. Nevertheless, a basic service of a semantic approach, such as inconsistency check, seems unavailable with this type of match.

First approaches based on standard inference services offered by DL reasoners were proposed by Di Sciascio, Donini, Mongiello, and Piscitelli (2001), Gonzales-Castillo, Trastour, and Bartolini (2001), and Trastour, Bartolini, and Priest (2002). Di Noia, Di Sciascio, Donini, and Mongiello (2003b, 2003c) described and motivated properties that a matchmaker should have in a DL-based framework, and algorithms to classify and rank matches into classes were presented. Matchmaking of Web services, providing a ranking of matches based on this DL-based approach was presented by Colucci, Di Noia, Di Sciascio, Donini, and Mongiello (2003b). An extension to the approach by Paolucci et al. (2002) was proposed by Li and Horrocks (2003) where two new levels for service profiles matching were introduced. Notice that the intersection satisfiable level was introduced, whose definition is close to the one of potential matching proposed by Di Noia et al. (2003b), but no measure of similarity among intersection satisfiable concepts was given.

Benatallah, Hacid, Rey, and Toumani (2003) proposed an approach to Web services discovery

based on the difference operator in DLs (Teege, 1994), followed by a set covering operation optimized using hypergraph techniques.

## **SEMANTIC-BASED RESOURCE RETRIEVAL**

### **The Need for New Non-Standard Reasoning Services**

In all those approaches where no explanation on the obtained results is requested or no belief revision is admitted, Subsumption and Consistency Checking are enough. The following are typical examples of the behaviour the reasoning services would have for resource retrieval:

- Subsumption: “Yes, your request is completely satisfied by resourceX”

$$\text{resourceX} \sqsubseteq \text{request}$$

- Consistency Checking: “No, your request is not compatible with resourceX”

$$\text{resourceX} \sqcap \text{request} \equiv \perp$$

Unfortunately, in a semantic-based resource retrieval system a simple yes/no answer cannot be enough; the requester is often interested in explanations especially when the system returns a negative answer. Some of the questions are:

- “What should I give up in my request in order to regain satisfiability with the offered resource?”
- “How should I contract my request?”
- “What should I revise in my request in order to be completely satisfied?”
- “What should I abduce in the available resource?”

Colucci et al. (2003a) and Di Noia et al. (2003a) introduced and defined Concept Abduction—for no-Subsumption explanation—and Concept Contraction—both for un-Consistency Checking explanation and for belief revision suggestion—as new non-standard inference services for DLs. In this subsection, we briefly recall their definitions, explaining their rationale and the need for them in resource retrieval.

### **Concept Contraction**

Starting with the concepts  $O$  and  $R$ , if the conjunction  $O \sqcap R$  is unsatisfiable in the TBox  $T$  representing the ontology—that is, they are not compatible with each other—we may want to retract requirements in  $R$ ,  $G$  (for Give up), to obtain a concept  $K$  (for Keep) such that  $K \sqcap O$  is satisfiable in  $T$ . This scenario can be formally depicted as:

**Definition 2:** Let  $L$  be a DL,  $O, R$ , be two concepts in  $L$ , and  $T$  be a set of axioms in  $L$ , where both  $O$  and  $R$  are satisfiable in  $T$ . A Concept Contraction Problem (CCP), identified by  $\langle L, R, O, T \rangle$ , is finding a pair of concepts  $\langle G, K \rangle \in L \times L$  such that  $T \models R \equiv G \sqcap K$ , and  $K \sqcap O$  is satisfiable in  $T$ . We call  $K$  a contraction of  $R$  according to  $O$  and  $T$ .

We use  $Q$  as a symbol for a CCP, and we denote with  $\text{SOLCCP}(Q)$  the set of all solutions to a CCP  $Q$ . We note that there is always the trivial solution  $\langle G, K \rangle = \langle R, T \rangle$  to a CCP. This solution corresponds to the most drastic contraction, that gives up everything of  $R$ . In our resource retrieval framework, it models the (infrequent) situation in which, in front of some very appealing resource  $O$ , incompatible with the requested one, a user just gives up completely his or her specifications  $R$  in order to meet  $O$ . On the other hand, when  $O \sqcap R$  is satisfiable in  $T$ , the “best” possible solution is  $\langle T, R \rangle$ , that is, give up nothing, if possible. Hence, a

Concept Contraction problem is an extension of a satisfiable one. Since usually one wants to give up as few things as possible, some minimality in the contraction must be defined (Gärdenfors, 1988). In most cases, a pure logic-based approach could not be sufficient to decide between which beliefs to give up and which to keep. There is the need of modeling and defining some extra-logical information to be taken into account. One approach is to give up minimal information (Colucci et al., 2003a). Another one considers some information more important than other and the information that should be retracted is the least important one, that is negotiable and strict constraints are introduced (Di Noia, Di Sciascio, & Donini, 2004).

