Zoé Lacroix
Edna Ruckhaus
Maria-Esther Vidal (Eds.)

LNCS 8194

Resource Discovery

5th International Workshop, RED 2012
Co-located with the 9th Extended Semantic Web Conference, ESWC 2012
Heraklion, Greece, May 2012, Revised Selected Papers

Springer

Zoé Lacroix   Edna Ruckhaus
Maria-Esther Vidal (Eds.)

Resource Discovery

5th International Workshop, RED 2012
Co-located with the 9th Extended
Semantic Web Conference, ESWC 2012
Heraklion, Greece, May 27, 2012
Revised Selected Papers

Springer

Volume Editors

Zoé Lacroix
Arizona State University
Tempe, AZ, USA
E-mail: zoe.lacroix@asu.edu

Edna Ruckhaus
Universidad Simón Bolívar
Caracas, Venezuela
E-mail: ruckhaus@ldc.usb.ve

Maria-Esther Vidal
Universidad Simón Bolívar
Caracas, Venezuela
E-mail: mvidal@ldc.usb.ve

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-45262-8 e-ISBN 978-3-642-45263-5
DOI 10.1007/978-3-642-45263-5
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013954802

CR Subject Classification (1998): H.3, D.2, H.4, H.2, I.2

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains extended papers of the works presented at the 5th International Workshop on Resource Discovery, held on May 27, 2012. All the papers included in this volume went through a two-step peer-review process: they were first reviewed by the Program Committee for acceptance to the workshop, then they were extended after the workshop and went through a second review phase. Papers were evaluated in terms of technical depth, significance, novelty, relevance and completeness of the references, approach evaluation, and quality of the presentation. We accepted seven out of nine submissions. Our sincere thanks to the Program Committee members and external reviewers for their valuable input and for accepting to contribute to the multiple phases of the review process.

A resource may be a data repository, a database management system, a SPARQL endpoint, a link between resources, an entity in a social network, a semantic wiki, or a linked service. Resources are characterized by core information including a name, a description of its functionality, its URLs, and various additional quality of service parameters that express its non-functional characteristics. Resource discovery is the process of identifying, locating, and selecting existing resources that satisfy specific functional and non-functional requirements; also, resource discovery includes the problem of predicting links between resources. Current research includes crawling, indexing, ranking, clustering, and rewriting techniques, for collecting and consuming the resources for a specific request; additionally, processing techniques are required to ensure the access of the resources.

After four successful events, first in Linz, Austria, held jointly with IIWAS (2008), then in Lyon, France, collocated with VLDB (2009), next in Pontoise, France, held jointly with IIWAS (2010), and the fourth edition in conjunction with ESWC11, finally, the 5th International Workshop on Resource Discovery (RED 2012) was run again together with ESWC in Heraklion, Greece. The 5th International Workshop on Resource Discovery aimed at bringing together researchers from the database, artificial intelligence and Semantic Web areas, to discuss research issues and experiences in developing and deploying concepts, techniques and applications that address various issues related to resource discovery. This fifth edition focused on techniques to efficiently collect and consume resources that are semantically described. Approaches of special interest contribute to solve the resource discovery problem such as query rewriting in databases, service selection and composition in service-oriented architectures, social network navigational techniques, link prediction techniques, and strategies to process queries against linked data or SPARQL endpoints. We set up an exciting program that included two sessions and three invited talks: one session was on "Techniques for Resource Discovery" and the other section on "Applications of Resource Discovery." The first invited talk was on "Semantic Source

Modeling" by José Luis Ambite; the second, on the advantages of using semantic annotations in medical image visualization given by Alexandra La Cruz; finally, Edna Ruckhaus presented "Probabilistic Models and Reasoning Techniques to Detect Inconsistencies in Linked Data." We are grateful to the ESWC organizers for their support in making this meeting successful.

July 2013

Zoé Lacroix
Edna Ruckhaus
Maria-Esther Vidal

Organization

Workshop Chairs and Organizing Committee

Zoè Lacroix	Arizona State University, USA
Edna Ruckhaus	Universidad Simón Bolívar, Venezuela
Maria-Esther Vidal	Universidad Simón Bolívar, Venezuela

Program Committee

Maribel Acosta	AIFB, Karlsruhe Institute of Technology, Germany
José Luis Ambite	University Southern California, USA
Yudith Cardinale	Universidad Simón Bolívar, Venezuela
Oscar Corcho	Universidad Politécnica de Madrid, Spain
José Cordeiro	Polytechnic Institute of Setubal, Portugal
Valeria De Antonelis	Università degli Studi di Brescia, Italy
Alberto Fernández	Universidad Juan Carlos I, Spain
Norbert Fuhr	University of Duisburg, Germany
Manolis Gergatsoulis	Ionian University, Greece
Marlene Goncalves	Universidad Simón Bolívar, Venezuela
Andreas Harth	AIFB, Karlsruhe Institute of Technology, Germany
H.V. Jagadish	University of Michigan, USA
Nikos Kiourtis	National Technical University of Athens, Greece
Birgitta Koning-Ries	University of Jena, Germany
Günter Ladwig	AIFB, Karlsruhe Institute of Technology, Germany
Paraskevas Lekeas	Talk3, Inc., Chicago, USA
Maria Maleshkova	KMI, The Open University, UK
Anja Metzner	University of Applied Sciences, Augsburg, Germany
Pascal Moli	Nantes University, LINA, France
Fatiha Sais	LRI (Paris-Sud 11 University and CNRS), France
Sherif Sakr	National ICT Australia (NICTA) and University of New South Wales (UNSW), Australia
Miguel-Angel Sicilia	University of Alcalá, Spain
Hala Skaf-Moli	Nantes University, LINA, France

Dimitrios Skotas	University of Hannover, Germany
Andreas Thor	Universität Leipzig, Germany
Maciej Zaremba	DERI and National University of Ireland, Ireland
Trish Whetzel	Stanford Center for Biomedical Information Research, Stanford University, USA

Sponsoring Institutions

The National Science Foundation (IIS 0944126) http://www.nsf.gov/
DID-USB http://www.did.usb.ve

Table of Contents

Techniques for the Identification of Semantically-Equivalent Online Identities

Keith Cortis, Simon Scerri, and Ismael Rivera

Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland
firstname.surname@deri.org

Abstract. The average person today is required to create and separately manage multiple online identities in heterogeneous online accounts. Their integration would enable a single entry point for the management of a person's digital personal information. Thus, we target the extraction, retrieval and integration of these identities, using a comprehensive ontology framework serving as a standard format. A major challenge to achieve this integration is the discovery of semantic equivalence between multiple online identities (through attributes, relationships, shared posts, etc.). In this paper we outline a hybrid syntactic/semantic-based approach to online identity reconciliation. We also discuss the results of syntactic matching experiments conducted on real data, the current status of the work and our future research and development plans in this direction.

Keywords: semantic equivalence, online profile, online identity, personal information model, ontologies, social networks, semantic lifting, semantic web, syntactic matching, string metrics.

1 Introduction

The typical computer literate person is forced to create an online identity through a personal profile for each online account s/he would like to use. Considering the latest shift towards the usage of remote data management and sharing services, this necessity is evermore pressing. The leading online accounts now vary from general social networking platforms to specific email, instant messaging, calendaring, event scheduling and file-sharing services as well as business-oriented customer management services. Personal data in these accounts ranges from the more static identity-related information, to more dynamic information about one's social network as well as physical and online presence. In the context of this paper, we refer to all these kinds of personal data, stored on one of many distinguishable online accounts, as a user's 'online profile'.

At present, the situation results in personal data being unnecessarily duplicated over different platforms, without the possibility to merge or port any part of it [2], thus restricting users to also manage such data separately and manually. A lot of people create multiple online identities within multiple accounts to target different domains and not share their information with unnecessary people. For example it is important to distinguish between personal and business information, which process results in several time spent on updating all online accounts [12]. These facts have been observed in a survey[1] that we conducted, where 16% *always*, 20% *frequently* and 38% *sometimes*

[1] http://smile.deri.ie/online-profile-management

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 1–22, 2013.

use the same personal information within their ‘business’ (e.g. professional networks) and ‘administrative’ (e.g. e-commerce) profiles. However, the ‘social/private’ profiles of 12% *always*, 6% *frequently* and 40% *sometimes* contain the same personal information as their business/administrative profiles. Our aim is to enable the user to create, aggregate and merge multiple online identities into one, through the di.me[2] userware - a single access point to the user’s personal information sphere [29]. The latter also refers to personal data on a user’s multiple devices (e.g. laptops, tablets, smartphones). This makes the di.me userware sophisticated and novel since it does not only ‘attack’ the distributed/duplicated online identity management problem, but targets the integration of personal information found across multiple local and remote sources. The already integrated data is stored as a machine-processable representation of the user’s Personal Information Model (PIM) using the Resource Description Framework (RDF)[3]. The PIM administers unique personal data that is of interest to the user, such as the user’s singular online identity, files and emails. It is an abstraction of the, possibly, multiple occurrences of the same data as available on multiple online accounts and devices. The users have complete control over their accessed personal data, since we do not target sharing of personal information, and this valuable information is stored on a personal server.

Given that our main motivation involves the retrieval of personal data, the detection of similarities between multiple online identities belonging to a particular person, is also useful to address certain privacy and trust issues. Survey results show that 55.1% of the people *quite a bit*, *always*, *frequently* or *sometimes* like to remain partly or fully anonymous, such that the provided details cannot be connected to them in real life, even though the information provided is not fictitious (e.g. using an alias, alternate email, etc.). Moreover, the same survey found that 10% *always*, 24% *frequently* and 28% *sometimes* use different personal profiles that show a selection of personal details to different people. Therefore, our technique can be useful to warn people when privacy concerns arise from an involuntary link between different online identities, as obtained from the same or different social networks. Our technique can also be useful to control privacy-sensitive information when it comes to the automatic sharing of personal data within social network sites [6]. Such a ‘Privacy Advisory’ would be appreciated by most people, as confirmed by a survey result, which outlines that 48% would *extremely* favour, 30% favour the idea *quite a bit*, whilst 14% are *moderately* in favour of a system that can warn you against sharing sensitive personal data or about untrustworthy users. These results provide us with additional motivation us further towards achieving our goals.

The main research contributions presented in this paper are:

- A hybrid syntactic/semantic-based profile matching approach towards solving the online identity reconciliation problem,
- An ontology-based approach for extending the semantic matching capabilities,
- Experiments conducted in preparation of the final implementation of the syntactic matching part of the approach.

[2] `http://www.dime-project.eu/`

[3] `http://www.w3.org/RDF/`

The profile matching task is not straightforward for two main reasons. First, no common standards exists for modelling profile data in online accounts [24]. This makes the retrieval and integration of federated heterogeneous personal data instantly a hard task, because profile attributes from different sources cannot be instantly mapped. In addition, even though two different source account schemas might at face-value appear to be similar, their semantics might differ considerably, resulting in several technical problems to be overcome. A second problem is that the nature of some of the personal data on digital community ecosystems [14], such as known contacts (resources) and presence information, is dynamic. To address these difficulties, the use of standard knowledge formats which are able to handle both the more static as well as dynamic profile data is proposed. In particular, the Nepomuk Contact Ontology (NCO) [22] and Digital.Me LivePost Ontology (DLPO)[4] cover the necessary knowledge representation requirements. Both ontologies form part of an integrated ontology framework consisting of a set of re-used, extended as well as new vocabularies provided by the OSCA Foundation (OSCAF)[5] (only the most relevant ontologies will be mentioned in this paper). Our approach is to map and integrate various online user profiles onto this one standard representation format.

The first stage of this approach consists of the relatively straightforward extraction of the user's semi/unstructured personal information, as is made available by online account APIs. These representations maintain links to the source account as well as to the external identifiers of the specific online profile attributes. Additionally, all attributes extracted from a particular online profile, are aggregated as a PIM representation, into what we refer to as the user's 'super profile'. The second stage of our approach targets the mapping of attributes for each of the represented online profiles with equivalent attributes for the super profile. The use of ontologies and RDF as the main data representation means that the mapping techniques we pursue consider both syntactic as well as semantic similarities in between online profile data representing the online identities. Semantic lifting is performed in our approach as opposed to traditional ontology matching, since we are discovering resources from a user's profile which are then mapped to the mentioned ontologies within our ontology framework. Linguistic analysis is also be performed on certain attributes, in order to further decompose them into sub-types, which can enhance the semantic lifting process. In the next step we then attempt to discover semantic equivalence between online identities that are known in multiple online accounts, based on the results of individual attribute matching. An appropriate semantic equivalence metric is one of the requirements for aspiring self-integrating system [25], such as the di.me userware.

Numerous techniques may be required for discovering if two or more online identities are semantically equivalent. The most popular techniques are syntax based, i.e. a value comparison is performed on the various person profile attributes of an online identity. Our ontology-based approach allows us to extend the matching capabilities 'semantically', ensuring more accurate results based on clearly-specified meanings of profile attribute types, as well as through an exploration of their semantic (in addition

[4] `http://www.semanticdesktop.org/ontologies/dlpo/` —currently a candidate OSCAF submission

[5] `http://www.oscaf.org/`

to syntactic) relatedness. The discovery of semantically equivalent online identity representations results in their semantic integration to the same person at the PIM level of the user's data. This will also enable the privacy advisory against untrustworthy people. In the remainder of the paper we start by discussing and comparing related work in Section 2. Details on our approach are then provided in Section 3. An update of the current status and prototype implementation is provided in Section 4. The experiments conducted are discussed in Section 5, before a list of our targeted future aspirations and concluding remarks are provided in Sections 6 and 7 respectively.

2 Related Work

The process of *matching* takes two schemas/ontologies (each of which is made up of a set of discrete entities such as tables, XML[6] elements, classes, properties, etc.) as an input, producing relationships (such as equivalence) found between these entities as output [30]. COMA++ [3] is one of the most relevant schema and ontology matching tools that finds out the semantic correspondences among meta-data structures or models. Given that these matching problems are overcome, it would benefit service interoperability and data integration in multiple application domains. Several techniques and prototypes were implemented for solving the matching problem in a semi-automatic manner such as Falcon-AO (ontology matching) [16], thus reducing manual intervention. Our approach is different to the mentioned traditional approach since we are not concerned with matching two conceptualisations (schemas or ontologies), but an online account schema e.g. a social network, to an ontology or set of ontologies. We refer to this process as semantic lifting, since we are lifting semi/unstructured information (the user's profile attributes) from a schema as discussed in Section 3.1, which is manually mapped to an interoperable standard (ontology framework) as discussed in Section 3.2.

Findings in [17] suggest that provided enough information is available, multiple user profiles can be linked at a relatively low cost. In fact, their technique produces very good results by considering a user's friends list on multiple online accounts. Earlier approaches rely on just a specific Inverse Functional Property (IFP) value e.g. email address or name [20],[12]. However, as pointed out in [5], IFP assumptions are rather shallow, since users are able to create multiple accounts even within the same social network (e.g. a business-related profile, social profile, etc.) each time using different identification, e.g. email addresses.

A number of approaches rely on formal semantic definitions, through the use of ontologies and RDF, to enable portability of online profiles. The work by [26] presents an online application that transforms a user's identity on a social network (Facebook) into a portable format, based on the Friend of a Friend (FOAF) ontology[7]. The approach described in [24] goes on step forward, attempting to integrate multiple online profiles that have been converted to FOAF. As opposed to IFP approaches, this approach takes into consideration all (FOAF) profile attributes, assigning different importance levels and similarity measures to each. Although FOAF enables a much richer means for profile attribute comparison, we use a more comprehensive conceptualisation through the

[6] `http://www.w3.org/XML/`

[7] `http://www.foaf-project.org/`

NCO, which is integrated into a comprehensive ontology framework. This integration enables attributes in multiple profiles to be semantically related to unique, abstract representations in the user's PIM. Once the technique in [24] sees the profiles transformed to a FOAF representation, a number of techniques are used for syntactic matching between short strings and entire sentences. In addition, the syntactic-based aspect of our matching will also perform a Linguistic Analysis to yield further information about the typed profile attributes. Named Entity Recognition (NER) can discover more specific types than the ones known (e.g. identifying city and country in a postal address) and recognise abbreviations or acronyms in attribute labels.

The different standard formats for representing data within heterogeneous information sources, such as different social networks, can lead to several data variations like typographical and optical recognition character recognition errors, acronyms, and abbreviations. Often the data is also semi/unstructured, resulting in a matching process being vital for the targeted online profile integration. In [4], the authors conduct an in-depth analysis on several string matching methods that can be used to calculate the similarity between two fields, which are of a string data type format. Edit-distance metrics such as the Levenshtein [18] distance, calculate the distance between string 's1' and 's2', as the cost of the best sequence of edit operations that is able to convert 's1' to 's2'. On the other hand, the Jaro metric [15] and its variants are based on the number and order of common characters between two strings. Other metrics are token-based such as term frequency-inverse document frequency (TF-IDF), and hybrid distance functions such as the Monge-Elkan recursive algorithm [21] and the "soft" version of TF-IDF. For both sets of metrics, the word order within a string is not normally important at all, since each string is converted into several token multi-sets, with the similarity being calculated on each multi-set.

Many approaches enhance the otherwise syntactic-based profile matching techniques with a semantic-based extension. In particular, the above-cited work by Raad et. al. is supplemented with an Explicit Semantic Analysis [11], whose aim is to detect semantic similarity between profile attributes through the computation of semantic relatedness between text, based on Wikipedia. A similar approach [31] uses snippets returned from an online encyclopedia to measure the semantic similarity between words through five web-based metrics. Our approach will consider semantic relatedness to determine similarity between entities not only based on their labels or values, but also on a semantic distance to other relevant concepts. For example, although an address in one profile might consist of just the city, and another address might refer to only the country, the fact that the city in the first profile is known to be in the country defined for the second profile will be considered as a partial match.

The calculation of such measures within different systems or domains is a very important task, given the increase in access to heterogeneous and independent data repositories [9]. Research efforts conducted by [33] identify three common approaches for calculating the semantic distance between two concepts, namely i) the knowledge-based approach which uses remote Knowledge Bases (KBs) such as WordNet[8] (count edge distance) [7], ii) lexico-syntactic patterns (makes binary decisions), and iii) statistical measures (uses contextual distributions or concept co-occurrences). The mentioned

[8] http://wordnet.princeton.edu/

techniques are not relevant for certain cases, as the concept distances cannot be calculated. This means that such a process is not straightforward, especially if a personal KB is used, where a good distance metric needs personal adjustments in order to cater for a particular user's needs. Normally for a personal ontology (can be domain specific), in our case the PIM, several concepts are not available within remote KBs. Therefore, it is impossible to calculate the semantic distance between two concepts, if remote KBs are used alone. Hierarchical semantic KBs such as the ones constructed on an "is a" structure, can be fundamental for finding the semantic distance between concepts [19].

There is one major distinction between our approach and the semantic-based approaches described above. Although remote KBs such as DBpedia[9] are to be considered as a backup, the KB on which we initially perform a similarity measure is the user's own PIM. The PIM is populated partly automatically - by crawling data on the user's devices, applications and online accounts, and partly by enabling the user to manually extend the representations of their own mental models. The advantage here is that the PIM contains information items that are of direct interest to the user, and is thus more relevant to the user than external structured or partly structured KBs. Therefore, the semantic matching of profiles is bound to yield more accurate results, based on a KB that is more personal and significantly smaller.

3 A Semantic Approach to Online Identity Reconciliation

Our online profile (instance) matching approach will involve four successive processes (A-D), as outlined by Fig. 1 and discussed below.

3.1 Retrieval of User Profile Data from Online Accounts

The first step is to retrieve personal information from various online accounts, such as Facebook, Twitter and LinkedIn, and is fairly straightforward once the required API calls are known. We target several categories of online profile data such as the user's own identity-related information, their online posts, as well as information about the user's social network, including the identities and posts shared by their contacts.