### Concept Abduction

If the offered resource  $O$  and the requested one  $R$  are compatible, the partial specifications problem still holds, that is, it could be the case that  $O$ —though compatible—does not imply  $R$ . Then, it is necessary to assess what should be hypothesized ( $H$ ) in  $O$  in order to completely satisfy  $R$ .

**Definition 3:** Let  $L$  be a DL,  $O, R$ , be two concepts in  $L$ , and  $T$  be a set of axioms in  $L$ , where both  $O$  and  $R$  are satisfiable in  $T$ . A Concept Abduction Problem (CAP), identified by  $\langle L, R, O, T \rangle$ , is finding a concept  $H \in L$  such that  $T \models O \sqcap H \sqsubseteq R$ , and moreover  $O \sqcap H$  is satisfiable in  $T$ . We call  $H$  a hypothesis about  $O$  according to  $R$  and  $T$ .

We use  $P$  as a symbol for a CAP, and  $SOL(P)$  to denote the set of all solutions to a CAP  $P$ . Observe that in the definition, we limit to satisfiable  $O$  and  $R$ , since  $R$  unsatisfiable implies that the CAP has no solution at all, while  $O$  unsatisfiable leads to counterintuitive results ( $\neg R$  would be a solution in that case). If  $O \sqsubseteq R$ , then we have  $H = T$  as a solution to the related CAP. Hence, Concept

Abduction extends subsumption. On the other hand, if  $O \equiv T$  then  $H \sqsubseteq R$ .

Notice that both Concept Abduction and Concept Contraction can be used for, respectively, subsumption and satisfiability explanation. For Concept Contraction, having two concepts not compatible with each other, in the solution  $\langle G, K \rangle$  to the CCP  $\langle L, R, O, T \rangle$ ,  $G$  represents “why”  $O$  and  $R$  are not compatible. For Concept Abduction, having  $R$  and  $O$  such that  $T \not\models O \sqsubseteq R$ , the solution  $H$  to the CAP  $\langle L, R, O, T \rangle$  represents “why” the subsumption relation does not hold.  $H$  amounts to what is specified in  $R$  and not in  $O$ .

Expected performances of inference services are obviously of paramount importance to evaluate the feasibility of an approach. We hence provide some insight into complexity issues of the services. We note that since Concept Abduction extends Concept Subsumption w.r.t. a TBox, complexity lower bounds of the latter problem carry over to decision problems related to a CAP.

- Proposition: Let  $P \langle L, R, O, T \rangle$ , be a CAP. If Concept Subsumption w.r.t. a TBox in  $L$  is a problem  $C$ -hard for a complexity class  $C$ , then deciding whether a concept belongs to  $SOL(P)$  is  $C$ -hard.

As Concept Abduction extends Subsumption, Concept Contraction extends satisfiability—in particular, satisfiability of a conjunction  $K \sqcap R$ .

- Proposition: Let  $L$  be a DL containing  $\mathcal{AL}$ , and let Concept Satisfiability w.r.t. a TBox in  $L$  be a problem  $C$ -hard for a complexity class  $C$ . Then, deciding whether a pair of concepts is a solution of a CCP  $Q = \langle L, R, O, T \rangle$ , is  $C$ -hard.

Both for Concept Abduction and Concept Contraction, for every single CAP—conversely CCP—there is not only one solution. Different kinds of solution can be classified with respect

to different minimality criteria. Colucci et al. (2003a), Di Noia et al. (2003a), and Colucci et al. (2004) present the definition of some minimality criteria and corresponding complexity results.

### Approximate Resource Retrieval via Concept Abduction and Concept Contraction

We now show how the previously introduced services can help in an approximate, semantic-based search of resources, fully exploiting their structured description. Let us suppose to have request  $R$  and an appealing resource  $O$  such that  $T \models R \sqcap O \equiv \perp$ , that is, they are incompatible. In order to gain compatibility, a Concept Contraction is needed so that giving up  $G$  in  $R$ , the remaining  $K$  can be satisfied by  $O$ . Now, if  $T \not\models O \sqsubseteq K$ , the solution  $HK$  to the CAP  $\langle L, R, O, T \rangle$  represents what is in  $K$  and is not specified in  $O$ .

As the  $O$  obtained is an approximated match of  $R$ , then a measure is needed on how good the approximation is. Given more than one appealing resource, which one is the best approximation? How can it be assigned a numerical score to the approximation, based on  $K$ ,  $H$  and  $G$ , in order to rank the resources?