3.2 Semantic Lifting of User Profile Data

Once online profile data has been retrieved from an online account, it is mapped to two particular ontologies in our ontology Framework. Identity-related online profile information is stored as an instance of the NCO Ontology, which represents information that is related to a particular contact. The term 'Contact' is quite broad, since it reflects every bit of data that identifies an entity or provides a way to communicate with it. In this context, the contact can also refer to the user's own contact information. Therefore, both the user and their contacts as defined in an online profile are represented as instances of *nco:Contact*. Presence and online post data for the user is stored as instances of the DLPO, a new ontology for the representation of dynamic personal information

[9] `http://dbpedia.org/`

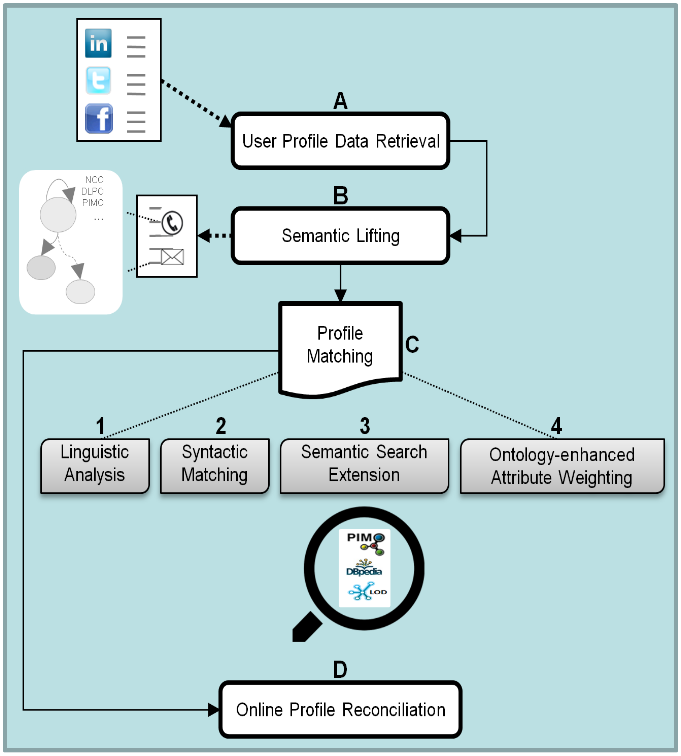

A
User Profile Data Retrieval
NCO
DLPO
PIMO
...
B
Semantic Lifting
Profile Matching
C
1
Linguistic Analysis
2
Syntactic Matching
3
Semantic Search Extension
4
Ontology-enhanced Attribute Weighting
PIMO
DBpedia
LOD
D
Online Profile Reconciliation

Fig. 1. Approach Process

that is popularly shared in online accounts, such as multimedia posts (video/audio/image), presence posts (availability/activity/event/checkin), messages (status messages/-comments) and web document posts (note/blog posts).

Fig. 2 demonstrates how the above ontologies can be used to store online profile data from an online account (OnlineAccountX). The figure also shows the user's super profile (di.meAccount). An explanation of how the other ontologies in the framework can be used to effectively integrate the two profiles once semantic equivalence is discovered, is provided later on. The upper part of the figure refers to the T-box, i.e. the ontological classes and attributes, whereas the lower part represents the A-box, containing examples of how the ontologies can be used in practice (straight lines between the A- and T-box denote an instance-of relationship).

The attributes of the online user profiles will be mapped to their corresponding properties within our ontology framework. The example shows five identity-related profile attributes that have been mapped to the NCO (affiliation, person name, organisation, phone number, postal address). Presence-related profile information is also available in the form of a complex-type 'livepost', consisting of a concurrent status message - "Having a beer with Anna @ESWC12 in Iraklion", a check-in (referring to the *pimo:City* representation for Heraklion through *dlpo:definingResource*) and an event post (referring to the *pimo:Event* instance representing the conference through

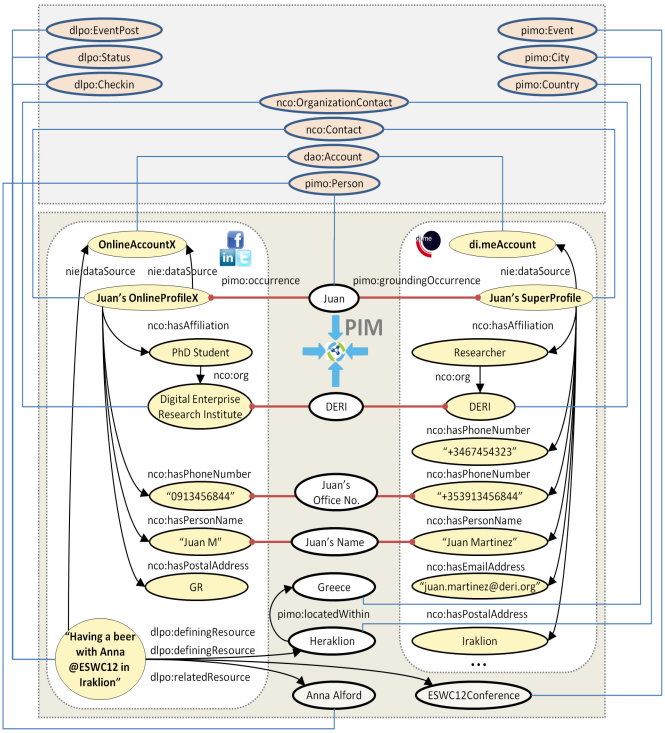


Fig. 2. Approach Scenario

dlpo:definingResource). *dlpo:definingResource* defines a direct relationship between a 'livepost' subtype and a PIM item. A person, "Anna" is also tagged in this post, as referred by *dlpo:relatedResource*. This property creates a semantic link between a 'livepost' and the relevant PIM items.

3.3 User Profile Matching

Our approach towards matching the user profile attributes i.e. metadata matching, considers the data both at a semantic and syntactic level. It involves four successive processes as outlined within the third level (C) of Fig. 1. Before we describe the matching technique, we first provide some formal notations to help with its understanding.

A person's online identity I is defined through a set of online profiles as retrieved from multiple online profiles p (refer to Section 3.1 above). In turn, each online profile p has a number of attributes a. These relationships are formally defined as:

$$\begin{aligned} I &= \{p_1, p_2, \ldots, p_m\} \\ \text{where: } p &= \{a_1, a_2, \ldots, a_n\}, \\ m, n &\in \mathbb{N} \end{aligned} \tag{1}$$

Different attributes correspond to a different attribute type, such that:

$$type(a) \subset T \tag{2}$$

where T corresponds to the set of attribute types supported by NCO[10] (e.g. address, phone number, name, email, etc.).

If we have two profiles $p_1 = \{a_1, a_2, \ldots, a_o\}$, and $p_2 = \{a_1, a_2, \ldots, a_q\}$, where o,$q \in \mathbb{N}$; the similarity between two attributes in two profiles is defined as follows:

$$sim_{\ a} = match\{a_o, a_q\} \tag{3}$$

For the most generic matching technique, only the common attributes amongst both profiles will be initially considered for calculating the profile similarity score, thus $type(a_o) = type(a_q)$. This type-checking process is done for each online profile through the semantic lifting process as specified in Section 3.2 above.

The generic similarity of a profile is then defined as:

$$sim_{\ p} = \frac{\sum_{r=1}^{t}(sim_a)}{t} \tag{4}$$

where $r, t \in \mathbb{N}$ and t refers to the number of attribute pairs of the same type between two profiles.

Finally, the set of attributes of an online profile as aggregated from an online account (depending on its schema) is defined as a function over p: $a(p)$. If the set of attributes of an online identity I is defined as a similar function $a(I)$, then:

$$a(p) \subset a(I) \tag{5}$$

meaning that the range of distinct attributes making up an online identity corresponds to the aggregation of distinct profile attributes defined by each online account.

The above definitions are only the generic ones for understanding the outcome of the reconciliation process. In the next sub-sections we describe additional techniques which extend them, such as adding weights to sim_a in Def. 4; and comparing attributes of a different type in Def. 3.

Linguistic Analysis. Once the transformation and mapping of the user's profile data to its RDF representation has been performed, a matching process is initiated against the user's PIM in order to find similar attributes or links and relations between them. In the case that the profile attribute is known to contain an atomic value (e.g. a person's name, phone number, etc.), no further linguistic analysis is performed. However, profiles attributes may contain more complex and unstructured information such as a postal address (e.g. "42 Upper Newcastle Road, Lower Dangan, Galway, Ireland"), or a full person name (e.g. "Dr. John Doe Jr."). For such attributes, a deeper linguistic analysis is required to discover further knowledge from their values; concretely, a decomposition into different entities or concepts is the goal pursued. In the postal address example, the aim is to find out that '42 Upper Newcastle Road' refers to the street address which is

[10] http://www.semanticdesktop.org/ontologies/2007/03/22/nco/

the most specific part of the address information, ‘Lower Dangan’ to an area or district, ‘Galway’ to a city, and ‘Ireland’ to a country. In the full person name example, the goal is to further classify ‘Dr.’ as a name prefix, ‘John’ as a first name, ‘Doe’ as a surname, and ‘Jr.’ as a name suffix. The techniques applied to extract or decompose the attribute values are regular expressions and gazetteer lookups. Typically both techniques work well when the domain or structure is known. Therefore, the algorithm distinguishes profile attributes by type or nature, which is known at this stage, to apply different regular expressions and use different gazetteers. Abbreviations and acronyms are also covered in this analysis by including entries for them in the gazetteers (e.g. a gazetteer for countries also includes the ISO 3166 codes).

Syntactic Matching. Straightforward value matching is applied on attributes having a non-string literal type (e.g. birth date or geographical position), since these have a strict, predefined structure. For attributes of type string (*xsd:string*), if their ontology type (e.g. person name) is either known beforehand or discovered through NER, standard string matching is applied. In both cases, the matching takes as input the attribute in consideration against PIM instances of a similar type. For example, in Fig. 2., the label of the organisation (*nco:OrganizationContact* instance) specified within the *nco:org* property for the user’s online account profile (i.e. ‘Digital Enterprise Research Institute’) is matched against other organisation instances within the PIM. The super profile instance ‘DERI’ is one example of other PIM instances having the same type. The fact that in this case one of these two equivalent profile organisation attributes is an acronym for the other, one will be taken into consideration by the employed string matching technique.

A string matching metric is used for syntactically matching user profile attribute values that are obtained from an online account to attribute values that are stored in the PIM KB. The recursive field matching algorithm proposed by Monge and Elkan [21] is applied for matching string values. A degree of 1.0 signifies that string ‘A’ and string ‘B’ fully match or one string abbreviates the other. On the other hand, a degree of 0.0 signifies that there is no match between two strings. All sub-fields of string ‘A’ and string ‘B’ are also compared against each other, where the sub-fields of string ‘A’ are expected to correspond to one particular sub-field of string ‘B’ with which it obtains the highest score. The mean of the highest scores calculated is taken as the overall matching score. The Monge-Elkan string matching metric (6) is defined as follows:

$$match\,(A,B) = \frac{1}{|A|}\sum_{i=1}^{|A|} \max\{sim'\,(A_i,B_j)\}_{j=1}^{|B|} \tag{6}$$

—where *sim’* is a particular secondary distance function that is able to calculate the similarity between two tokens ‘A’ and ‘B’. This metric was mainly chosen since it holds for matching an attribute value to its abbreviation or acronym, unlike some edit-distance metrics considered, such as Jaro and Jaro-Winkler [32].

This advantage is attributed to the recursive nature of the algorithm, which subdivides strings at token boundaries, alternatively to separators such as commas; therefore, making it possible to handle sub-fields, sub-sub-fields, and so forth. This makes it more likely to find a match between a string and its corresponding incomplete string—which can be in several formats—irrespective of its domain, e.g. postal address, mailing address, etc. This technique considers abbreviations that are either: i) a prefix, ii) a combination of a prefix and suffix, iii) an acronym of the expanded string, or iv) a concatenation of prefixes from the expanded string.

Semantic Search Extension. Once the syntactic matching is complete, a semantic search extension process follows. Referring again to our example, the user's address known for the super profile (di.meAccount) is listed as 'Iraklion', and is related to an instance of a *pimo:City*, 'Heraklion'. The one just retrieved from the online account profile (OnlineAccountX) refers to 'GR', which is found to be related to a particular instance of *pimo:Country*, 'Greece'. Although the two address attributes do not match syntactically, they are semantically related. Therefore, the attribute similarity function defined in (3) is not true for semantically related attributes since, the attribute types defined in (2) are not the same, thus $type(a_o) \neq type(a_p)$. Given that the profile in question is the user's, it is highly likely that through some other data which is either automatically crawled or enriched by the user, the PIM contains references to both these locations, and that semantic relationships exist in between. In the example, through the PIM KB, the system already knows that the city and country instances related to both addresses are in fact related through *pimo:locatedWithin*. This constitutes a partial semantic match, to be taken into consideration when assigning semantic-based attribute weights. If such data did not exist within the PIM KB (main KB for matching), remote KBs, such as DPBedia or any other dataset that is part of the Linked Open Data cloud[11], will be accessed to determine any possible semantic relationship. Another example centres around Juan's two roles, listed as a 'Researcher' for 'DERI' within his super profile, and as a 'PhD Student' for 'Digital Enterprise Research Institute' on his online account. Although less straightforward, a semantic search here would largely support the syntactic search in determining that there is a high match between these two profile attributes, after finding that DERI is a research institute which employs several Researchers and PhD students.

Ontology-Enhanced Attribute Weighting. To discover semantic equivalence between persons in online profiles or otherwise, an appropriate metric is required for weighting the attributes which were syntactically and/or semantically matched. Factors that will be taken into account by the metric are the total number of attributes that were mapped to our ontology framework, the number of syntactically matched attributes, and the number of attributes matched based on the semantic search extension. In addition, ontology-enhanced attribute weights are an added benefit of our ontology framework over other ontologies such as FOAF. Attribute constraints defined in the NCO ontology, such as cardinality and IFPs, enable the assignment of different predefined weights to the attributes. Thus, the properties that have a maximum or an exact cardinality of 1 have a higher impact on the likelihood that two particular profiles are semantically equivalent. Carrying even a higher predefined weight are IFPs, which uniquely identify one user. Examples of attributes having cardinality constraints are first name, last name and date

[11] `http://lod-cloud.net/`

of birth, whereas an example of an IFP is a private email address or a cell phone number. As profile attributes such as username, affiliation, postal address, and roles within organisations have no cardinality constraints defined in the ontology, they have a lower weight. All the factors mentioned will effect the computation of Def. 3, since it will vary according to the attribute type.

Once the initial matching score of an attribute pair from two different online profiles is calculated, two different approaches can be used to calculate the final score. In the first, the initial matching score is used directly. In the second, a particular threshold is set for each attribute type, and the final score depends on whether it is over or below this threshold. In the former case the final score is taken as '1' (meaning that it is a full match), elsewhere the final score is '0'. Further on in this paper we will present experiments to determine an appropriate threshold for each profile attribute.

3.4 Online Profile Reconciliation

Based on the score obtained from the attribute weighting metric as defined in Def. 4, we define a threshold for discovering semantic equivalence between elements of a user's online profiles, i.e. personal identity, and social network (i.e. contact) information that is already known and represented at the PIM level. A user can then be suggested to merge duplicate contact online profiles as a whole in order to reconcile them to the same person representation, both depending on the defined threshold. A user would also have the ability to mark contacts for the same unique person as 'known' over multiple online accounts.

The actual integration of semantically-equivalent personal information across distributed locations is realised through the 'lifting' of duplicated data representations onto a more abstract but unique representation in the PIM. The Personal Information Model Ontology (PIMO) [27] provides a framework for representing a user's entire PIM, modelling data that is of direct interest to the user. By definition, PIMO representations are independent of the way the user accesses the data, as well as their source, format, and author. Initially, the PIM will be populated with any personal information that is crawled from a user's particular online account or device. Therefore, if there is no match of a particular entity, a new instance is created. In the example shown in Fig. 2, Juan's PIM (grey area) 'glues' together all the things he works with uniquely, irrespective of their multiple 'occurrences' on different devices and/or online accounts. First and foremost, the PIM includes a representation for the user himself, as a *pimo:Person* instance. This instance refers to the two shown profiles through the *pimo:occurrence* property, which relates an 'abstract' but unique subject to one or more of its occurrences (*pimo:groundingOccurrence* is a special sub-property that is only used to identify the super profile). For example, the unique *pimo:City* instance has multiple occurrences in multiple accounts, and is related to both Juan's postal address and his check-in as defined on his online account. The advantage of using ontologies is evident here - resources can be linked at the semantic level, rather than the syntactic or format level. For example, although the user's name or organisation differ syntactically, the discovery that they are semantically equivalent is registered within the PIM.

Table 1. Ontology Mapping of Facebook Attributes

Query	Attribute	Ontology Mapping
https://graph.facebook.com/username?fields=id,name,first_name,last_name,email,gender,username,bio,birthday,location,work,picture,link,interests	id	$\xrightarrow{nco:contactUID}$ <value>
	name	$\xrightarrow{nco:hasPersonName}$ nco:PersonName $\xrightarrow{nco:fullname}$ <value>
	first-name	$\xrightarrow{nco:hasPersonName}$ nco:PersonName $\xrightarrow{nco:nameGiven}$ <value>
	last-name	$\xrightarrow{nco:hasPersonName}$ nco:PersonName $\xrightarrow{nco:nameFamily}$ <value>
	email	$\xrightarrow{nco:hasEmailAddress}$ nco:EmailAddress $\xrightarrow{nco:emailAddress}$ <value>
	gender	$\xrightarrow{nco:gender}$ <nco:male/nco:female>
	username	$\xrightarrow{nco:hasPersonName}$ nco:PersonName $\xrightarrow{nco:nickname}$ <value>
https://graph.facebook.com/username?fields=friends	bio	$\xrightarrow{nao:description}$ <value>
	birthday	$\xrightarrow{nco:hasBirthDate}$ nco:BirthDate $\xrightarrow{nco:birthdate}$ <value>
	location	$\xrightarrow{nco:hasPostalAddress}$ nco:PostalAddress $\xrightarrow{nao:prefLabel}$ <value>
	work	$\xrightarrow{nco:hasAffiliation}$ nco:Affiliation $\xrightarrow{nco:role/start/end/org}$ <value>
	picture	$\xrightarrow{nco:photo}$ <value>
	link	$\xrightarrow{nco:url}$ <value>
	interests	$\xrightarrow{nco:hobby}$ <value>
https://graph.facebook.com/username?fields=statuses	id	$dlpo : LivePost \xrightarrow{nao:externalIdentifier}$ <value>
	updated_time	$dlpo : LivePost \xrightarrow{dlpo:timestamp}$ <value>
	message	$dlpo : LivePost \xrightarrow{dlpo:textualContent}$ <value>
	likes	$dlpo : LivePost \xrightarrow{dlpo:favouritedBy}$ nco:PersonContact
	comments	$dlpo : LivePost \xrightarrow{dlpo:hasReply}$ dlpo:Comment

4 Implementation

This section describes the development progress so far. The current prototype employs the Scribe OAuth Java library[12] to retrieve data from a user's LinkedIn, Facebook (as shown in the example provided in Table 1 above), and/or Twitter profile. Scribe supports major 1.0a and 2.0 OAuth APIs such as Google, Facebook, LinkedIn, Twitter, Foursquare, Dropbox and Flickr, and thus it can be used to extend future profiles.

Table 1 shows different types of Facebook service calls that our prototype supports (column one). The first retrieves a user's profile data, whereas the second retrieves a user's contact profile data ('id' and 'name' only, unless authorised by the contact). The third query retrieves status updates from the user, including any 'likes' and/or 'replies' related to them. The calls return a set of Facebook profile data for the user or their connections, of which we currently map the shown list (column two) to the specific concepts and properties in our ontology framework (column three). The first set of ontology properties in the third column are attached to the *nco:PersonContact* instance representing the user or one of their contacts (omitted from the Table), whereas the second set of ontology properties are attached to the respect *dlpo:LivePost* instance. Both instances are linked to the online account from which they where retrieved via *dao:source*, this case being a representation of the Facebook online account.

Since the social data from the LinkedIn API is returned in XML, a transformation of this data into an RDF representation is required for mapping it to our ontologies. The translation between XML to RDF is quite a tedious and error-prone task, despite the available tools and languages. Although an existing approach is to rely on Extensible Stylesheet Language Transformations (XSLT)[13], the latter was designed to handle

[12] https://github.com/fernandezpablo85/scribe-java

[13] http://www.w3.org/TR/xslt

XML data, which in contrast to RDF possesses a simple and known hierarchical structure. Therefore, we use the XSPARQL [1] query language. XSPARQL (W3C member submission) provides for a more natural approach based on merging XQuery[14] and SPARQL[15] (both W3C Recommendations) into a novel language for these transformations. Given that SPARQL operates in RDF and XQuery in the XML world, this brings both representations closer together. One restriction in XSPARQL is that it does not handle JSON data. Therefore, if any social data retrieved from a particular service account, such as Facebook and Twitter is returned in JSON, this will first have to be transformed to XML, in order for XSPARQL to be used. The JSON-lib[16] java library is being used for performing such an operation. The reason being that this library has the functionality of converting beans, maps, collections, java arrays and XML to JSON and vice-versa. The transformation from XML (LinkedIn/Twitter/Facebook) data to RDF data (using Turtle[17] as the serialization format) is declaratively expressed in a XSPARQL query.

For the linguistic analysis and NER process, presented in Section 3.3, the General Architecture for Text Engineering (GATE) platform has been selected. It is an open-source software tool for Language Engineering (LE). Its architecture is able to decompose complex processes—or *'pipelines'*— into smaller tasks or modules, thus distributing the work to the appropriate components, whilst ensuring that each component interacts with each other as required. The GATE framework can be extended and customised according to a user's specific needs, thus reducing the time that developers normally spend for building new LE systems or tweaking existing ones. It has a built-in Information Extraction (IE) component set – A Nearly-New IE System (ANNIE) [10] which contains several main processing resources for common NLP tasks, such as a: tokeniser, sentence splitter, Part-of-speech (POS) tagger, gazetteer, finite state transducer, orthomatcher and coreference resolver. Some pre-defined gazetteers for common entity types (e.g. countries, organizations, etc.), were extended with acronyms or abbreviations where necessary, according to the required needs. The NER capability is to be extended so that it will be able to extract other entities besides the ones that it already caters for. The *Large KB Gazetteer* module is used to make use of the information stored within the user's PIM, since it can get populated dynamically by loading any ontology from RDF data. The gazetteer can then be used to retrieve lookup annotations that have both instance and class URI.

Listing 1.1 shows an example of online profile data retrieved from the Facebook account for user "Juan Martinez". The RDF representation (in Turtle syntax) shows how the data is mapped to our ontology framework, through the XSPARQL transformer. The Facebook account representation (_:acct1 as an instance of *dao:Account*) contains references to two contacts known within (_:c1, _:c2 as instances of *nco:PersonContact*), one of which (_:c1) is the Juan's own contact representation. Shown attached to Juan's contact instance is a series of identity-related information as well as one status message post (instance of *dlpo:Status*). This example highlights the comprehensiveness of

[14] http://www.w3.org/TR/xquery/

[15] http://www.w3.org/TR/sparql11-query/

[16] http://json-lib.sourceforge.net/

[17] http://www.w3.org/TR/turtle/

```
#Facebook Profile Metadata
_:acct1 a dao:Account .
_:acct1 dao:accountType "Facebook" .
_:c1 a nco:PersonContact .
_:c1 nie:dataSource _:acct1 .
_:c1 nco:contactUID "1004545677" .
_:c1 nco:hasPersonName _:cn12 .
_:cn12 a nco:PersonName .
_:cn12 nco:nameGiven "Juan" .
_:cn12 nco:nameFamily "Martinez" .
_:cn12 nco:fullname "Juan Martinez" .
_:cn12 nco:nickname "juanmartinez" .
_:c1 nco:hasAffiliation _:pos8 .
_:pos8 a nco:Affiliation .
_:pos8 nao:externalID "11446645687695" .
_:pos8 nco:role "Software Developer" .
_:pos8 nco:start "2006-01-01T00:00:00Z
     "^^xsd:dateTime .
_:pos8 nco:org _:org16 .
_:org16 a nco:OrganizationContact .
_:org16 nie:title "Ingeneria Ltd." .
...
_:stms644819790 a dlpo:Status .
_:stms644819790 nie:dataSource _:acct1 .
_:stms644819790 nao:externalIdentifier "
     s6448190" .
_:stms644819790 dlpo:timestamp
     "2012-10-02T14:51:01" .
_:stms644819790 dlpo:textualContent "
     rainy day in Galway" .
...
_:c2 a nco:PersonContact .
_:c2 dao:source _:acct1 .
_:c2 nco:contactUID "12" .
_:c2 nco:hasPersonName _:pn22 .
_:pn22 a nco:PersonName .
_:pn22 nco:fullname "Anna Alford" .
_:pn22 nco:nickname "aalford" .
...

#PIM Metadata
_:PIM a pimo:PersonalInformationModel .
_:PIM pimo:creator _:juanUser .
_:juanUser a pimo:Person .
_:juanUser pimo:occurrence _:c1 .
_:juanUser pimo:occurrence _:c18 .
...
_:juanUser foaf:knows _:annaUser .
_:annaUser pimo:occurrence _:c2 .
...
```

Listing 1.1. User Profile Transformer Output and PIM Integration

our integrated ontology framework in dealing with various types of online profile data, when compared to other integrated ontology approaches, such as the use of FOAF and Semantically-Interlinked Online Communities[18]. More importantly, it also illustrates how integration of online profile data is achieved at the semantic level. Once the two contacts in the online profile (including the one for the user) are discovered to be semantically equivalent to persons that are already represented in the PIM, a link is created between them through *pimo:occurrence*. The PIM Metadata at the bottom of Listing 1.1 demonstrates how the same unique person representations at the level of the PIM can point to multiple occurrences for that person, e.g. contacts for that person as discovered in online accounts, including the ones just retrieved from Facebook.

5 Experiments

In this Section we describe a number of experiments that have been carried out in preparation for the final implementation stage. In particular, we try to determine *i) which of the available similarity metric libraries performs best for the various types of personal information we are trying to match*, and *ii) which secondary distance function performs best with the Monge-Elkan recursive metric*, which was identified as the most suitable technique for our syntactic matching task (Section 3.3). In some instances, the di.me ontologies provide for more fine-grained data structures than some of the targeted data sources. In the context of this paper, a relevant case arises when retrieving data such as addresses and names from a contact's online profile. Whereas some of the targeted online sources simply store this information in two separate fields, NCO provides for their

[18] http://sioc-project.org/

breakdown into 8 (e.g. street, city) and 5 (e.g. first name, title) sub-fields respectively[19]. However, a third question that we ask already in this paper is: *iii) does the automatic breakdown of personal attributes, such as name and address, into their sub-fields have the potential to improve the recognition of semantically-equivalent contacts?*

To answer the above three questions in a reliable manner, we identified a number of real datasets on which to conduct our experiments. Most of the datasets were also used for evaluating other string metrics in earlier work [8], [4]. The number of matching value pairs identified by each of the nine datasets and in consideration for our experiments is shown in Table 2. Each dataset correspond to the different profile attributes being considered by our matching technique. Datasets 1-4[20] correspond to place names (parks), organisation names (business), person names (ucd-people) and composite entity details (restaurant - composite), respectively. In addition, the latter dataset was also available as four separate sub-datasets (4a,b,c,d)[21], broken down into address name, street, city/region, and contact number. The use of datasets 4a-d against dataset 4 in our experiments is therefore useful to answer the third question posed above. However, the contact numbers in dataset 4d were determined to be too limited in format (xxx-xxx-xxxx or xxx/xxx-xxxx) for the other experiments. Thus, we instructed three volunteers to compile an additional dataset (5) of numbers from various international online directories. They then randomly created additional types of ‘standard’[22] variations (with and without country codes) for each.

5.1 Method

To answer the first two questions, we employed two string matching libraries, SimMetrics[23] (SM) and SecondString[24] (SS), to match all pairs of records in datasets 1-5 and obtain the maximum similarity values for each. The experiment was carried out by applying the Monge-Elkan recursive metric in combination with the following 6 tokenisers, the first five of which are provided by SM[25] and the last one by SS:

1. SM Whitespace (W) - simple whitespace tokeniser
2. SM Q-Gram3 (QG3) - 3 letters Q-Gram string tokeniser
3. SM Q-Gram3 Extended (QG3E) - 3 letters Q-Gram string tokeniser, extends beyond input using padding characters
4. SM Q-Gram2 (QG2) - 2 letters Q-Gram string tokeniser

[19] Although it will be dealt with in future work, an evaluation of how reliable the information extraction technique is, is out of the scope for this paper

[20] Obtained from Cohen’s personal web page:
`http://www.cs.cmu.edu/wcohen/~match.tar.gz`

[21] Obtained from the ANU Data Mining Group:
`http://sourceforge.net/projects/febrl/`

[22] Following online conventions such as:
`http://stdcxx.apache.org/doc/stdlibug/26-1.html`

[23] `http://sourceforge.net/projects/simmetrics/`

[24] `http://secondstring.sourceforge.net/`

[25] A sixth SM tokeniser, SM CSV Basic (CSVB) - simple CSV tokeniser, was excluded from the experiments since it halts whenever a comma is encountered

5. SM Q-Gram2 Extended (QG2E) - 2 letters Q-Gram string tokeniser, extends beyond input using padding characters
6. SS Default Simple - Alphanumeric sequence/punctuation string tokeniser

In addition, 4 edit-distance metrics were considered as secondary distance functions for the Monge-Elkan metric. Besides the 2 edit-distance metrics mentioned earlier, i.e., Jaro (J) and Jaro-Winkler (J-W); we also applied the Levenshtein (L) and Smith-Waterman-Gotoh metrics (S-W-G) [13]. Thus, this resulted into 24 different alternatives of the required string matching technique (6 tokenisers with 4 Monge-Elkan variants each). To determine the best technique for each dataset, we first identified the tokeniser that produced the best results for that dataset, i.e., the maximum number of matches. For each of the best-performing tokenisers, we then also checked which Monge-Elkan variant (i.e. which secondary distance function) produced the best results.

Although each dataset referred to a number of pairs (Table 2), an observation of the paired values revealed a percentage of mismatches for most of the datasets. This ranged from 0% for the ucd-people dataset (3) to 34% for the parks dataset (1). Therefore, it was decided that to arrive to a sound conclusion in the above-described experiment it was not sufficient to consider the automatically-obtained maximum similarity values. To address this problem, we decided to calculate the F-measure for each dataset, whereby two humans were instructed to manually determine whether the paired values for each dataset should constitute a true/false positive/negative. We chose the F2-measure, which weighs recall higher than precision, because for privacy-related issues, we are more concerned with missing potential person matches than presenting a higher number of (potentially-wrong) matches. For further optimising the results, we also decided to vary the F2-measure thresholds, since if the threshold is set at 1, only perfect matches would be considered, thus also possibly missing potential matches. The threshold was varied between 0.7 and 1.0 inclusive, with increasing intervals of 0.05, resulting in 7 different F2-measures for each of the 6 tokenisers per dataset. The results of this elaborate exercise, i.e., which technique and Monge-Elkan variant produced the best results for each dataset, and at which F2-measure threshold, is discussed in Section 5.2.

In addressing the third question, we wanted to find out if correct pairs from dataset 4 (composite entities) had a better chance at being matched as a whole, as opposed to being matched by comparing each of their paired attributes separately, as available in datasets 4a-d. In addition, given that the results of the first batch of experiments (described above) indicate the best matching technique for each attribute type, we investigate whether this can also increase the chances of identifying semantic equivalence between two entities. This resulted in the following three related experiments:

1. Determining the similarity score for the composite entities in dataset 4, as a whole string, using the optimal string matching technique(s) for that dataset, as indicated by the previous experiments (Experiment A).
2. Determining the similarity score for the entities in dataset 4 based on the average similarity score for each of the 4 corresponding attributes in datasets 4a-d, using a fixed string matching technique for all datasets. Since the most-suitable technique varied for each dataset, all the techniques were considered in order to find one that is most suitable for them collectively (Experiment B-1).

3. This experiment is a variation of the previous, whereby the optimal string matching technique(s) was applied for each of the four datasets, as indicated by the previous experiments (Experiment B-2).

5.2 Results and Discussion

All results[26] of the experiments that were outlined in the previous section are discussed within this section. The results of varying the threshold for each of the 6 tokenisers executing on each of the 9 datasets are shown in Table 2. A value in the table indicates the highest F2-measure obtained, and at which highest threshold (in brackets), for each of the tokenisers executing on all datasets. For example the SM-QG3E and SS tokenisers were the best performers for dataset 1. In addition, they both performed best when, automatic pair-matching scores of 0.85 (threshold) and over were considered as a match, when calculating the resulting F2-measure (0.95). As the results suggests, there was no particular tokeniser which performed best overall, although the SM-G3E tokeniser obtained a shared highest score in all datasets but one. For that particular dataset in question (business-2), SS was the only tokeniser which yielded an acceptable score (0.92). This can be attributed to an observation we made during the experiments, where it was noted that all SM tokenisers are very case-sensitive. For example even though the following two business name strings “Avnet, Inc” and “AVNET INC” are very similar, they failed to produce high similarity results. Such a result was reflected in other cases of the same dataset where both strings were either very similar or the same.

Table 2. Results of best F2-Measure thresholds

	Dataset	#Pairs	SM-W	SM-QG3	SM-QG3E	SM-QG2	SM-QG2E	SS
1.	parks	750	0.93 (0.7)	0.94(0.85)	0.95 (0.85)	0.94 (0.85)	0.94 (0.8)	0.95 (0.85)
2.	business	795	0.08 (0.7)	0.01 (0.7)	0.02 (0.7)	0.01 (0.7)	0.01 (0.7)	0.92 (0.75)
3.	ucd-people	45	0.71 (0.7)	0.81 (0.7)	0.83 (0.7)	0.83 (0.7)	0.83 (0.7)	0.75 (0.7)
4.	restaurants - composite	212	1 (0.7)	1 (0.8)	1 (0.8)	1 (0.8)	1 (0.8)	1 (0.75)
4a.	restaurants - name	112	0.97 (0.7)	0.99 (0.7)	0.99 (0.7)	0.99 (0.7)	0.99 (0.7)	0.98 (0.7)
4b.	restaurants - street	112	0.94 (0.7)	0.97 (0.7)	0.98 (0.7)	0.97 (0.7)	0.97 (0.7)	0.93 (0.7)
4c.	restaurants - city	112	1 (1.0)	1 (1.0)	1 (0.95)	1 (1.0)	1 (0.95)	1 (1.0)
4d.	restaurants - phone	112	1 (0.85)	1 (0.85)	1 (0.9)	1 (0.85)	1 (0.85)	1 (1.0)
5.	phone numbers	150	0.99 (0.7)	1 (0.75)	1 (0.8)	1 (0.75)	1 (0.75)	0.95 (0.75)

For the next stage of the experiments, the top-performing tokenisers for each dataset (highlighted in grey) were then executed with each of the 4 edit-distance metrics. The metric/s producing the highest F2-measure for the indicated threshold, were identified as the optimal combination(s) of tokeniser-secondary distance function for each dataset, as shown in Table 3. If the same highest F2-measure at the optimal threshold was obtained for multiple tokenisers, the best variant/s were chosen by determining the one/s that obtained the highest F2-measure overall. For example, three tokenisers produced the highest F2-measure for dataset 4a (SM-QG3E, SM-QG2 and SM-QG2E), when executed with two of the secondary distance functions (J and J-W). The SM-QG3E [J-W]

[26] All results in detail can be found on:
`http://smile.deri.ie/syntactic-matching-experiments`

technique performed best on five occasions (datasets 3, 4, 4a, 4b, 5), whilst the SM-QG2 [J-W] technique performed second best on four occasions (datasets 3, 4, 4a, 4c). The Jaro-Winkler metric achieved the best results when used as the secondary distance function of the Monge-Elkan recursive metric for the majority of the datasets as can be seen in Table 3. This result is supported in [8], and similarly in [23]. A limitation observed in relation to the SS tokeniser was attributed to the Levenshtein metric when used as a secondary distance function, which always generated results of zero or less, resulting in the variant not producing the expected results.

Table 3. Datasets used and the results based on F2-Measure

	Dataset	**Library (Tokeniser) - Secondary Distance Function**
1.	parks	SM-QG3E [J]
2.	business	SS [S-W-G]
3.	ucd-people	SM-QG3E, SM-QG2, SM-QG2E [J-W]
4.	restaurants - composite	SM-QG3, SM-QG3E [J-W] ; SM-QG2, SM-QG2E [J, J-W]
4a.	restaurants - name	SM-QG3E, SM-QG2, SM-QG2E [J, J-W]
4b.	restaurants - street	SM-QG3E [J-W]
4c.	restaurants - city	SM-W, SM-QG3, SM-QG2 [S-W-G, J, J-W, L] ; SS [S-W-G, J, J-W]
4d.	restaurants - phone	SS [S-W-G, J, J-W]
5.	phone numbers	SM-QG3E [J, J-W]

The results of the final experiment, which questioned whether splitting up an entity's details into attributes (Experiments B-1,2) will produce better similarity results than comparing the details as a whole string (Experiment A) are shown in Table 4. The results indicate the average results of the similarity values for all restaurant pairs. Since as shown in Table 3 more than one technique produced an optimal F2-measure for some of the datasets, when selecting a separate technique for these datasets in experiment B2, we selected the technique which yielded the best results overall. This selection was based on the consistency of the most popular tokeniser and secondary distance function. The best tokenisers overall were SM-QG3E (6 times) and SM-QG2 (4 times), whilst the best secondary functions were Jaro-Winkler (9 times) as highlighted earlier, and Jaro (7 times). As a result, the techniques applied to each of the four attributes in experiment B2 are as follows: names and streets, SM-QG3E [J-W]; cities, SM-QG2 [J-W]; and phone numbers, SS [J-W].

Table 4. Similarity Matching Results

Experiment	**A**	**B-1**	**B-2**
Similarity Value	0.95	0.96	***0.97***

Experiment B-1,2 produced marginally better results than experiment A, indicating that the breakdown of entity details into attributes, does increase the likelihood of determining a syntactic match. Furthermore, the result of Experiment B-2 justify our attempt to find an optimal technique for each attribute type. However, the improvement in both cases is marginal. This can be attributed to the following observation, whereby it was noted that some attributes (e.g. address) have variations that will cause a higher impact to the similarity score when split, as opposed to when they are not. For example, if two full addresses (string) are exactly the same with the exception of the sub-string denoting the city, the similarity score will not be highly impacted. If the address sub-strings are matched separately, a different city will produce a very low score, resulting in a higher impact on the average score for the address. The best result for experiment B-1 was produced by the SS [S-W-G] technique, whilst the SM-QG2 [J-W] technique produced the best result for experiment A.

6 Future Work

The current prototype is able to retrieve a user's profile data from LinkedIn, Facebook and Twitter, but more online accounts are being targeted. The envisaged semantic extension to the current syntactic-based profile attribute matching technique is our most challenging future enhancement. Research contributions will on the other hand focus on defining an appropriate semantic-based attribute weighting for each matched attribute, together with the definition of a metric which takes into account all the resulting weighted matches and the identification of a threshold that determines whether two or more online profile refer to the same person. Online posts are also taken into consideration [28]. An analysis of posts from multiple accounts can help us discover whether two or more online profiles are semantically equivalent.