In table 1, we present a simple algorithm to provide answers to the raised issues.

Notice that  $H = \text{abduce}(O, R, T)$  [rows 3,6] determines  $H$  is a solution for the CAP  $\langle L, R, O, T \rangle$ ;  $\langle G, K \rangle = \text{contract}(O, R, T)$  [row 2] determines  $\langle G, K \rangle$  is a solution for the CCP  $\langle L, R, O, T \rangle$ .

The algorithm retrieve returns values useful in a retrieval system where explanation of the results is needed and/or a belief revision process is admitted.

[rows 1-4] Having a requested resource  $R$  and an offered one  $O$ , if their descriptions conjunction is not satisfiable w.r.t. the ontology they refer to (i.e., they are not compatible with each other for some concepts in their descriptions), first a contraction on  $R$  is performed in order to regain compatibility [row 2], and then what is to be hypothesized in  $O$  in order to completely satisfy  $R$  (its contraction) is computed [row 3]. The returned values represent:

- $G$ : What is to be given up in the request in order to continue the process, or, in other words, why  $R$  is not compatible with  $O$ .
- $HK$ : After the contraction of  $R$ , the request is represented by  $K$ , that is, the portion of  $R$  which is compatible with  $O$ .  $HK$  represents what is to be hypothesized in  $O$  in order to completely satisfy  $K$ , or, in other words, why  $O$  does not completely satisfy  $K$ .

Table 1.

<p><b>algorithm</b> <i>Retrieve</i>(<math>O, R, T, L</math>)  <b>input</b> <math>O, R \sqcap K \sqsupseteq G</math> concepts in <math>L</math> such that both <math>T \sqcap O</math> and <math>T \sqcap R</math>.  <b>output</b> <math>\langle G, H \rangle</math>, respectively, the part in <math>R</math> that should be given up and the part in <math>O</math> that should be hypothesized in order to find an exact match between <math>O</math> and <math>R</math>.  <b>begin algorithm</b>  1:     <b>if</b> <math>T \sqcap R \sqsupseteq O \sqcap</math> <b>then</b>  2:         <math>\langle G, K \rangle = \text{contract}(O, R, T)</math>;  3:         <math>H_K = \text{abduce}(O, K, T)</math>;  4:         <b>return</b> <math>\langle G, H_K \rangle</math>;  5:     <b>else</b>  6:         <math>H = \text{abduce}(O, R, T)</math>;  7:         <b>return</b> <math>\langle \emptyset, H \rangle</math>;  <b>end algorithm.</b></p>
---

[rows 5-7] If the conjunction of R's and O's description is satisfiable w.r.t. the ontology they refer to, then no contraction is needed and only an abductive process is carried out.

The algorithm does not depend on the particular DL adopted. Based on the minimality criteria proposed by Di Noia et al. (2003a), the length |H| of the solution to a CAP for an ALN DL can be computed as proposed by Di Noia et al. (2003c). Hence, a relevance ranking score can be computed by an utility function defined as  $U(|G|,|K|,|HK|)$ .

The rationale of the retrieve algorithm is hereafter presented with the aid of a simple example. Let  $\mathcal{T}, \mathcal{R}, \mathcal{O}$  be a set of axioms, a searched resource description, and an available resource description, respectively, defined as follows:

$$\begin{aligned} \mathcal{T} = \{ & \\ PC & \sqsubseteq \text{Computer} \sqcap \exists \text{hasOS} \\ \text{HomePC} & \sqsubseteq PC \sqcap \forall \text{hasOS.MS} \sqcap \exists \text{pointer} \\ \text{HighLevel} & \sqsubseteq \exists \text{cost} \sqcap \forall \text{cost.Expensive} \\ \text{Expensive} & \sqsubseteq \neg \text{Cheap} \\ \text{MS} & \sqsubseteq \neg \text{Unix} \\ & \} \\ \mathcal{R} & = \text{HomePC} \sqcap \exists \text{monitor} \sqcap \forall \text{pointer.} \forall \text{cost.} \\ & \quad \text{Cheap} \\ \mathcal{O} & = PC \sqcap \exists \text{pointer} \sqcap \forall \text{pointer.} (\text{Mouse} \sqcap \text{High-} \\ & \quad \text{Level}) \sqcap \forall \text{hasOS.Unix} \end{aligned}$$