Finally, two comprehensive evaluations will be performed. The first will compare the success of a syntactic-only technique versus a hybrid syntactic-semantic technique. Both will be performed on a dataset of user profiles in order to find out if the envisaged hybrid syntactic/semantic-based matching technique actually yields better results. This evaluation will also determine which attribute scoring approach fares best for the second evaluation. The second evaluation involves a complete user evaluation of the system on at least 15-20 participants. Each participant will be asked to integrate any two of their LinkedIn, Facebook and Twitter profiles, where a list of possible duplicate contacts will be presented to the participant, in order for them to determine if the recommendations actually refer to the same person or not.

7 Conclusions

In this paper, we discuss the possibility of eliminating the need for the user to separately manage multiple online identities in unrelated online accounts. Our approach targets the extraction and retrieval of user profile data from these accounts, and their mapping onto our comprehensive ontology framework, which serves as a standard format for the representation of profile data originating from heterogeneous distributed sources. The aggregated profile data is integrated into a unique PIM representation which then serves as the user's super profile. The main objective discussed in this paper is the discovery of semantic equivalence between online identities as apparent in online profiles. For the purpose, we present a weighted similarity technique that compares newly-retrieved profiles with known identities, at both syntactic and semantic levels. In contrast to other approaches, we consider all profile attributes, and not just a handful. In order to demonstrate the added value of performing a matching technique based on such 'fine-grained' entities, we conduct an experiment to determine whether it is more likely for two identical entities (having similar 'identity' attributes to people profiles) to yield a syntactic 'match' when considering i) their information as a whole, or ii) their attributes separately. The latter approach has the advantage of optimising different existing techniques for different attributes. In fact, the results confirm that the decomposition of entities into attributes is likely to produce better results when comparing them syntactically.

Acknowledgements. This work is supported in part by the European Commission under the Seventh Framework Program FP7/2007-2013 (*digital.me* – ICT-257787) and in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (*Líon-2*).

References

1. Akhtar, W., Kopecký, J., Krennwallner, T., Polleres, A.: XSPARQL: Traveling between the XML and RDF worlds – and avoiding the XSLT pilgrimage. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 432–447. Springer, Heidelberg (2008)
2. Appelquist, D., Brickley, D., Carvahlo, M., Iannella, R., Passant, A., Perey, C., Story, H.: A standards-based, open and privacy-aware social web. W3c incubator group report, W3C (December 2010)
3. Aumueller, D., Do, H., Massmann, S., Rahm, E.: Schema and ontology matching with coma++. In: Proc. ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp. 906–908 (2005)
4. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive name matching in information integration. IEEE Intelligent Systems 18(5), 16–23 (2003)
5. Bortoli, S., Stoermer, H., Bouquet, P., Wache, H.: Foaf-o-matic - solving the identity problem in the foaf network. In: Proc. Fourth Italian Semantic Web Workshop, SWAP 2007 (2007)
6. Bourimi, M., Scerri, S., Cortis, K., Rivera, I., Heupel, M., Thiel, S.: Integrating multi-source user data to enhance privacy in social interaction. In: Proceedings of the 13th International Conference on Interacción Persona-Ordenador, INTERACCION 2012, pp. 51:1–51:7. ACM, New York (2012)
7. Budanitsky, A., Hirst, G.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: Proc. Workshop on Wordnet and other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (2001)
8. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string metrics for matching names and records. In: Proceedings of the KDD 2003 Workshop on Data, Washington, DC, pp. 13–18 (2003)
9. Cross, V.: Fuzzy semantic distance measures between ontological concepts. In: Proc. IEEE Annual Meeting of the Fuzzy Information Processing Society, NAFIPS 2004, vol. 2, pp. 635–640 (June 2004)
10. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, ACL 2002 (2002)
11. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proc. Twentieth International Joint Conference for Artificial Intelligence, IJCAI 2007, Hyderabad, India, January 6-12, pp. 1606–1611 (2007)
12. Golbeck, J., Rothstein, M.: Linking social networks on the web with foaf: A semantic web case study. In: Proc. Twenty-Third Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, pp. 1138–1143 (2008)
13. Gotoh, O.: An improved algorithm for matching biological sequences. Journal of Molecular Biology 162, 705–708 (1981)

14. Ion, M., Telesca, L., Botto, F., Koshutanski, H.: An open distributed identity and trust management approach for digital community ecosystems. In: Proc. International Workshop on ICT for Business Clusters in Emerging Markets. Michigan State University (June 2007)
15. Jaro, M.A.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association 84(406), 414–420 (1989)
16. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: Proc. K-Cap 2005 Workshop on Integrating Ontologies, pp. 87–93 (2005)
17. Labitzke, S., Taranu, I., Hartenstein, H.: What your friends tell others about you: Low cost linkability of social network profiles. In: Proc. 5th International ACM Workshop on Social Network Mining and Analysis, San Diego, CA, USA, August 20 (2001)
18. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10, 707 (1966)
19. Li, Y., Bandar, Z.A., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering 15, 871–882 (2003)
20. Mika, P.: Flink: Semantic web technology for the extraction and analysis of social networks. Journal of Web Semantics 3(2-3), 211–223 (2005)
21. Monge, A., Elkan, C.: The field matching problem: Algorithms and applications. In: Proc. Second International Conference on Knowledge Discovery and Data Mining, pp. 267–270 (1996)
22. Mylka, A., Sauermann, L., Sintek, M., van Elst, L.: Nepomuk contact ontology. Technical report (2007)
23. Piskorski, J., Sydow, M.: Usability of string distance metrics for name matching tasks in polish. In: Proceedings of the 3rd Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, LTC 2007 (2007)
24. Raad, E., Chbeir, R., Dipanda, A.: User profile matching in social networks. In: Proc. 13th International Conference on Network-Based Information Systems, Takayama, Gifu, Japan, pp. 297–304 (2010)
25. Ray, S.R.: Interoperability standards in the semantic web. Journal of Computing and Information Science in Engineering, ASME 2, 65–69 (2002)
26. Rowe, M., Ciravegna, F.: Getting to me: Exporting semantic social network from facebook. In: Proc. Social Data on the Web Workshop, International Semantic Web Conference (2008)
27. Sauermann, L., van Elst, L., Möller, K.: Personal information model (pimo). Oscaf recommendation, OSCAF (February 2009)
28. Scerri, S., Cortis, K., Rivera, I., Handschuh, S.: Knowledge discovery in distributed social web sharing activities. In: Making Sense of Microposts (#MSM 2012), pp. 26–33 (2012)
29. Scerri, S., Gimenez, R., Herman, F., Bourimi, M., Thiel, S.: Digital.me - towards an integrated personal information sphere. In: Proc. Federated Social Web Europe Conference, FSW 2011, Berlin, Germany (2011)
30. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 146–171. Springer, Heidelberg (2005)
31. Takale, S.A., Nandgaonkar, S.S.: Measuring semantic similarity between words using web documents. International Journal of Advanced Computer Science and Applications (IJACSA) 1(4) (2010)
32. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proc. The Section on Survey Research, pp. 354–359 (1990)
33. Yang, H., Callan, J.: Learning the distance metric in a personal ontology. In: Proc. 2nd International Workshop on Ontologies and Information Systems for the Semantic Web, New York, NY, USA, pp. 17–24 (2008)

Modeling Snapshot of Composite WS Execution by Colored Petri Nets

Yudith Cardinale[1], Marta Rukoz[2], and Rafael Angarita[1]

[1] Departamento de Computación, Universidad Simón Bolívar,
Caracas, Venezuela 1080
[2] LAMSADE, Université Paris Dauphine
Université Paris Ouest Nanterre La Défense
Paris, France
{yudith,rangarita}@ldc.usb.ve, marta.rukoz@lamsade.dauphine.fr

Abstract. The global transactional property of a Transactional Composite Web Service (TCWS) allows recovery processes if a Web Service (WS) fails during the execution process. The following actions can be performed if a WS fails: retry the faulty WS, substitute the faulty WS, or compensate the executed WSs. In consequence, these fault-tolerance mechanisms ensure the atomicity property of a TCWS with an all-or-nothing endeavor. In this paper, we present a formal definition of a checkpointing approach based in Colored Petri-Nets (CPNs) properties, in which the execution process and the actions performed in case of failures rely on unrolling processes of CPNs. Our checkpointing approach allows to relax the atomic transactional property of a TCWS in case of failures. The all-or-nothing transactional property becomes to the *something-to-all* property. A snapshot of the most possible advanced partial result is taken in case of failures and it is returned to the user (user gets *something*), providing the possibility of restarting the TCWS from an advanced execution state to complete the result (user gets *all* later), without affecting its original transactional property. We present the execution algorithms with the additionally capacity of taking snapshot in case of failures and experimental results to show the reception of partial outputs due to the relaxation of the *all-or-nothing* property.

1 Introduction

Web Service (WS) technology has gained popularity in both research and commercial sectors, based on the Semantic Web approach which makes part of the Web 3.0. With machine intelligence, users can resolve complex problems that require the interaction among different tasks. One of the major goals of the Web 3.0 is to support automatic and transparent WS composition and execution allowing a complex user request to be satisfied by a Composite Web Service (CWS), in which several WSs work together to resolve the complex query [2].

Automatic selection, composition, and execution of CWSs are issues that have been extensively treated in the literature by guaranteeing functional requirements (i.e., the set of input attributes provided in the query and the attributes that will be returned as output) and *QoS* criteria (e.g., response time

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 23–44, 2013.

and price) [3–5, 11–13, 16, 20]. In some cases, transactional properties (e.g., atomic, compensable or not) are also considered to ensure fault-tolerant execution with Transactional Composite Web Services (TCWSs) [6, 7, 10, 14]. In a TCWS all its component WSs have transactional properties, which in turn could define an aggregated transactional property.

Because WSs can be created and updated on-the-fly, the execution system needs to dynamically detect changes during run-time, and adapt execution to the availability of the existing WSs. In this sense, TCWS becomes a key mechanism to cope with challenges of open software in dynamic changing environments to ensure that the whole system remains in a consistent state even in presence of failures [23].

Even if all component WSs of a Composite WS are transactional, the composition itself could be not transactional (e.g., a WS with an atomic but non-compensable transactional property, cannot be followed by another WS whose transactional property does not ensure a successful execution; if the second one fails, the first one cannot be compensated). Thus, to ensure the transactional property of a TCWS, the WSs selection process is made according to their transactional properties and their execution order. In this context, failures during the execution of a TCWS can be repaired by backward or forward recovery processes. Backward recovery implies to undo the work done until the failure and go back to the initial consistent state (before the execution started), by rollback and compensation techniques. Forward recovery tries to repair the failure and continues the execution; retry and substitution are some techniques used.

In both backward and forward recovery processes, the atomic (all-or-nothing) transactional property is comply to ensure system consistency. However, backward recovery means that users do not get the desired answer to their queries and forward recovery could imply long waiting time, because of the invested time to repair failures, for users to finally get the response. For some queries, partial responses may have sense for users; thus, they need alternative recovery strategies that provide this facility in case of failures.

In previous works, we provided the definition of backward recovery (compensation process) and forward recovery (retry and substitution) approaches [8, 9] based on Colored Petri Nets (CPNs) formalism. In CPNs, transitions represent WSs, places represent input/output WS attributes, and colors are used to represent transactional properties of transitions and types of values in places. In [8] unrolling algorithms of CPNs to control the execution and backward recovery were presented. This work was extended in [9] to consider forward recovery based on WS replacement; formal definitions for WSs substitution process, in case of failures, were presented. In [9], we also proposed an Executor architecture, independent of its implementation, to execute a TCWS following our proposed fault-tolerant execution approach. In [1, 18], we have presented implementations of our fault-tolerant approaches. In [1], we present FaCETa, a framework which implements the backward and forward recovery proposed in [8, 9]. In [18], we present the framework FaCETa*, an extension of FaCETa, in which the fault-tolerant approach is extended with checkpoints, i.e., in case of failures, the execution state of a TCWS is checkpointed and the execution flow goes on as much as it is possible, therefore users can have partial responses and later restart the execution of the TCWS.

The contribution of this paper is focused in formally defining the checkpointing approach, based also in CPN properties, as a way to relax the atomic transactional property (all-or-nothing) of a TCWS in case of failures. It means that, instead of all-or-nothing, users can have a *something-to-all.* If a failure occurs, a snapshot that contains the execution state of the most possible advanced partial result is taken and it is returned to the user (user gets *something*). The checkpointed TCWS can be re-started from an advanced point of execution (snapshot) to complete the desired result (user gets *all* later), without affecting its aggregated transactional property. We also present the execution algorithms with the additionally capacity of taking snapshot in case of failures, the extended framework incorporating checkpointing facilities, and experimental results to show the results of the prototype implementation of the checkpointing mechanism.

This paper is organized as follows. Section 2 recalls some important concepts and formal definitions necessary for the understanding of this work, such as Web Service composition and their properties and execution. Section 3 introduces an alternative fault-tolerance approach as a way to relax the *all-or-nothing* transactional property to a *something-to-all* property. Section 4 presents the formal definitions to allow the execution of the checkpointing mechanism in case of failure. Section 5 presents the overall architecture of our extended framework and some results showing the reception of partial results in case of failure. Section 6 discusses related work in the field of checkpointing for TCWSs. Finally, Section 7 presents our conclusions.

2 Preliminaries

This Section recalls some important concepts and formal definitions about Web Service composition and their properties and execution.

2.1 Web Service

A Web Service, ws, consists of a finite set of operations, denoted as $ws = \{op_i, i = 1..n\}$, with $op_i = (I_i, O_i, Q_i, T_i)$, where $I_i = \{I_{i1}, I_{i2}, ..\}$ is a set of input attributes of op_i, $O_i = \{O_{i1}, O_{i2}, ..\}$ is a set of output attributes whose values are produced by op_i, $Q_i = \{Q_{i1}, Q_{i2}, ...\}$ is a set of QoS values of op_i for a set of QoS criteria $\{q_1, q_2, ...\}$, and $T_i \in \{p, a, c, cr, pr, ar\}$ is the transactional property of op_i (transactional properties are defined in Section 2.3). In this work, without loss of generality, we consider that ws has only one op_i and we use the term ws to denote the op of ws.

2.2 Composite Web Service

A Composite Web Service, described as $CWS = \{ws_i, i = 1..m\}$, is a combination of several WSs to produce more complex services that satisfy more complex user requests. It concerns *which* and *how* WSs are combined to obtain the desired results. A CWS can be represented in structures such as workflows, graphs, or Petri Nets indicating, for example, the control flow, data flow, WSs execution order, and/or WSs behavior. The structure representing a CWS can be manually or automatically generated. Users can manually specify *how* functionality of WSs are combined or a Composer can automatically decide *which* and *how* WSs are combined, according the desired query. In both cases, the execution of a CWS is carried out by an Executor, that decides *which* WSs comply each functionality manually specified by users and invokes them, or invokes the WSs automatically decided by the Composer. In this paper, we represent a CWS by a colored Petri Net as it is established in Definition 1 and suppose that it was generated automatically by a Composer [6].

2.3 Transactional Properties

The transactional property (TP) of a WS allows to recover the system in case of failures during the execution. A single WS is transactional (denoted as TWS), if when it fails, it has no effect at all. The most basic transactional property that implements this characteristic is **pivot (p)**: A TWS is called **pivot (p)** WS, if its effects remain forever and cannot be semantically undone once it has completed successfully. For a CWS, it is transactional, named Transactional Composite WS (TCWS), if when it fails, its partial effects can be semantically undone. In this case, the basic transactional property is called **atomic (a)**: a TCWS is **atomic (a)**, if the effects of the TCWS remain forever and cannot be semantically undone once it has completed successfully. There exist other transactional properties for TWS and TCWS, which complement transactionality. A TCWS or TWS can be associated with another TCWS or TWS which can semantically undo its successfully execution; in this case, the TCWS or TWS is called **compensatable** (c). A TCWS or TWS can be combined with a retriable property, which guarantees a successfully termination after a finite number of invocations. In this case, we obtain **pivot retriable** (pr), **atomic retriable** (ar), and **compensatable retriable** (cr) WSs. WSs that provide transactional properties are useful to guarantee reliable TCWSs execution and to ensure the whole system consistent state even in presence of failures. Failures during the execution of a TCWS can be supported according to the TP of its component WSs by a forward recovery process, in which the failure is repaired to allow the failed WS to continue its execution or by a backward recovery process, wherein its partial effects are semantically undone.

2.4 User Query

We define a query in terms of functional conditions, expressed as input and output attributes; QoS constraints, expressed as weights over criteria; and the required global TP as follows. A query Q is a 4-tuple $Q = (I_Q, O_Q, W_Q, T_Q)$, where:

- I_Q is a set of input attributes whose values are provided by the user,

- O_Q is a set of output attributes whose values have to be produced by the system,
- $W_Q = \{(w_i, q_i) \mid w_i \in [0,1]$ with $\sum_i w_i = 1$ and q_i is a QoS criterion$\}$, and
- T_Q is the required transactional property: $T_Q \in \{T_0, T_1\}$; If $T_Q = T_0$, the system guarantees that a semantic recovery can be done by the user. If $T_Q = T_1$, the system does not guarantee that the result can be compensated. In any case, if the execution is not successful, nothing is changed on the system and its state is consistent.

2.5 Execution Control

A TCWS, which answers and satisfies a query Q, is modeled as an acyclic marked CPN, denoted CPN-EP_Q, as following.

Definition 1 Transactional Composite Web Service. *A* TCWS *is a 4-tuple (A, S, F, ξ), where:*

- *A is a finite non-empty set of places, corresponding to input and output attributes of the* WSs*;*
- *S is a finite set of transitions corresponding to the set of* WSs $\in$ TCWS *;*
- *$F : (A \times S) \cup (S \times A) \rightarrow \{0,1\}$ is a dataflow relation indicating the presence (1) or the absence (0) of arcs between places and transitions defined as follows: $\forall s \in S$, $(\exists a \in A \mid F(a,s) = 1) \Leftrightarrow$ (a is an input place of s) and $\forall s \in S$, $(\exists a \in A \mid F(s,a) = 1) \Leftrightarrow$ (a is an output place of s);*
- *ξ is a color function such that $\xi : S \rightarrow \Sigma_S$, with $\Sigma_S = \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ representing the TP of $s \in S$.*

According to CPN notation, we have that for each $x \in (A \cup S)$, $(^\bullet x) = \{y \in A \cup S : F(y,x) = 1\}$ is the set of its predecessors, and $(x^\bullet) = \{y \in A \cup S : F(x,y) = 1\}$ is the set of its successors.

We suppose that a TCWS is well constructed, i.e., its component WSs satisfy the transactional rules presented in Table 1. Let us illustrate the rules in Table 1 with the following examples. If the TP of a ws_i is p or a, another ws_j, whose TP is pr, ar, or cr can be executed after ws_i (sequential execution, rule 1); ws_i can be executed in parallel with a ws_k with TP cr (rule 2). This rules guaranteeing that the resulting TCWS satisfies the transactional properties presented in section 2.3 [10] .

Definition 2 Marked CPN-EP_Q. *A marked CPN-EP_Q is a pair (*TCWS*,M), where* TCWS$=(A, S, F, \xi)$ *and M is a function which assigns tokens (values) to places such that $\forall a \in A$, $M(a) \in N$.*

Given a user query $Q = (I_Q, O_Q, W_Q, T_Q)$, a marked CPN-$EP_Q = ((A, S, F, \xi), M)$ satisfies Q if:

- $\forall x \in I_Q$, $\exists a \in A$ such that a is an input place.
- $\forall x \in O_Q$, $\exists a \in A$ such that a is an output place.