First, we observe that  $\mathcal{T} \models \mathcal{R} \sqcap \mathcal{O} \equiv \perp$ , due to the specifications on both Operating System and cost of the pointer. Hence, the algorithm performs a Concept Contraction solving the CCP  $\langle \mathcal{L}, \mathcal{R}, \mathcal{O}, \mathcal{T} \rangle$ . A solution for the previous CCP is:

$$\langle G, K \rangle = \langle \text{HomePC} \sqcap \forall \text{pointer.} \forall \text{cost. Cheap, } PC \sqcap \exists \text{pointer} \sqcap \exists \text{monitor} \rangle$$

After the contraction operation, the remaining part of R is not yet satisfied by O. That is,  $\mathcal{O} \not\sqsubseteq K$  does not hold. To compute what is needed in order to realize the subsumption relation, retrieve solves

the CAP  $\langle \mathcal{L}, K, \mathcal{O}, \mathcal{T} \rangle$ . A solution for the previous CAP is:  $HK = \exists \text{monitor}$

If the searching agent—with the term agent used in its broadest sense—is interested in O, it must give up  $\text{HomePC} \sqcap \forall \text{pointer.} \forall \text{cost. Cheap}$  in its R and ask for further information about  $\exists \text{monitor}$ .

Retrieval performances have been usually evaluated in classical full text information retrieval in terms of precision and recall. Although such measures require large datasets to have any significance, it can be expected that semantic-based retrieval can provide at least a noteworthy improvement in precision, with respect to free-text probabilistic approaches.

## FUTURE TRENDS

As the Semantic Web initiative gets momentum, more and more resources described using structured descriptions—based on ontologies—will become available (Schwartz, 2003). Current and future application scenarios of the semantic-based retrieval techniques presented here include: electronic-marketplaces of tangible or intangible goods, skill management systems, mediators for Web service discovery and for grid-based computational resources, and dating and personnel recruitment agencies. The increased availability of semantically annotated descriptions will hence boost the emergence of knowledge-based systems able to take full advantage of these structured descriptions to obtain accurate and efficient retrieval. The framework and services described in this article are general enough to be used in the approximate search and retrieval of a variety of resources, and systems using them can provide—adopting different minimality criteria—logically motivated relevance-based rankings in the retrieval process. The necessary trade-off between expressivity and performance of semantic-based systems is likely to be ex-

exploited adopting various approaches. Among them, tableaux-based algorithms (Colucci et al., 2004); careful choice of constructs able to keep complexity tractable, as proposed for example with DL-Lite by Calvanese, De Giacomo, Lenzerini, Rosati, & Vetere (2004), and combined use of DL-based reasoners with classical relational databases to face scalability issues when dealing with large numbers of individuals (Horrocks, Li, Turi, & Bechhofer, 2004).

## CONCLUSION

We have presented and motivated new DL-based inference services for semantic-based resource retrieval. Currently, our approach is fully devised, and algorithms and a prototype system have been implemented for an ALN description logic. Work also is in progress to extend the approach to more expressive DLs, while keeping time performances still acceptable.

## REFERENCES

Arens, Y., Knoblock, C.A., & Shen, W. (1996). Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems*, 6, 99-130.

Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., & Patel-Schneider, P. (Eds.). (2002). *The description logic handbook*. Cambridge, MA: Cambridge University Press.

Baader, F., & Turhan, A. (2002). On the problem of computing small representations of least common subsumers. *KI 2002: Advances in Artificial Intelligence, 25th Annual German Conference on AI, Vol. 2479 of Lecture Notes in Computer Science* (pp. 99-113). Berlin: Springer.

Benatallah, B., Hacid, M.-S., Rey, C., & Toumani, F. (2003). Request rewriting-based Web service

discovery. *International Semantic Web Conference, vol. 2870 of Lecture Notes in Computer Science* (pp. 242-257). Berlin: Springer.

Brachman, R., & Levesque, H. (1984). The tractability of subsumption in frame-based description languages. *Proceedings of the 4th National Conference on Artificial Intelligence (AAAI-84)* (pp. 34-37). Cambridge, MA: Morgan Kaufmann.

Calvanese, D., De Giacomo, G., & Lenzerini, M. (1998). On the decidability of query containment under constraints. *Proceedings of the 17th ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98)* (pp. 149-158).

Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R., & Vetere, G. (2004). DL-Lite: Practical reasoning for rich Dls. *Proceedings of the 17th International Workshop on Description Logics (DL'04)*, vol. 104 of CEUR Workshop Proceedings.

Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2003a). Concept abduction and contraction in description logics. *Proceedings of the 16th International Workshop on Description Logics (DL'03)*, vol. 81 of CEUR Workshop Proceedings.

Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2003b). Logic based approach to Web services discovery and matchmaking. *Proceedings of the E-Services Workshop at ICEC'03*.

Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2004). Uniform tableaux-based approach to concept abduction and contraction in ALN DL. *Proceedings of the 17th International Workshop on Description Logics (DL'04)*, vol. 104 of CEUR Workshop Proceedings.

Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2003a). Abductive matchmaking using description logics. *Proceedings of the 18th*

*International Joint Conference on Artificial Intelligence (IJCAI 2003)* (pp. 337-342). Morgan Kaufmann.

Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2003b). Semantic matchmaking in a P-2-P electronic marketplace. *Proceedings of the Symposium on Applied Computing (SAC '03)* (pp. 582-586). ACM.

Di Noia, T., Di Sciascio, E., Donini, F., & Mongiello, M. (2003c). A system for principled matchmaking in an electronic marketplace. *Proceeding for the International World Wide Web Conference (WWW '03)* (pp. 321-330). ACM.

Di Noia, T., Di Sciascio, E., & Donini, F. (2004). Extending semantic-based matchmaking via concept abduction and contraction. *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, vol. 3257 of Lecture Notes in Computer Science (pp. 307-320). Springer.

Di Sciascio, E., Donini, F., Mongiello, M., & Piscitelli, G. (2001). A knowledge-based system for person-to-person e-commerce. *Proceedings Workshop on Applications of Description Logics (KI 2001)*, vol. 44 of CEUR Workshop Proceedings.

Donini, F. M., Lenzerini, M., Nardi, D., & Schaerf, A. (1996). Reasoning in description logics. In Brewka, G. (Ed.), *Principles of knowledge representation: Studies in logic, language and information* (pp. 193-238). Stanford, CA: CSLI Publications.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D., & Patel-Schneider, P.F. (2001). OIL: An ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2), 38-45.

Finin, T., Fritzson, R., McKay, D., & McEntire, R. (1994). KQML as an agent communication language. *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)* (pp. 456-463). ACM.

Gärdenfors, P. (1988). *Knowledge in flux: Modeling the dynamics of epistemic states*. Cambridge, MA: Bradford Books, MIT Press.

Genesereth, M.R. (1991). Knowledge interchange format. *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference* (pp. 599-600). Cambridge, MA: Morgan Kaufmann.

Gil, Y., & Ramachandran, S. (2001). PHOSPHORUS: A task based agent matchmaker. *Proceedings for AGENTS '01* (pp. 110-111). ACM.

Gonzales-Castillo, J., Trastour, D., & Bartolini, C. (2001). Description logics for matchmaking of services. *Proceedings of the Workshop on Application of Description Logics (KI 2001)*, vol. 44 of CEUR Workshop Proceedings.

Gruber, T. (1993). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5), 907-928.

Horrocks, I., Li, L., Turi, D., & Bechhofer, S. (2004). The instance store: DL reasoning with large numbers of individuals. *Proceedings of the 17th International Workshop on Description Logics (DL'04)*, vol. 104 of CEUR Workshop Proceedings.

Jacobs, N., & Shea, R. (1995). Carnot and Infosleuth: Database technology and the Web. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 443-444). ACM.

Kuokka, D., & Harada, L. (1996). Integrating information via matchmaking. *Journal of Intelligent Information Systems*, 6, 261-279.

Li, L., & Horrocks, I. (2003). A software framework for matchmaking based on Semantic Web technology. *Proceedings of the International World Wide Web Conference (WWW '03)* (pp. 331-339). ACM.

### **Description Logic-Based Resource Retrieval**

- Motro, A. (1988). VAGUE: A user interface to relational databases that permits vague queries. *ACM Transactions on Office Information Systems*, 6(3), 187-214.
- Paolucci, M., Kawamura, T., Payne, T., & Sycara, K. (2002). Semantic matching of Web services capabilities. *The Semantic Web—ISWC 2002, number 2342 in Lecture Notes in Computer Science* (pp. 333-347). Berlin: Springer-Verlag.
- Schwartz, David G., Open IS (2003). Semantics and the Semantic Web: The road ahead. *IEEE Intelligent Systems*, 18(3), 52-58.
- Sycara, K., Widoff, S., Klusch, M., & Lu, J. (2002). LARKS: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous agents and multi-agent systems*, 5, 173-203.
- Teege, G. (1994). Making the difference: A subtraction operation for description logics. *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning (KR'94)* (pp. 540-550). Cambridge, MA: Morgan Kaufmann.
- Trastour, D., Bartolini, C., & Priest, C. (2002). Semantic Web support for the business-to-business e-commerce lifecycle. *Proceedings of the WWW '02* (pp. 89-98). ACM.