Table 1. Transactional rules of [10]

Transactional property of a WS	Sequential compatibility	Parallel compatibility
p, $\boldsymbol{a}$	$pr \cup \boldsymbol{ar} \cup cr$ (rule 1)	cr (rule 2)
pr, $\boldsymbol{ar}$	$pr \cup \boldsymbol{ar} \cup cr$ (rule 3)	$pr \cup \boldsymbol{ar} \cup cr$ (rule 4)
c	Σ_S (rule 5)	cr (rule 6)
cr	Σ_S (rule 7)	Σ_S (rule 8)

- let $M_Q = \{\forall a \in (A \cap I_Q),\ M(a) = 1$ and $\forall a \in (A - I_Q),\ M(a) = 0\}$ and $M_F = \{\forall a \in (A \cap O_Q),\ M(a) \geq 1\}$, the initial and final marking, respectively; there exist a firing sequence σ, such that: $M_Q \overset{\sigma}{\rightarrow} M_F$ and such that transitions of σ represent a TCWS whose components satisfy the transactional rules, locally optimize the QoS and $\forall s_i \in \sigma \mid \xi_Q(s) \in \{c, cr\}$ if $T_Q =' compensatable'$, and $\forall s_i \in \sigma \mid \xi_Q(s) \in \{pr, ar, cr\}$ if $T_Q =' retriable'$.

The marking of a CPN-EP_Q represents the current values of attributes that have been produced either for some component WSs or by the user, and that can be used for others component WSs to be invoked. A Marked CPN denotes which transitions can be fired.

In order to finally resolve the query Q, the given TCWS has to be executed by invoking its component WSs according to the execution flow depicted by the CPN representing the TCWS(i.e. CPN-EP_Q). In fact, during the execution, CPN-EP_Q represents the execution plan of TCWS. As the composition process presented in [9], the execution process is controlled by an unrolling algorithm over CPN-EP_Q.

Definition 3 Fireable Transition. *A marking M enables a transition s (to invoke the ws it represents) iff all its input places contain tokens such that* $\forall x \in ({}^\bullet s)$, $M(x) \geq card({}^\bullet x)$.

To start the execution algorithm, the CPN-EP_Q is marked with the *Initial Marking* and some transitions become fireable. When a transition is fireable, it can be fired according to the firing rules (see definition 4). The firing of a transition of a CPN-EP_Q corresponds to the execution of a WS, let us say s, which participates in the composition.When s finishes, other transitions become fireable, and so on.

Definition 4 CPN-EP_Q Firing Rules. *The firing of a fireable transition s for a marking M defines a new marking M', such that: all tokens are deleted from its input places ($\forall x \in {}^\bullet s$, $M(x) = 0$) and the* WS *s is invoked. These actions are atomically executed. After* WS *s finishes correctly, tokens are added to its output places ($\forall x \in (s^\bullet)$, $M(x) = M(x) + 1$).*

Note that during the execution, in CPN-EP_Q a transition is fireable (its corresponding WS can be invoked) only if all its predecessor transitions have been fired (each input place has as many tokens as WSs produce them or one token if the user provides them) and several transitions can be fireable at the same time. In this way, the execution control, followed according CPN-EP_Q, respect sequential and parallel executions, which in turn keeps the global transactional property.

Once a WS is executed, its input places are unmarked and its output places (if any) are marked. We illustrate these definitions with the example shown in Figure 1, where $Q = (I_Q, O_Q, W_Q, T_Q)$ with $I_Q = \{a_1, a_2\}$ and $O_Q = \{a_5, a_6, a_7\}$. Note that ws_3 needs two tokens in a_3 to be invoked; this data flow dependency indicates that it has to be executed in sequential order with ws_1 and ws_2, and can be executed in parallel with ws_4. Note that a_3 is produced by ws_1 and ws_2, ws_1 was already executed and it produced a token on a_3, and ws_2 is still running. Even if ws_3 could be invoked with the values produced by ws_1, if ws_3 is fired, it will be executed in parallel with ws_2; however, it could be possible that transactional properties of ws_2 and ws_3 dictate that they have to be executed in sequential order as the data flow indicates. Then, ws_3 has to wait for all its predecessors transitions to finish in order to be invoked. Once ws_2 finishes, ws_3 and ws_4 can be executed in parallel.

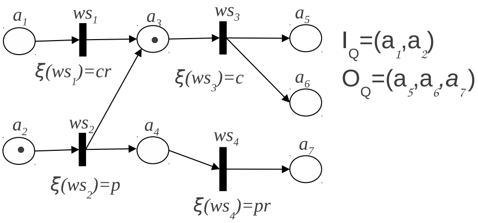


Fig. 1. Example of Fireable Transitions

2.6 Fault Tolerant Execution Control

The global TP of a TCWS allows recovery processes if a WS s fails during the execution process. In previous works, we have presented a recovery mechanism [9] based on TP properties of its component WSs. In these works, if a WS s fails, the following actions are executed:

- if $TP(s)$ is **retriable** (pr, $\boldsymbol{ar}$, cr), s is re-invoked until it successfully finishes (forward recovery);
- otherwise, another Transactional substitute WS, s^*, is selected to replace s and the unrolling algorithm goes on (trying a forward recovery);
- if there not exists any substitute s^*, a backward recovery is needed, i.e., all executed WSs must be compensated in the inverse order they were executed; for parallel executed WSs, the order does not matter. The compensation flow is represented by a backward recovery CPN (BRCPN-$TCWS_Q$),

which depicts the inverse order of the execution flow. The corresponding BRCPN-$TCWS_Q$ for a TCWS can be automatically generated by the same COMPOSER that built the TCWS.

These actions guarantee the atomicity (all-or-nothing) property of a TCWS. In the next section we present a checkpointing mechanism allowing to relax this property by returning *something* in case of failures. Then, a re-execution from an advanced execution state is possible to finally get the final desired response (user finally gets *all*).

3 Relaxing All-or-Nothing Transactional Properties by Checkpointing Mechanisms

Approaches previously presented in [8, 9] provide fault-tolerant mechanisms relying on WSs replacement, on a compensation protocol, and on unrolling processes of CPNs. Although these recovery processes ensure system consistency, they represent an "all-or-nothing" approach, since users either receive full answer to their queries or they do not get any answer (in case of failure, partial answers, if any, are undone). In this section we present an alternative fault-tolerant approach as a way to relax this transactional property to a *something-to-all* property. In case of failures, the unrolling process of the CPN controlling the execution of a TCWS is checkpointed and the execution flow continues as much as possible. In consequence, users can receive partial responses (*something*) as soon as they are produced and resubmit the checkpointed CPN to restart its execution from an advanced point of execution and finish the TCWS (to get *all*), without affecting the original transactional property.

For this purpose, when a WS associated to a fireable transition t fails, the execution control, instead of executing backward recovery, it saves the subnet of CPN-EP_Q that could not be executed. For that, the inputs and output places, and valid attributes (attributes already produced) of transition t are saved, and the same attributes are saved recursively for any other transition that depends on t.

The checkpointing mechanism is illustrated in the following. The marked CPN-EP_Q depicted in Figure 2 is the state when ws_4 fails and the unrolling of the CPN-EP_Q continues to allow the execution of all the WSs not affected by the failure of ws_4. The only WSs affected by this failure are ws_7 and ws_8; therefore, assuming that there will not be more failures, the output attribute corresponding to a_{10} will be obtained.

Figure 3 shows the execution state when all the WSs not affected by the failure were executed and the a_{10} value was received. Red places and transitions in Figure 3 represent the part of the marked CPN-EP_Q involved in the execution restart process, called CPN-$check_{Q'}$, (the associated WSs of these transitions were not executed because they need values produced by the failed WS or any other that depend of it). The red tokens in Figure 3 represent the values already produced during the normal execution (these tokens will be the initial marking of the CPN-$check_{Q'}$).

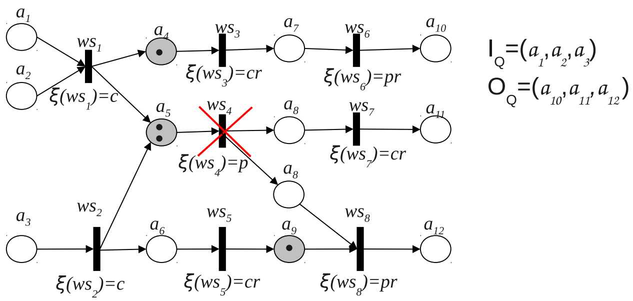


Fig. 2. Marked CPN-TCWS$_Q$ when ws_4 fails

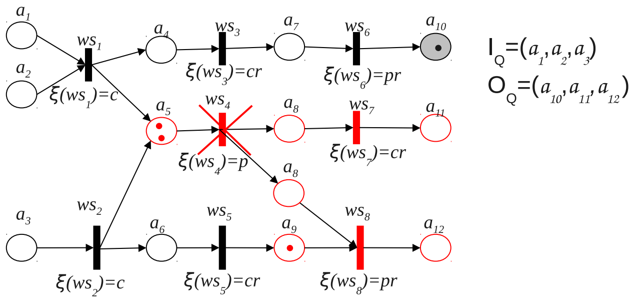


Fig. 3. Marked CPN-TCWS$_Q$ just before checkpointing

Figure 4 represents the CPN-$check_{Q'}$. Note that the initial marking contains tokens representing values produced during the CPN-EP_Q execution, whilst a_{11} and a_{12} are the only output attributes expected as a result of the CPN-$check_{Q'}$ execution. Note that ws_8 has to wait only for one of its two inputs, since it had already received a_9 before the execution was restarted.

4 Modeling Checkpointing Based on Petri-Net Formalism

We extend the CPN-EP_Q unrolling execution process to take into account checkpoints in case of failure by modifying definitions 1, 2, 3, and 4; additionally, new definitions regarding to the checkpointing mechanism are presented. In a general way, these definitions express the idea presented in the following paragraph.

If a WS fails, its corresponding transition informs its successors about the failure, thus they know they are not going to receive the values corresponding to the outputs of the failed WS. If one transition is notified that one or more of its predecessor transitions will not be able to produce its output values, it still waits until all its predecessor transitions have finished. Therefore, it is possible for transitions to receive their required input values partially. Then, it informs its

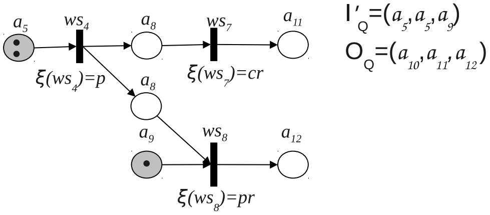


Fig. 4. Marked CPN-$check_{Q'}$

successor transitions about its inability to invoke its corresponding WS. When the unrolling process has finished, one or more O_Q attributes will have faulty values, whilst the rest will have correct values. At this point, a snapshot representing the non-successfully executed part of the CPN-EP_Q is saved. User is provided with the possibility to restart the execution later, executing only the previously failed WSs and the WSs that were never fired for execution. Note that it is possible that none of the O_Q attributes has a correct value, however at least the snapshot represents an advanced execution state.

Using CPNs, information can be modeled by tokens and the type of information can be modeled by the color of those tokens. We define the following colors associated to places in order to model the unrolling process for checkpointing.

- Valid (v): if a token belonging to a place has color v, it means that the WS that produced its value was executed successfully.
- Invalid (i): if a token belonging to a place has color i, it means that the WS that produces its value was not executed successfully; i.e., the WS supposed to produce its value failed or it was not executed because one of its WSs predecessors failed.

The following definitions allow the execution of the checkpointing mechanism in case of failure.

Definition 5 Transactional Composite Web Service. *A* TCWS *is a 4-tuple* (A, S, F, ξ)*, where:*

- *A is a finite non-empty set of places, corresponding to input and output attributes of the* WSs*;*
- *S is a finite set of transitions corresponding to the set of WSs* $\in$ TCWS *;*
- $\forall s \in S$, *(*$\exists a \in A \mid F(a, s) = 1$*)* $\Leftrightarrow$ *(a is an input place of s) and* $\forall s \in S$, *(*$\exists a \in A \mid F(s, a) = 1$*)* $\Leftrightarrow$ *(a is an output place of s);*
- *ξ is a color function such that ξ: $C_A \cup C_S$ with C_A: $A \rightarrow \sum_A$, a color function such that $\sum_A = \{v, i\}$ representing, for $a \in A$, either the success or failure of its predecessor transitions, and $C_S : S \rightarrow \sum_S$, a color function such that $\sum_S = \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ represents the TP of $s \in S$ $(TP(s))$.*

Definition 6 Marked Executable CPN-EP_Q. *A marked CPN-EP_Q=(A, S, F, ξ) is a pair (CPN-EP_Q,M), where M is a function which assigns tokens (values) to places such that $\forall a \in A,\ M(a) \subseteq \{\emptyset, Bag(\sum_A)\}$, where Bag corresponds to a set which can contain several occurrences of the same element. The marking of a CPN represents the current state of the system, i.e., the set of attributes produced correctly by the system and/or signals indicating failures.*

A transition s is fireable when all its predecessor transitions have added a token to their output places (input places of s). If all of them are valid tokens we said that s is fireable for execution, otherwise (i.e., at least one of them is an invalid token) we said that s is fireable for checkpointing. During the unrolling process for the execution of a TCWS all the predecessor places of s will have the required tokens for the invocation of s. In case of failures, some of these tokens will be invalid, as it is shown in the following definition.

Definition 7 Fireable Transition. *A marking M enables a transition s for execution iff all its input places contain tokens such that $\forall x \in ({}^\bullet s),\ card(M(x)) \geq card({}^\bullet x)\ \wedge\ M(x) \subseteq Bag(\{v\})$. A marking M enables a transition s for checkpointing iff all its input places contain tokens such that $(\forall x \in ({}^\bullet s),\ card(M(x)) \geq card({}^\bullet x))\ \wedge\ (\exists x \in ({}^\bullet s),\ \{i\} \in M(x))$.*

Definition 8 CPN-EP_Q Firing Rules. *The firing for execution of a fireable transition s for a marking M defines a new marking M′, such that: all tokens are deleted from its input places ($\forall x \in {}^\bullet s,\ M(x) = 0$) and the* WS *s is invoked. These actions are atomically executed. After* WS *s finishes, tokens are added to its output places ($\forall x \in (s^\bullet),\ (M(x) \leftarrow M(x) \cup \{v\})$). The firing for checkpointing of a fireable transition s for a marking M, defines a new marking M′, such that: s, its inputs and outputs places, and its valid input attributes are saved; and tokens are added to its output places $(\forall x \in (s^\bullet),\ (M(x) \leftarrow M(x) \cup \{i\}))$. These actions are also atomically executed.*

Definition 9 Local Snapshot. *A Local Snapshot is the set of data representing the state of a transition in a CPN. It contains:*

- *Inputs_ws represents the information about input attributes required by s to become fireable. For each predecessor place of s it contains a set of pairs token-value, where token contains either v or i depending on whether the attribute was generated correctly or not, and value contains the actual received value iff the attribute was generated correctly (value will be empty iff the token value is i). Inputs_ws will be empty if the transition did not consume the value of any of its predecessor places, or it can contain the value consumed prior to the checkpointing. If s was executed unsuccessfully, then InputsNeeded_ws will contain all the input values required by s;*
- *$Results_{ws}$ represents the output attributes of s iff s was executed successfully.*

Definition 10 Global Snapshot. *A Global Snapshot (GS) is the set of data necessary to restart the execution of a* TCWS. *A GS contains the union of all* *local snapshots, which is the CPN-check$_{Q'}$ containing the information of the part of the* TCWS *to be restarted, the user query Q, and the I'_Q necessary to restart the execution.*

5 Framework Architecture

In [1], we presented a framework which implements the backward and forward recovery proposed in [8, 9]. In [18], we present a proposal to extend our framework for considering checkpointing mechanisms. In this section, we first present a deeper description of the overall architecture of our extended framework and a detailed explanation of the fault-tolerance algorithms incorporated to the framework. Finally, we present some results showing the reception of partial results in case of failure, which relaxes the *all-or-nothing* property to *something-or-nothing.*

During the TCWS execution there exist two basic variants of execution scenarios for component WSs. In *sequential* scenario, WSs work on the result of previous services and cannot be invoked until previous services have finished. In *parallel* scenario, several services can be invoked simultaneously because they do not have data flow dependencies. The global *TP* of TCWSs is affected by the execution scenarios. Hence, it is mandatory to follow the same CPN unrolling algorithm taken by the Composer in order to ensure that sequential and parallel execution satisfies the global *TP*.

The execution of a TCWS in our framework (referenced as Executor) is managed by an Execution Engine and a collection of software components called Engine Threads, organized in a three-level architecture. Figure 5 depicts the overall architecture of our Executor. In the first level, the Execution Engine receives the TCWS and its corresponding *BRCPN* (the compensation order), both represented by CPNs automatically generated by the Composer. It launches, in the second layer, an Engine Thread for each WS in the TCWS. Each Engine Thread is responsible for the execution control of its WS. They receive WS inputs, invoke their respective WS, and forward their results to their peers to continue the execution flow. Hence, the Execution Engine is responsible for initiating the Engine Threads and the unrolling algorithm, while Engine Threads are responsible for the actual invocation of WSs monitoring its execution, and forwarding results to its peers to continue the execution flow. In case of failure, all of them participate in the recovery process.

By distributing the responsibility of executing a TCWS across several Engine Threads, the logical model of our Executor enables distributed execution and it is independent of its implementation; i.e., this model can be implemented in a distributed memory environment supported by message passing or in a shared memory platform, e.g., supported by a distributed shared memory or tuplespace system. The idea is to place the Executor in different physical nodes (e.g., a high available and reliable computer cluster) from those where actual WSs are placed. Engine Threads remotely invoke the actual component WSs. The Execution Engine needs to have access to the WSs Registry, which contains the *WSDL* and *OWL-S* documents. The knowledge required at runtime by each Engine Thread (e.g., WS semantic and ontological descriptions, WSs predecessors and successors, and execution flow control) can be directly extracted from the CPNs in a shared memory implementation or sent by the Execution Engine in a distributed implementation.

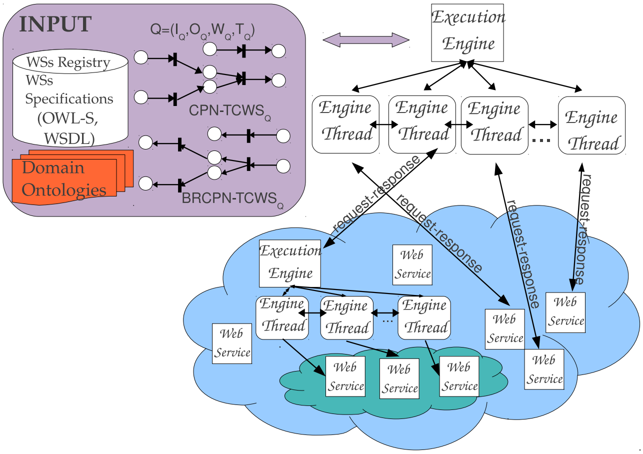


Fig. 5. Executor Architecture

Typically, WSs are distinguished in *atomic* and *composite* WSs. An atomic WS is one that solely invokes local operations that it consist of (e.g., *WSDL* and *OWL-S* documents define atomic WSs as a collection of operations together with abstract descriptions of the data being exchanged). A composite WS is one that additionally accesses other WSs or invokes operations of other WSs. We consider that transitions in the CPN, representing the TCWS to be executed, could be atomic WSs or TCWSs. Atomic WSs have its corresponding *WSDL* and *OWL-S* documents. TCWSs can be encapsulated into an Executor; in this case, the Execution Engine has its corresponding *WSDL* and *OWL-S* documents. Hence, TCWSs may themselves become a WS, making the TCWS execution a recursive operation, as it is shown in Figure 5.

5.1 Checkpointing Algorithms

This section explains how the fault-tolerant execution control was extended in order to incorporate the checkpointing mechanism. The whole execution process is divided in several phases, in which the Execution Engine and Engine Threads can participate.

Initial Phase: Whenever an Execution Engine receives a CPN-EP_Q and its corresponding BRCPN-EP_Q, it starts an Engine Thread responsible for each transition in CPN-EP_Q, indicating to each one its predecessor and successor transitions according to the CPN-EP_Q structure; this step means that the Execution Engine sends the part of the CPN-EP_Q that each Engine Thread concerns on; then it sends values of attributes in I_Q to Engine Threads in charge of WSs who receive them. In Algorithm 4, lines 1 to 14 describe these steps.

WS Invocation Phase: Once each Engine Thread is started, it waits until its inputs are produced. When an Engine Thread receives all the needed inputs, it invokes its corresponding WS. When a WS finishes successfully, the Engine Thread sends values of WS outputs to Engine Threads representing successors of its WS. This step emulates the firing rules in the CPN. Note that all fireable transitions can be invoked in parallel. If a WS fails during the execution, the *Checkpointing phase* is executed, in this case the Engine Thread sends faulty values to its successors to initiate the checkpointing process. When an Engine Thread receives at least one faulty value among its needed inputs, the *Checkpointing phase* is executed. Algorithm 3, lines 1 to 7 and Algorithm 4, line 17 to 18 describe these steps for Engine Thread and the Execution Engine, respectively.

Final Phase: This phase is carried out by both Execution Engine and Engine Threads. If the TCWS was successfully executed, the Execution Engine notifies all Engine Threads by sending the *finish* message, recalculates the Quality of TCWS in case some WSs were replaced, and returns the values of attributes in O_Q to the user. When an Engine Threads receives the *finish* message, it exits. In case that compensation is needed, the Execution Engine receives a *compensate* message, the process of executing the TCWS is stopped, and the compensation process is started by sending a *compensate* message to all Engine Threads. If an Engine Thread receives a *compensate* message, it launches the compensation protocol. If an Execution Engine receives a faulty value in at least one of the O_Q attributes, it executes the *Checkpointing phase*. Algorithm 3, lines 8 to 10, describe these steps for Engine Threads, and Algorithm 4, lines 15 to 21 describe these steps for Execution Engine.

Replacing Phase: This phase is carried out by an Engine Thread when a failure occurs during the execution of its WS. The Engine Thread tries to replace the faulty WS by a substitute and from candidates, it selects the best one according a quality function. According to the transactional property of TCWS, this phase should be executed until success or can be executed for a maximum number of times ($MAXTries$).

Compensation Phase: This phase, carried out by both Execution Engine and Engine Threads, is executed if a failure occurs in order to leave the system in a consistent state. The Engine Thread responsible of the faulty WS informs Execution Engine about this failure. The Execution Engine sends a message *compensate* to all Engine Threads and starts the compensation process following a unrolling algorithm over BRCPN-$TCWS_Q$. Once the rest of Engine Threads receive the message *compensate*, they apply the firing rules in BRCPN-$TCWS_Q$ to follow the compensation process.

Checkpointing Phase: This phase is carried out by the Execution Engine and the Engine Threads who cannot invoke their corresponding WSs, because they are in the path of a failure. The Engine Thread sends faulty values to its successors, saves its state (snapshot), and sends it to the Execution Engine. The snapshot consists of values of input attributes (correct and faulty), the name of its WS, and successors. The correct values obtained in the input attributes of the failed transitions will be the I'_Q required to restart the execution of the TCWS. The Execution Engine saves the correct values of O_Q attributes, collects the snapshots of Engine Threads and return this partial response to the user along with the global snapshot, which is the part of CPN-EP_Q that could no be executed (PARTIAL-CPN-EP_Q). Algorithm 1 shows this phase for the Execution Engine and Engine Threads.

Restart Phase: This phase is carried out by the Execution Engine. First, all the required data is obtained from the previously saved global snapshot. Similar to the *Initial phase*, the Execution Engine starts an Engine Thread responsible for each transition in PARTIAL-CPN-EP_Q, it removes the valid tokens and values from failed transitions and builds I'_Q with those values, sends the I'_Q to the corresponding Engine Thread and the unrolling algorithm over PARTIAL-CPN-EP_Q is started by executing *Invocation phase* and *Final phase.* Algorithm 2 describes this phase for the Execution Engine; whilst the Engine Threads do not take any special action for this phase.

Algorithms for the Replacing and Compensation phases are not shown here for space reasons. They can be found in [8]. Figure 7 depicts the flow diagrams showing the phases previously described for the Execution Engine and Engine Threads, respectively.

5.2 Experimental Results

We developed a prototype of our proposed approach using Java 6 and MPJ Express 0.38 library to allow the execution in distributed memory environments. The deployment was made in a cluster of PCs: one node for the Execution Engine and one node for each Engine Thread needed to execute the TCWS. All PCs have the same configuration: Intel Pentium 3.4GHz CPU, 1GB RAM, Debian GNU/Linux 6.0, and Java 6. They are connected through a 100Mbps Ethernet interface.

We generated 80 TCWSs of sizes from 3 to 10 WSs. All those TCWSs were automatically generated by a composition process [6], from synthetic datasets comprised by 800 WSs with 7 replicas each, for a total of 6400 WSs. Replicas of WSs have different response times.

The OWLS-API 3.0 was used to parse the WS definitions and to deal with the OWL classification process.

Algorithm 1. Checkpointing

```
begin
    Execution Engine:
    begin
        Save received right values of $O_Q$;
        Collect $Snapshots$ from Engine Threads
        ($ETWS_{ws}$);
        $I'_Q \leftarrow correctvaluesfromallSnapshots$;
        Build PARTIAL-CPN-$EP_Q$;
        Save PARTIAL-CPN-$EP_Q$ as $globalsnapshot$;
        Return $GlobalSnapshot\ reference$;
    end
    Engine Threads:
    begin
        Send faulty values to $Sucessors\_ETWS_{ws}$;
        $Snapshot\_ETWS_{ws} \leftarrow$ received right values and
        $Sucessors\_ETWS_{ws}$;
        Send $Snapshot\_ETWS_{ws}$ to EXECUTION ENGINE;
        Return /* the ENGINE THREAD finishes */;
    end
end
```

Algorithm 2. EXECUTION ENGINE Restart

```
Input: $GS$: a reference to a Global Snapshot
begin
    Execution Engine:
    begin
        Load $Q$, PARTIAL-CPN-$EP_Q$, BRCPN-$EP_Q$,
        $OWS$ (Ontology of WSs), $OV_Q$ (list of values of
        $o \mid o \in O_Q$), $I'_Q$, $InputsNeeded$ from $GS$;
        /*$I'_Q$ represents the correct values obtained before
        failure */
    end
    repeat
        Instantiate an $ETWS_{ws}$;
        Send $Predecessors\_ETWS_{ws} \leftarrow^{\bullet} (^{\bullet}ws)$;
        Send $Successors\_ETWS_{ws} \leftarrow (ws^{\bullet})^{\bullet}$;
        Send $InputsNeeded\_ETWS_{ws}$; /*Inputs already
        received by the $ETWS_{ws}$*/
        /* each ENGINE THREAD keeps the part of CPN-$EP_Q$
        and BRCPN-$EP_Q$ which it concerns on*/
    until
    $\forall ws \in S \mid (ws \neq ws_{EE_i}) \wedge (ws \neq ws_{EE_f}) \wedge \neg(\forall a \in$
    $InputsNeeded\_ETWS_{ws}, M(a) = card(^{\bullet}a))$;
    Send values of $I'_Q$ to $ETWS_{ws}$ receiving them ;
    Execute Final phase;
end
```

Fig. 6. Checkpointing & Restart Algorithms

In order to test the checkpointing mechanism, a randomly selected WS fails during the execution of each TCWS allowing to continue the Petri Net unrolling and receive all the outputs that were not affected by the failure. We executed TCWSs comprised of 3, 4, 5, 6, 7, 8, 9, and 10 WSs. Each TCWS was executed 100 times, for a total of 8000 executions. Table 2 shows the different percentages of outputs received in presence of failures during the 100 executions and the

Algorithm 3. Engine Thread Algorithm

```
   Input: $Predecessors\_ETWS_{ws}$, predecessors WSs of $ws$
   Input: $Successors\_ETWS_{ws}$, successors WSs of $ws$
   Input: $WSDL_{ws}, OWLS_{ws}$, semantic web documents
   Input: $MAXTries$: Max number of tries to replace a faulty WS
 1 Invocation phase:
   begin
       $InputsNeeded\_ETWS_{ws} \leftarrow getInputs(WSDL_{ws}, OWLS_{ws})$;
 2     repeat
           Wait Result from ($Predecessors\_ETWS_{ws}$);
           Set values to $InputsNeeded\_ETWS_{ws}$;
       until $\forall a \in InputsNeeded\_ETWS_{ws}, M(a) = card(^{\bullet}a)$;
       /* all the predecessor transitions have finished */
       if $\exists a \in InputsNeeded\_ETWS_{ws} \mid M(a) \in Bag(\{e\})$ then
           /*one or more predecessors transitions have finished unsuccessfully
           Execute Checkpointing phase;
 3     $success \leftarrow false$;
       $cantry \leftarrow true$;
       $tries \leftarrow 0$;
       $equivalents \leftarrow getEquivalents(WSDL_{ws}, OWLS_{ws})$;
 4     $\zeta(ws') \leftarrow R$;
 5     repeat
           Invoke $ws$;
           if $(ws\ fails)$ then
               if $TP(ws) \in \{pr, ar, cr\}$ then
 6                 Re-invoke WS;
               else
                   Execute Replacing phase;
               /*forward recovery*/
           else
               Wait Result from $ws$;
               $\zeta(ws') \leftarrow E$;
               Remove tokens from inputs of $ws$;
               Send Results to $Successors\_ETWS_{ws}$;
               $success \leftarrow true$;
       until $(success) \vee (\neg cantry)$;
 7     if $\neg success$ then
           if $checkpointing\ is\ enabled$ then
               Execute Checkpointing phase;
           else
               Send $compensate$ to Execution Engine;
               $\zeta(ws') \leftarrow C$ ;
               Execute Compensation phase;
           /*backward recovery*/
       else
           Execute Final phase;
   end
 8 Final phase:
   begin
 9     Wait $message$;
       if $message\ is\ Finish$ then
           Send $Finish$ message to $Predecessors\_ETWS_{ws}$;
           Return;
       else
           if $message\ is\ Snapshot$ then
               Send $ETWS_{ws}\ snapshot$ message to Execution Engine;
10             Return;
           else
               Execute Compensation phase;
   end
```

Algorithm 4. Execution Engine Algorithm

```
Input: $Q = (I_Q, O_Q, W_Q, T_Q)$, the user query
Input: CPN-$EP_Q = (A, S, F, \xi)$, a CPN representing a TCWS
Input: BRCPN-$EP_Q = (A', S', F^{-1}, \zeta)$, a CPN representing the compensation flow of the
       TCWS
Input: $OWS$: Ontology of WSs
Output: $OV_Q$: List of values of $o \mid o \in O_Q$
Output: $Quality_Q$: quality obtained after executing CPN-$EP_Q$
Initial phase:
begin
    Insert $ws_{EE_i}$ in CPN-$EP_Q \mid ((ws_{EE_i})^\bullet = I_Q) \wedge ((^\bullet ws_{EE_i}) = \emptyset)$;
    Insert $ws'_{EE_i}$ in BRCPN-$EP_Q \mid (^\bullet ws'_{EE_i} = \{a' \in A' \mid (a')^\bullet = \emptyset\}) \wedge ((ws'_{EE_i})^\bullet = \emptyset)$;
    Insert $ws_{EE_f}$ in CPN-$EP_Q \mid ((ws_{EE_f})^\bullet = \emptyset) \wedge ((^\bullet ws_{EE_f}) = O_Q)$;
    Insert $ws'_{EE_f}$ in BRCPN-$EP_Q \mid (^\bullet ws'_{EE_f} = \emptyset) \wedge ((ws'_{EE_f})^\bullet = \{a' \in A' \mid {}^\bullet a' = \emptyset\})$;
    $\forall a \in (A \cap I_Q),\ M(a) = 1 \wedge \forall a \in (A - I_Q),\ M(a) = 0$;
    /* Marks the CPN-$EP_Q$ with the Initial Marking*/
    $\forall s' \in S',\ \zeta(s') \leftarrow I$;
    /* the state of all transitions in BRCPN-$EP_Q$ is inicial */
    repeat
        Instantiate an $ETWS_{ws}$;
        Send $Predecessors\_ETWS_{ws} \leftarrow^\bullet (^\bullet ws)$;
        Send $Successors\_ETWS_{ws} \leftarrow (ws^\bullet)^\bullet$;
        Send $WSDL_{ws}, OWLS_{ws}$; /* documentos semánticos */
        /* each Engine Thread keeps the part of CPN-$EP_Q$ and BRCPN-$EP_Q$ which it
        concerns on*/
    until $\forall ws \in S \mid (ws \neq ws_{EE_i}) \wedge (ws \neq ws_{EE_f})$;
    Send values of $I_Q$ to $(ws_{EE_i})^\bullet$;
    Execute Final phase;
end
Final phase:
begin
    repeat
        Wait Result from $(^\bullet(^\bullet ws_{EE_f}))$;
        if message compensate is received then
            Execute Compensation phase; /*this phase is shown in [8]*/ Exit Final
            phase;
        else
            Set values to $OV_Q$;
    until $(\forall o \in O_Q, M(o) = card(^\bullet o)$;
    /*$o$ has a value an all predecessor transitions have finished*/
    if $\exists a \in OV_Q \mid M(o) \in Bag(\{e\})$ then
        /*one or more predecessors transitions of $ws_{EE_f}$ have finished unsuccessfully*/
        Execute Checkpointing phase;
    else
        Send Finish message to $^\bullet(^\bullet ws_{EE_f})$;
        $Quality_Q \leftarrow recalculate\_Quality(S)$;/* Quality is recalculated in case some WSs
        were replaced */
        Return $OV_Q$,$Quality_Q$;
end
```

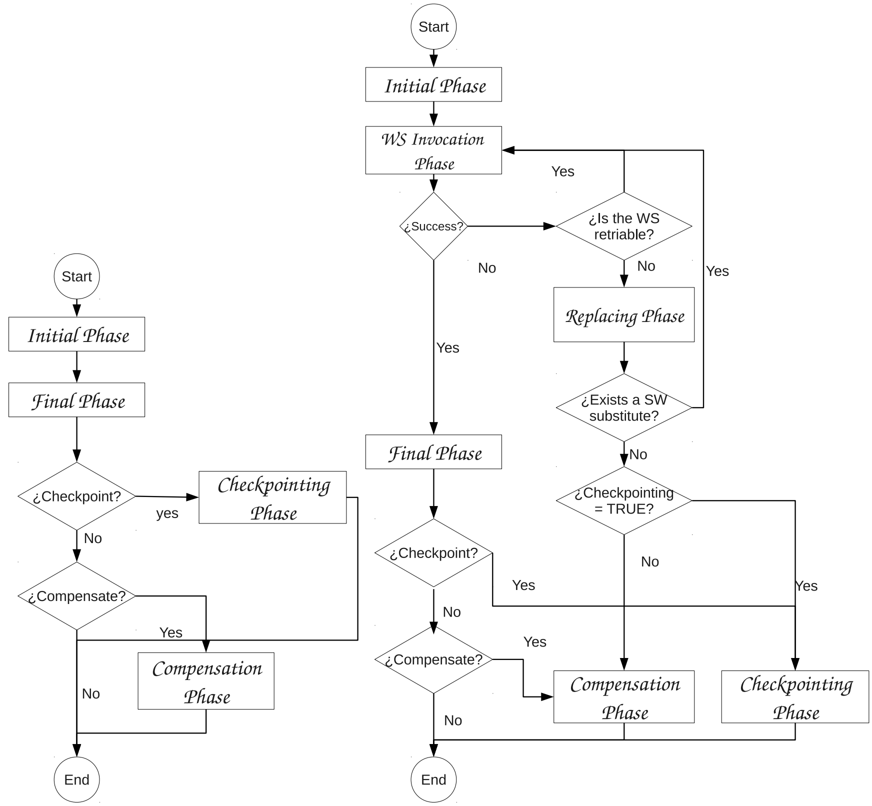


Fig. 7. Execution Engine & Engine Thread flow diagrams

number of WSs that will take part on the execution restart in order to get the 100% of the outputs. For example, for the TCWS of size 10, there was obtained the 91%, 86%, or 81% of the outputs during different executions, and there was not possible to execute a maximum of 3 WSs out of 10.

The *all-or-nothing* property is then relaxed to *something-to-all*, since it is possible to generate partial results and deliver them to the user, whilst the rest of the execution can be performed later; of course, it is up to the user to determine the usefulness of the partial results.

6 Related Work

Related work in the field of checkpointing for TCWSs is scarce. Prior works can be classified into two broad categories: works that require the user to specify the exact checkpointing location [15, 19, 21] and works that perform checkpointing in an automatic fashion[17, 22].

Table 2. Partial Outputs Results

TCWS **size**	**Output received (%)**	**WSs not executed after the failure**
03	72 - 58 - 43	1 - 2
04	78 - 67 - 56	1 - 2
05	82 - 73 - 64	1 - 2
06	85 - 77 - 70	1 - 2
07	87 - 80 - 74 - 60	1 - 2 - 3
08	88 - 82 - 77	1 - 2
09	90 - 85 - 79	1 - 2
10	91 - 86 - 81	1 - 2 - 3

The problem addressed in [15] is the strong mobility of CWSs; which is defined as the ability to migrate a running WS-BPEL process from a host to another to be resumed from a previous execution state. The proposed solution uses Aspect-Oriented Programming (AOP) in order to enable dynamic capture and recovery of a WS-BPEL process state. In [21] authors present a checkpointing approach based on Assurance Points (APs) and the use of integration rules. An AP is a combined logical and physical checkpoint, which during normal execution, stores execution state and invokes integration rules that check pre-conditions, post-conditions, and other application rule conditions. APs are also used as rollback points. Integration rules can invoke backward recovery to specific APs using compensation as well as forward recovery through rechecking preconditions before retry attempts or through execution of contingencies and alternative execution paths. APs together with integration rules provide an increased level of consistency checking as well as backward and forward recovery actions. This work does not specify the use of APs to restart the execution of the CWS later, or in another system. The goal of [19] is to provide a checkpointing scheme as the foundation for a recovery strategy for interorganizational information exchange. The authors adopt concepts from the mobile computing literature to decompose workflows into mobile agent-driven processes that will prospectively attach to web services-based organizational *docking stations*. This decomposition is extended in order to define logical points, within the dynamics of the entire workflow execution, that provide for locating accurate and consistent states of the system for recovery in case of a failure.

In contrast with works presented in [15, 19, 21], our checkpointing strategy is transparent to users and WS developers. They only have to ask for that facility, when a TCWS is submitted to be executed. As these works do, our strategy can be combined with backward and forward recovery techniques.

Recently research has been done in contrast to the checkpointing techniques wherein users have to specify the checkpointing location. In [22] authors propose a checkpointing policy which specifies that when a WS calls another WS, the calling WS has to save its state. The proposed checkpointing policy uses Predicted Execution Time (PET) and Mean Time Between Failures (MTBF), to decide on each WS invocation whether a checkpoint has to be taken or not.

For example, is a WS with PET $<$ MTBF is called, then it is known that it will complete its execution within its MTBF and there is no need for checkpointing. In [17] the idea of checkpoints is rather to keep the execution history containing all successful operations, and at resume time, the system starts the workflow from the beginning but skips all operations that succeeded earlier.

As our approach, works described in [17, 22], proceed with checkpoints, without user intervention. In contrast, in our strategy, checkpoints are taken only in case of failures, so we do not increase the overhead while the execution is free of failures.

7 Conclusions and Future Work

In this work, we have presented a formal definition based on CPN properties for our checkpointing approach, providing an alternative to the previously presented all-or-nothing fault-tolerance mechanisms of WS retry, WS substitution, and compensation. The checkpointing mechanism defined in this paper allows to relax the all-or-nothing property to a *something-to-all* property. The idea is to execute a TCWS as much as possible (in the presence of failures) and then, taking a snapshot of that state. This mechanism allows users to receive partial answers as soon as they are produced (*something*) and provides the option of restarting the TCWS (to get *all* later) without losing the work previously done and without affecting the original transactional property. The formal definition was done by extending definitions of the CPN unrolling execution process and introducing new ones specific to checkpointing. We are currently working on an implementation comprising all our proposed fault-tolerance mechanisms in order to study and compare the performance among them and to provide a fully working real-world implementation.

References

1. Angarita, R., Cardinale, Y., Rukoz, M.: Faceta: Backward and forward recovery for execution of transactional composite ws. In: RED 2012 (2012)
2. Benjamins, R., Davies, J., Dorner, E., Domingue, J., Fensel, D., López, O., Volz, R., Wahler, A., Zaremba, M.: Service web 3.0, Tech. report, Semantic Technology Institutes International (2007)
3. Blanco, E., Cardinale, Y., Vidal, M.-E.: Aggregating Functional and Non-Functional Properties to Identify Service Compositions, pp. 1–36. IGI BOOK (2011)
4. Brogi, A., Corfini, S., Popescu, R.: Semantics-based composition-oriented discovery of web services. ACM Trans. Internet Techn. 8(4), 1–39 (2008)
5. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: Web service selection for transactional composition. In: Int. Conf. on Computational Science (ICCS). Elsevier Science-Procedia Computer Science Series, vol. 1(1), pp. 2689–2698 (2010)
6. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: CPN-TWS: A colored petri-net approach for transactional-qos driven web service composition. Int. Journal of Web and Grid Services 7(1), 91–115 (2011)

7. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: Transactional-aware Web Service Composition: A Survey. IGI Global - Advances in Knowledge Management (AKM) Book Series, ch. 6, pp. 2–20 (2011)
8. Cardinale, Y., Rukoz, M.: Fault tolerant execution of transactional composite web services: An approach. In: Proc. of the Fifth Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM (2011)
9. Cardinale, Y., Rukoz, M.: A framework for reliable execution of transactional composite web services. In: Proc. of the Int. Conf. on Management of Emergent Digital EcoSystems, MEDES (2011)
10. El Haddad, J., Manouvrier, M., Rukoz, M.: TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition. IEEE Trans. on Services Computing 3(1), 73–85 (2010)
11. Hoffmann, J., Weber, I., Scicluna, J., Kaczmarek, T., Ankolekar, A.: Combining Scalability and Expressivity in the Automatic Composition of Semantic Web Services. In: Proc. of 8th Int. Conf. on Web Eng. (ICWE), pp. 98–107 (2008)
12. Hogg, C., Kuter, U., Munoz-Avila, H.: Learning Hierarchical Task Networks for Nondeterministic Planning Domains. In: The 21st Int. Joint Conf. on Artificial Intelligence, IJCAI 2009 (2009)
13. Ben Lakhal, N., Kobayashi, T., Yokota, H.: FENECIA: failure endurable nested-transaction based execution of composite Web services with incorporated state analysis. VLDB Journal 18(1), 1–56 (2009)
14. Liu, A., Li, Q., Huang, L., Xiao, M.: FACTS: A Framework for Fault Tolerant Composition of Transactional Web Services. IEEE Trans. on Services Computing 3(1), 46–59 (2010)
15. Marzouk, S., Maâlej, A.J., Jmaiel, M.: Aspect-oriented checkpointing approach of composed web services. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 301–312. Springer, Heidelberg (2010)
16. Mei, X., Jiang, A., Li, S., Huang, C., Zheng, X., Fan, Y.: A compensation paired net-based refinement method for web services composition. Advances in Information Sciences and Service Sciences 3(4) (2011)
17. Podhorszki, N., Ludaescher, B., Klasky, S.A.: Workflow automation for processing plasma fusion simulation data. In: WORKS 2007: Proceedings of the 2nd Workshop on Workflows in Support of Large-Scale Science, pp. 35–44. ACM, New York (2007)
18. Rukoz, M., Cardinale, Y., Angarita, R.: Faceta*: Checkpointing for transactional composite web service execution based on petri-nets. Procedia Computer Science 10, 874–879 (2012)
19. Sen, S., Demirkan, H., Goul, M.: Towards a verifiable checkpointing scheme for agent-based interorganizational workflow system "docking station" standards. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS 2005, vol. 07, p. 165.1. IEEE Computer Society, Washington, DC (2005)
20. Thakker, D., Osman, T., Al-Dabass, D.: Knowledge-intensive semantic web services composition. In: Tenth Int. Conf. on Computer Modeling and Simulation, pp. 673–678 (2008)
21. Urban, S.D., Gao, L., Shrestha, R., Courter, A.: Achieving recovery in service composition with assurance points and integration rules. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010, Part I. LNCS, vol. 6426, pp. 428–437. Springer, Heidelberg (2010)
22. Vani Vathsala, A.: Article: Optimal call based checkpointing for orchestrated web services. International Journal of Computer Applications 36(8), 44–50 (2011)
23. Yu, Q., Liu, X., Bouguettaya, A., Medjahed, B.: Deploying and managing web services: issues, solutions, and directions. The VLDB Journal 17, 537–572 (2008)

A Game Theoretic Approach for Resource Usage[⋆]

Paraskevas V. Lekeas

Talk3, P.O. Box 441
Wilmette, IL 60091 USA
plekeas@gmail.com

Abstract. Existing web infrastructures support the publication of a tremendous amount of resources, and over the past few years Data Resource Usage has become an everyday task for millions of users all over the world. In this work we model Resource Usage as a Cooperative Cournot Game in which a resource user and the various resource services are engaged. We give quantified answers as to when it is of interest for the user to stop using part of a resource and to switch to a different one. Moreover, we do the same from the perspective of a resource's provider. We show that providers should use their resources in a cooperative manner to prevent a user's deviation. We also prove that providers can avoid losing users if they reduce user cost for the offered services, if they increase their resource size, and, finally, if they reduce the number of their unpopular services.

Keywords: Resource Usage, Cournot Competition, Game, Cooperation, Deviation, Google.

1 Introduction

Lately, Data Resource Usage is an everyday task for millions of users all over the world. In this paper when we refer to a resource we have in mind that in the background there exists a set of electronic mechanisms or internet infrastructures that created this resource for the purpose of value generation (either a profit when there is an underlying company or some other social gain, such as [2]). See also [3] for a related taxonomy. Exchanging information, communicating, working and various other aspects of our life have been inevitably affected by data repositories, which can be accessed through various channels, such as the Web and the Internet via different technologies, interfaces and infrastructures [4]. Usually, once someone has identified an appropriate resource, he interacts with it by exchanging information. This sort of interaction is heavily commercialized, and a huge industry has been established, which invests a great amount of money in marketing web services and products that provide access to resources

[⋆] Work done while author was visiting the Department of Electrical and Mechanical Engineering, School of Greek Army Engineer Officers, Athens, Greece (STEAMX). A preliminary version of this paper appeared in [1].

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 45–55, 2013.

quite often freely. To give a picture, only in 2009 end-user spending for IT services worldwide was estimated to be $763 billion [5]. This is why in this paper from now on we will use the word "provider" to refer to the underlying structure responsible for a resource. These providers most of the time are extremely interested in developing integrated resource services[1] in order to attract users, and, more importantly, to convince them to keep using these. This is because users are valuable: They provide information to the resource by interacting with it, they bring money through the adds or subscription fees, and of course they bring new users to broaden the profit cycle.

One living example is *Google*, which proposes a web resource experience through the integration of different technologies in order for users to continue using its services. Opening a *Google* account is a fairly easy one-minute process and instantaneously the new user has access to different cutting-edge services like *Android* OS apps, customized web search through the *Google* search engine, cloud services, access to landline phone calls, teleconferencing services and much more. A "perfect" user of *Google* would be the one who uses all these services explicitly through the *Google* APIs, sharing no data with any other competitive resource (e.g. *AWS* [6], *Ubuntu One* [7] or *Yahoo!* [8]) and thus enriching only *Google*'s resource knowledge repositories. However, many times it is the case that not every service or technology of a resource is welcomed by users and sometimes users tend to accept only specific services from a resource, while ignoring some others. Also a situation that is not so good for a provider is the case where users decide to quit its resource and switch to a different one that provides similar or better services [9].

Researching the above we face the following questions: When do users tend to partially[2] abandon a specific resource? What can resource providers do about that? Is there a way to formulate the above trends in order to be evaluated and measured?

Driven by these questions we model the various user - service interactions within a resource with different plays of the user, which are engaged either in a cooperative or in a non-cooperative manner. Each of these plays generates some value, which is to be conceived as a measure of the user's satisfaction for the appropriate service.

Having this in mind, in the rest of the paper we proceed as follows: Section 2 presents the production problem, a problem borrowed from economics, that can help us understand the notions of cooperation and non-cooperation. Section 3 presents the model and section 4 applies it to Resource Usage. Finally, section 5 concludes.

2 The Production Problem

Adopting the following problem from economics will help us better understand our proposed model for Resource Usage.

[1] We prefer this term instead of the term "web service" since many other alternative channels exist like satellite and cellphone grids, ad hoc networks, etc.

[2] Partially means that the user is unsatisfied only with some of the services and wants to switch but likes the rest and wants to keep them.

Visualize that in a market two car factories A_1, A_2 compete. Factory A_1 produces q_1 cars, while factory A_2 produces q_2 cars. Since they produce the same category of cars we may assume, as a working hypothesis, that the cost for each car produced is fixed, c. Now these two factories want to sell these cars in the market. If α is a factor that represents the size of the market and $Q = q_1 + q_2$ is the total (cummulative) production of both factories, then $\alpha - Q$ is a factor that can describe the way prices fluctuate in the market. And this is because if more cars are produced, then Q will increase and the prices will fall since consumers would find plenty of cars in the market to choose from. If, on the other hand, less cars are produced, then Q will decrease and the prices will go up since now cars would be sparse in the market. So the factories decide to follow the above trends and sell their products with the following prices:

$$\begin{aligned} \text{selling price of } A_1 : &\quad (\alpha - Q)q_1 \\ \text{selling price of } A_2 : &\quad (\alpha - Q)q_2 \end{aligned}$$

Each factory's net profit π_i, $i = 1, 2$ after selling its production is found if we subtract the cost of production from the revenue (the selling price):

$$\begin{aligned} \pi_1 &= (\alpha - Q)q_1 - cq_1 \\ \pi_2 &= (\alpha - Q)q_2 - cq_2 \end{aligned} \tag{1}$$

Moreover, since each factory is interested in maximizing its profit, A_1 and A_2 should try to maximize equations (1) with respect to quantities q_i, $i = 1, 2$, and this is because factories only control their production in the market and nothing else. So the production problem is the following: *How many cars should each factory produce in order to achieve maximum profit?*

Effectively, there are two approaches to solve the production problem. The first one is a non-cooperative approach and uses the idea of Cournot competition.

2.1 A Non-cooperative Approach

Regarding this approach we assume that the factories are engaged in a Cournot competition [10]. In this competition both factories act independently and care only about themselves. They take into account each other's potential decisions, but once they decide about their production they announce it simultaneously to the market and start producing. Thus, using (1) factories face the following optimization problems:

$$\pi_i = \max_{q_i} (\alpha - q_1 - q_2 - c)q_i, \; i = 1, 2 \tag{2}$$

Calculating (2) we have:

$$\begin{aligned} \frac{\partial \pi_1}{\partial q_1} = 0 \Rightarrow \frac{\alpha - q_2 - c}{2} = q_1 \\ \frac{\partial \pi_2}{\partial q_2} = 0 \Rightarrow \frac{\alpha - q_1 - c}{2} = q_2 \end{aligned}$$

Yielding the solution is straightforward since we get that $q_1 = q_2 = \frac{\alpha - c}{3}$ and using (1) we have that the maximum profit of each factory is:

$$\frac{(\alpha - c)^2}{9} \tag{3}$$

If we assume that there exist n factories in the market, we can easily generalize the above result to get the so-called Cournot theorem:

Theorem 1. *(Cournot, 1838) When n factories compete in the market in a Cournot competition each factory's maximum profit is* $\left(\frac{\alpha - c}{n+1}\right)^2$.

Let us now present the second approach to the production problem which is based on the notion of cooperation.

2.2 A Cooperative Approach

Suppose that one day factory A_1 comes up with the following idea: *What if I cooperate with A_2, we merge our productions, and then split the profit?* So A_1 proposes this to A_2, and after some thought they agree to merge their forces and act as a unity, producing cars, and, after selling them, to split the profit. Since they still want to maximize their profits they should decide on how much the new factory (after the merge) should produce, but since now the two factories act as a unity they only face a single equation to maximize:

$$\pi = \max_q (\alpha - q - c)q$$

where q is the common production. Solving this equation gives:

$$\frac{\partial \pi}{\partial q} = 0 \Rightarrow \frac{\alpha - c}{2} = q \Rightarrow \pi = \frac{(\alpha - c)^2}{4} \tag{4}$$

but[3] since they agreed to split (equally) the profit each factory will get:

$$\frac{\pi}{2} = \frac{(\alpha - c)^2}{8} \tag{5}$$

Using (4) we can easily generalize the above result for the case where n factories equally split the cooperative profit to have:

$$\frac{\pi}{n} = \frac{(\alpha - c)^2}{4n} \tag{6}$$

In our two-factory example and from (3) and (5) we can see that since:

$$\frac{(\alpha - c)^2}{8} > \frac{(\alpha - c)^2}{9}$$

cooperation is always better than non-cooperation. This fundamental difference between the cooperative and the non-cooperative solution is the key issue to our model that we will present in the next section. Fig. 1 illustrates the two approaches.

[3] The same result can be derived if we apply the Cournot theorem for $n = 1$, having in mind that now both factories are merged and constitute a unity.

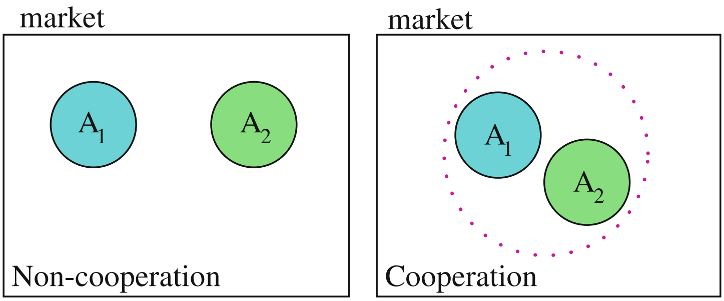


Fig. 1. On the left and under non-cooperation each factory earns $\frac{(\alpha-c)^2}{9}$, while on the right and upon cooperation the average profit is $\frac{(\alpha-c)^2}{8}$

3 The Model

Suppose that a user u decides to sign up for a resource R in order to *use* its i, $i \in \{1, \cdots, n\}$, different services. Let us use *Google* [11] in place of R in order to make the exposition more attractive to the reader. The phrase "use a service" refers to the interaction of u with the service in order for some desired tasks to be completed. For example, using *Gmail* to send an email, entering data by typing in *GoogleDocs*, downloading a *.jpg* file from *Picasa* can be perceived as parts of such an interaction. Let p_i denote the interaction of u with service i. Call each p_i a *play* that u does with the respective service. Every play[4] generates some value for u. Since u needs to make an effort to generate this value, we assume that the *per unit cost* to u for playing p_i is c. This cost can represent the time spent by the user to complete a specific task. We assume for clarity of the exposition that c is the same for every p_i, $i = 1, \cdots, n$. If we now take the value generated from a play and subtract the cost spent to produce it, we will find this play's *worth* to the user, or in other words the profit from using the specific service, which intuitively represents a measure of how satisfied the user is with the service, i.e.:

$$\textit{worth of i-th play} = (\textit{value generated from i-th play}) - (\textit{cost spent for i-th play}).$$

So each play contributes some worth to the user, and if we add all these contributions from all the plays we will have the total worth to the user u from using all the services.

As is generally accepted, maximizing user satisfaction constitutes the key issue to every service. This forces every play p_i to seek to maximize its contribution to the total worth earned by u. But this happens under the following restriction. The user u has a limited time to spend interacting with the services, and thus he must split this time into his needs wisely in order to acomplish his different tasks. This means that no play can monopolize all the available time of u. Moreover, u's multitasking abilities are limited by nature. So spending more time on service

[4] From now on the terms "play" and "interaction" will be used interchangeably.

i might, on the one hand, raise satisfaction from i but on the other might lower satisfaction from service j ($j \neq i$). From the above, the analogy between the production problem and the model just described is clear. Instead of having a market with factories seeking to maximize their profits, we now have the resource and the plays that seek to maximize a user's satisfaction. Thus the production problem in this case can be stated as follows: *How much value should each play with Google produce in order to maximize a user's satisfaction?*

We have already seen in the previous section how we can answer the above question in two different ways. Restating the solutions we can say that whenever the plays do not cooperate, the value generated by each service is given by the Cournot theorem: $\left(\frac{\alpha-c}{n+1}\right)^2$, whereas when they cooperate each play produces a value of $\frac{(\alpha-c)^2}{4n}$ for the user.

Before elaborating on the above and seeing how they can be of help to model resource usage, let us explain what we mean by plays cooperating and not cooperating.

When we say that two or more plays cooperate we mean that they help one another to maximize user satisfaction. For example, consider the following scenario. The user u uses *Gmail* to read an important email that has an attachment in *.doc* format. Unfortunately, the PC that u uses does not have *Microsoft Office* installed, so u is not able to read this important attachment. Moreover, the administrator of the network prevents u from downloading a *.doc* viewer and installing it. However, since *Gmail* cooperates with *GoogleDocs*, user u gets a notification informing him that he can read the attachment using a *.doc* reader available online from *GoogleDocs*. He can also store the attachment there and even edit it. So this is great news for the user because in this cooperative way he accomplishes his task and manages to read the attachment. We have another example of cooperation when the same user u in trying to send an email uses *Picassa* to attach a *.jpg* file to his email through *Gmail*.

On the other hand, it is clear that in a non-cooperative setting all the above could not be accomplished by u and each service would be isolated, acting as an independent satisfaction maximizer.

Before turning our attention to how this idea of cooperation can be of help to u let us go back to the production problem and slightly modify it to exhibit the idea of deviation.

3.1 Production Problem and Deviation

As seen in section 2 there were two approaches to solve the production problem, a cooperative and a non-cooperative one. It can be proven that between these two extremes lies another very interesting idea, that of deviation. For this let us slightly change the setting of the production problem and instead of having 2 factories let us have 4 factories A_1, A_2, A_3, and A_4. These factories realized a long time ago that it is to their benefit to cooperate and thus agreed to regulate the quantities produced, maximizing their profits in the market. However, one day factories A_2, A_3 discover in their laboratories a new innovative technology that can futher boost their profits in the market. This technology is not revealed to the other two factories A_1, A_4, and thus A_2 and A_3 think of not cooperating with the others anymore and of breaking their agreement. But since they want to be careful not to announce something that they are not sure of, they try to see which situation is better for them. Is it good to stay and cooperate with the rest or is it better to deviate and, using this new technology, to earn more? On the one hand, using the cooperative solution (6) for $n = 4$ they know that on average they earn $(\alpha - c)^2/16$. On the other hand, if their worth under this new technology is estimated to be $v(A_2 + A_3)$, then on average upon deviation they will earn $v(A_2 + A_3)/2$. So they should definitely deviate if it is the case that $v(A_2 + A_3)/2 > (\alpha - c)^2/16$. Fig. 2 illustrates the two approaches.

Using the above let us now return to our resource usage problem.

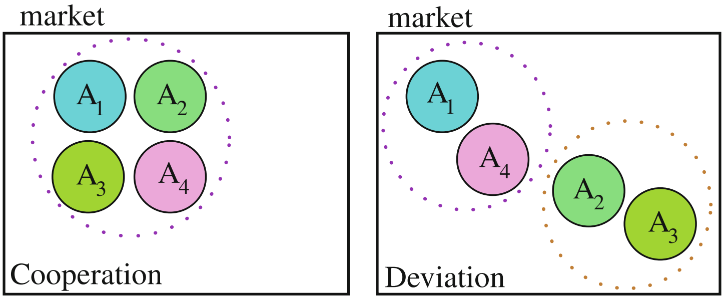


Fig. 2. On the left, when all factories cooperate the average profit is $\frac{(\alpha-c)^2}{16}$, while on the right when two factories deviate each deviant on average will earn $\frac{v(A_2+A_3)}{2}$, where $v(A_2 + A_3)$ is the value of the deviants in the market

4 Modeling Resource Usage

This idea of deviation can be valuable in two ways. First it can be used to reason about the loyalty of u to *Google* and second it can be used from the perspective of a resource provider.

4.1 User Loyalty

Consider the following scenario: u for a long period of time uses a non-empty set N of *Google*'s services, but he realizes that he is not satisfied with a subset $S \subset N$, $S \neq \emptyset$ of them. One day u finds out that a different resource R' (say *Yahoo!*) also offers this set S of services. The user u thinks that he should start trying to use *Yahoo!*'s services as well to compare. Suppose that after doing so he estimates that he earns $v(S)$ from *Yahoo!*. In order to make his final decisions about which resource to use for the set S he reasons as follows: "If I stay loyal to *Google*, from (6) on average I earn $\frac{(\alpha-c)^2}{4n}$. On the other hand, if I abandon S from *Google* and partially switch to *Yahoo!* I would earn on average $\frac{v(S)}{s}$, where $|N| = n$ and $|S| = s$. So I partially switch if $\frac{v(S)}{s} > \frac{(\alpha-c)^2}{4n}$." The previous result is very valuable to u because it provides him with a metric as to how he can approximate his value $v(S)$ using only 3 different parameters[5], the size of the resource α, the number of deviant services s, and the cost c.

4.2 Resource Provider's View

As previously mentioned above, our approach is not only valuable to users but also to a resource provider (*Google*). This is because under our model the provider can follow some strategies to fight a potential rejection by a user. Let us see why:

First of all since

$$\frac{(\alpha-c)^2}{4n} > \left(\frac{\alpha-c}{n+1}\right)^2, \forall n \geq 2 \tag{7}$$

Google realizes that the cooperative solution is always better than the non-cooperative, so it should always provide its services in a cooperative way. And this happens because through this strategy it maximizes satisfaction for users. We can observe this trend in many resource providers nowadays. For example, *Google* in March 2012 unified its services in order to act cooperatively. In this way any two services will cooperate in order to produce a common value and increase a user's satisfaction. Thus through *Gmail* you can view or process your attachments through *GoogleDocs* or use the *Dashboard* to synchronize all your e-mail contacts with your *Android* device. Fig. 3 gives an example of five services of *Google* acting cooperatively.

Moreover, *Google* should definitely try to prevent the following inequality from holding:

$$\frac{v(S)}{s} > \frac{(\alpha-c)^2}{4n} \tag{8}$$

because by not doing so, *Google* will start partially (or totally) losing users! So what *Google* should strive for is to try to increase the right hand side of (8): $\frac{(\alpha-c)^2}{4n}$ in order to maintain its users. This can be done in various ways:

First, *Google* can help u to reduce its cost c. And this can be achieved, for example, by upgrading its hardware, by hiring a qualified service [12], by adopting process completeness strategies [13] or by improving the service tutorials and introducing online help desks [14,15]. Second, *Google* might consider increasing

[5] We consider n known to the user.

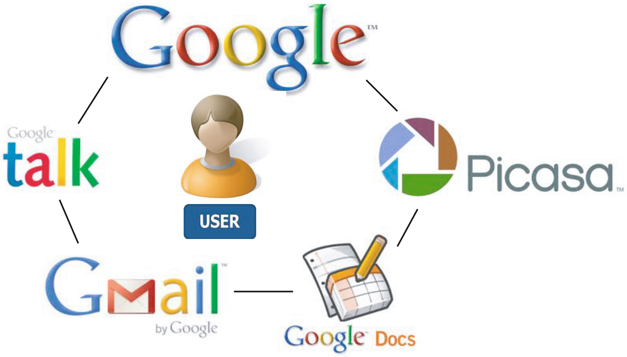


Fig. 3. An example of five services of *Google* acting in a cooperative manner. If we assume that the cost c is the same for every service, then the user earns on average $\frac{(\alpha-c)^2}{20}$ value of satisfaction.

the factor α, which as we said represents the "size" of the resource. For example, for *Google* α might represent the size of the web crawled or indexed by the search engine or how much these crawls are[6]. So it is of benefit for *Google* to broaden the percentage of the web crawled and to keep these data constantly updated. Finally, it might consider reducing the number of services it provides (reducing the denominator of $\frac{(\alpha-c)^2}{4n}$ increases the fraction), for example, by discarding obsolete services or not very popular ones.

Another important factor for *Google* is to collect user rating information data for its services so as to compute its own estimations of how satisfied the users are. According to (8) the closer the provider's estimation $v_{provider}(S)$ is to the user's $v_{user}(S)$, the more an effective strategy can be adopted to maintain its customers, and this is because in this way *Google* has a clearer image of what its users like. Also *Google* should follow user trends to estimate the potential S's that make its resource vulnerable either by asking for feedback from the users or by outsourcing this task to experts [17,18].

Moreover, more complex scenarios can be adopted from *Google*'s point of view in order to further refine its strategies. To do this more complicated games can be designed in which users would engage themselves. For example, *Google* will not only consider how to satisfy the user but also how to earn more money, so instead of having the players compete in order to find the equilibria between value produced and cost spent, we could have players competing between satisfaction offered, money earned and cost spent.

[6] For a social network α might represent the number of active users, so this social network should try to attract more users thus increasing α so that the current users will belong to a bigger society and become more satisfied, resulting in more options for interaction. The same idea also applies to the critical mass of Service Overlay Networks [16], etc.

5 Discussion and Future Work

As stated previously, the value generated by the plays and the cost spent are related to the user's loyalty to the resource. But can the above be calculated by the user? An approach can be the following: On the one hand, the value generated through a user - service interaction must be perceived as a combination A) of the amount of information in the form of data created, exchanged, stored, or retrieved by the user, and B) of the user's personal metrics. For example, one such metric is the one we adopted as a cost factor, i.e. the amount of time the user invested to produce the value through interaction (programming, typing, asking queries, etc.). And this is something natural, but other functions can be used too, such as money spent by the user, bandwidth or CPU resources used, or a combination of the above. One can consult [19] and most of the references therein for a recent treatment of QoS properties and measurement metrics of services.

In our model we used as a notion of fairness one in which the value generated by the players must split equally among them. This is a special case of a so-called game with transferable utilities since the user u makes the decision based on the average value calculated by all the plays. This means that the transferring of profit between any two services x, y is allowed. But since there exist many different notions of fairness such as the Shapley Value, it is of particular interest to extend the analysis to these notions as well.

Finally, since under our model we assumed that each service is somewhat of the same nature as every other service, in a more realistic scenario in which the services differ, we could have used a differentiation parameter γ and the demand from service i would change, thus resulting in a more complex worth function.

Acknowledgement. To the memory of my father, Vasileios G. Lekeas.

References

1. Lekeas, P.V.: Should I Quit Using my Resource? Modeling Resource Usage through Game Theory. In: 5th International Workshop on REsource Discovery (RED 2012), Herakleion, Crete, Greece, May 27-31, in Conjunction with the 9th Extended Semantic Web Conference (ESWC 2012) (2012)
2. https://www.findthemissing.org/en
3. Vanthournout, K., Deconinck, G., Belmans, R.: A taxonomy for resource discovery. Personal Ubiquitous Computing 9(2), 81–89 (2005)
4. Hilbert, M., Lopez, P.: The World's Technological Capacity to Store, Communicate, and Compute Information. Science 332(6025), 60–65 (2011)
5. Blackmore, D., Hale, K., Harris, J., Ng, F., Morikawa, C.: Market Share Analysis: IT Services Rankings, Worldwide, 2009, April 29. Gartner, Inc. (2010)
6. http://aws.amazon.com/
7. https://one.ubuntu.com/
8. http://www.yahoo.com

9. Torkjazi, M., Rejaie, R., Willinger, W.: Hot Today, Gone Tomorrow: On the Migration of MySpace Users. In: Proceedings of the 2nd ACM Workshop on Online Social Networks (WOSN), Barcelona, Spain, pp. 43–48 (2009)
10. Cournot, A.A.: Researches into the Mathematical Principles of the Theory of Wealth (English translation of the original). Macmillan, New York (1897)
11. `http://www.google.com/`
12. Wanchun, D., Lianyong, Q., Xuyun, Z., Jinjun, C.: An evaluation method of outsourcing services for developing an elastic cloud platform. The Journal of Supercomputing (2010), doi:10.1007/s11227-010-0491-2
13. Piccoli, G., Brohman, M.K., Watson, R.T., Parasuraman, A.: Process completeness: Strategies for aligning service systems with customers' service needs. Business Horizons 52(4), 367–376 (2009)
14. Schubert, F., Siu, C., Cheung, H., Peng Chor, L., Shigong, L.: An integrated help desk support for customer services over the World Wide Web - a case study. Computers in Industry 41(2), 129–145 (2000)
15. Schubert, F., Siu, C., Cheung, H., Peng Chor, L.: Web-based intelligent helpdesk-support environment. International Journal of Systems Science 33(6), 389–402 (2002)
16. Lam, N.: Capacity Allocation in Service Overlay Networks. Ph.D. Dissertation, McGill University, Canada (2011)
17. Benko, C.: Outsourcing Evaluation. A Profitable Process. Information Systems Management 10(2) (1993)
18. McIvor, R.: How the transaction cost and resource-based theories of the firm inform outsourcing evaluation. Journal of Operations Management 27(1), 45–63 (2009)
19. D'Mello, D.A., Ananthanarayana, V.S.: Dynamic selection mechanism for quality of service aware web services. Enterprise Information Systems 4(1), 23–60 (2010)

LiQuate-Estimating the Quality of Links in the Linking Open Data Cloud

Edna Ruckhaus and Maria-Esther Vidal

Universidad Simón Bolívar
Caracas, Venezuela
{ruckhaus,mvidal}@ldc.usb.ve

Abstract. During the last years, RDF datasets from almost any knowledge domain have been published in the Linking Open Data (LOD) cloud. The Linked Open Data guidelines establish the conditions to be satisfied by resources in order to be included as part of the LOD cloud, as well as connected to previously published data. The process of publication and linkage of resources in the LOD cloud relies on: *i*) data cleaning and transformation into existing RDF formats, *ii*) storage of the data into RDF storage systems, and *iii*) data interlinking. Because of data source heterogeneity, generated RDF data may be ambiguous and links may be incomplete with respect to this data. Users of the Web of Data require linked data to meet high quality standards in order to develop applications that can produce trustworthy results, but data in the LOD cloud has not been curated; thus, tools are necessary to detect data quality problems. For example, researchers that study Life Sciences datasets to explain phenomena or identify anomalies, demand that their findings correspond to current discoveries, and not to the effect of low data quality standards of completeness or redundancy. In this paper we propose LiQuate, a system that uses Bayesian networks to study the incompleteness of links, and ambiguities between labels and between links in the LOD cloud, and can be applied to any domain. Additionally, a probabilistic rule-based system is used to infer new links that associate equivalent resources, and allow to resolve the ambiguities and incompleteness identified during the exploration of the Bayesian network. As a proof of concept, we applied LiQuate to existing Life Sciences linked datasets, and detected ambiguities in the data, that may compromise the confidence of the results of applications such as link prediction or pattern discovery. We illustrate a variety of identified problems and propose a set of enriched intra- and inter-links that may improve the quality of data items and links of specific datasets of the LOD cloud.

1 Introduction

During the last years, the Linked Open Data guidelines have provided the basis for creating and populating the Web of Data with more than 30 billion data items and around 500 million associations between them. The "W3C SWEO Linking Open Data Community Project" reports datasets in almost any domain of knowledge; biological data items such as genes and proteins, as well as government, cross domain, climatological measurements and scientific papers have been published, and continuously new datasets become available. The process of publication of a dataset in the LOD cloud is composed

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 56–82, 2013.

of several tasks; one of the activities in this process is the *Generation* of RDF data, which is comprised of the transformation, data cleaning and linking tasks [23]. Duplicates in the data can be detected by using entity matching techniques; further, links may be generated manually, or semi-automatically. Usually, this task involves determining the related datasets that may be linking targets, discovering the related items in these datasets, and validating the links that have been created. Due to the nature of these tasks, links among datasets may be incomplete or ambiguous, and some duplicates may not be identified.

Useful tools like the Silk framework[13] and xCurator [12], provide an entity resolution component that discovers missing links between a known set of instances. Maali et al. [17] propose a system, the LGD Publishing Pipeline, where interlinking is done through an extension of Google Refine's reconciliation facility. In a first step, reconciliation is done against any RDF data available through a SPARQL endpoint or as a dump file, and then users may choose some of the reconciliation results or include additional properties in their request, in order to improve the precision of the reconciliation process. All of these tools rely on user specifications about the links to be discovered and the similarity functions that will be used. Similarity functions are approximate, thus, the effectiveness of the linking process may be compromised.

Furthermore, current LOD basic statistics[1] give a disproportionate number of triples and links in the order of 25 billion triples vs 437 million links; other statistics indicate that there are datasets with a total of approximately 6 million triples that have no links; these statistics suggest that links in the LOD cloud may be incomplete.

Quality problems in the LOD cloud have been presented in different contexts [10,14,21]. Halpin et al.[10], discuss the problem of the use and misuse of the *owl:sameAs* construct. If the W3C definition is respected [25], then all *owl:sameAs* related entities should share exactly the same properties. Further, accuracy problems of the *owl:sameAs* property are related to its misuse for related or similar entities.

Jentzsch et al [14] examine the applicability and benefits of using linked data in the Life Sciences domain, specifically for clinical trials, drugs, and related sources. Data quality problems arise also in different areas of research that use linked data; in [21], linked data metrics are used for Expert search in the context of an Open Innovation Scenario, where it is assumed that expert communities use different communication channels and leave different traces, and these traces are available as linked data. Linked data metrics based on data quantity and topic distribution are used to measure the performance of the expert search. However, in their experimental setup, data cleaning was needed because of quality issues, and new topics and links were added to the data sources that were used in the study.

In this work we present LiQuate (Linked Data Quality Assessment) a semi-automatic tool that combines Bayesian networks and rule-based systems to analyze the quality of data in the LOD cloud. Bayesian networks allow a compact representation of the joint distribution of the datasets concepts and thus, are suitable for studying the probability of occurrence of a certain link among datasets or a value within a single dataset. In general, Bayesian networks assume a number of discrete values for each of the variables considered in the network. However, in the context of the Semantic Web, variables that

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/

represent the concepts in large-sized RDF datasets may contain a very large number of values; thus, in our approach we implement structures that allow the aggregation of the data associated with each node in the network. Queries against the Bayesian network represent the probability that different resources have redundant labels or that a link between two resources is missing; thus, the returned probabilities can suggest ambiguities or possible incompleteness and inconsistencies in the data or links. Finally, LiQuate can use a probabilistic rule-based system to infer new links that associate duplicated resources, and to resolve ambiguities and incompleteness identified during the exploration of the Bayesian network.

As a proof of concept, an initial experimental study has been conducted on several datasets in the Life Sciences domain. Some of the detected quality problems are related to drug and disease ambiguities between labels. For example, drug interventions in clinical trials may be linked (*owl:sameAs*, *rdfs:seeAlso* links) to drugs in the **Drugbank**, **DBPedia** and **DailyMed** datasets; however, in practice, redundant interventions do not share the same links to these datasets. Additionally, in some cases there is clearly a lack of *owl:sameAs* or *rdfs:seeAlso* links among datasets, e.g., conditions in clinical trials and diseases in datasets such as **Diseasome** or **DBPedia**.

The structure of this paper is as follows: first, we present two motivating examples; following, we describe our approach comprised of the LiQuate Bayesian network and the Process Model for Link Completeness Assessment. Then, the LiQuate System is described, and the steps involved in constructing a LiQuate Bayesian network are presented. Following, the quality process workflows are illustrated with an example. In the next section, results of an empirical evaluation are reported. The description of the related work follows, and finally, section 8 concludes and outlines interesting future directions.

2 Motivating Examples

Following, we will present motivating examples in two different domains: the first example is related to information that has been published in the LOD cloud about countries and cities through the **DBPedia**[2] and *Telegraphis*[3] hubs; the second example is related to the Linked Clinical Trials, **LinkedCT**[4] project that publishes open semantic web data sources for clinical trial data [11].

Example 1. *Resources in the* **DBPedia** *and Telegraphis datasets related to countries, cities and municipalities may be linked through owl:sameAs links. One of the ambiguities encountered in this domain is that the attributes of two linked owl:sameAs countries may be not the same. This is the case of the population, e.g., Venezuela in* **DBPedia***:* `dbpedia:property/populationCensus` *taken in 2011, is* $23,054,985$*, whereas, in Telegraphis,* `geonames:population` *is* $26,414,000$*; there is no information on the year when this data was valid. Also:* `dbpedia:property/populationEstimate` *is* $27,150$*, which seems an inaccurate number. Additionally:* `dbpedia:property/areaKm` *is* $916,445$*, while*

[2] `http://dbpedia.org/About`, accessed on March 2013.

[3] `http://datahub.io/dataset/telegraphis`, accessed on March 2013.

[4] `http://www.linkedct.org`, data downloaded on September 2011.

the area in the Telegraphis dataset is $912,050$ *square kilometers. Other countries present the same type of ambiguities, e.g., British Virgin Islands have a population of* $27,000$ *in* **DBPedia** *and the population presented in Telegraphis is* $21,730$*.*

State-of-the-art approaches as the one proposed by Hassas et. al [12] rely on similarity functions and word spell checker services to identify resources that possibly correspond to the same country, e.g., Venezuela and venesuela. However, because semantic information that describes the domain of knowledge is not encoded in their linking and cleaning processes, these approaches will not be able to identify or solve the ambiguities presented in the country population published by these two datasets.

Example 2. *Figure 1 represents some of the links between data concepts of the linked datasets:* **Diseasome**, **Drugbank**, **DBPedia** *and* **LinkedCT**. *A clinical trial of the interventional type has studied interventions for certain conditions; intra-dataset links are used to represent these associations. An intervention may be linked to a drug through the owl:sameAs or rdfs:seeAlso property. Also, a disease may be linked to a condition through a owl:sameAs or rdfs:seeAlso inter-dataset link. Finally, a drug may have possible disease targets, and a disease may be treated with one or more drugs; these associations correspond to inter-dataset links. Some quality issues are: i) clinical trials support for the relationship between a drug and a possible disease target, ii) ambiguities between disease and condition labels, completeness and accuracy of owl:sameAs and rdfs:seeAlso links, and iii) the relationship of a disease and the possible drugs for its treatment is the inverse of the relationship between a drug and its possible disease targets. Suppose we want to study the quality of the links of all interventions in clinical trials that were done with drug Alemtuzumab, used in the treatment of chronic lymphocytic leukemia (CLL), cutaneous T-cell lymphoma (CTCL) and T-cell lymphoma, among others. In detail, we want to determine if all of the Alemtuzumab interventions are linked to the Alemtuzumab drug in Drugbank and DBPedia through owl:sameAs links. In LinkedCT circa September 2010, for the group of* 47 *Alemtuzumab interventions,* 21.28% *of these interventions do not have rdfs:seeAlso links to the Drugbank and*

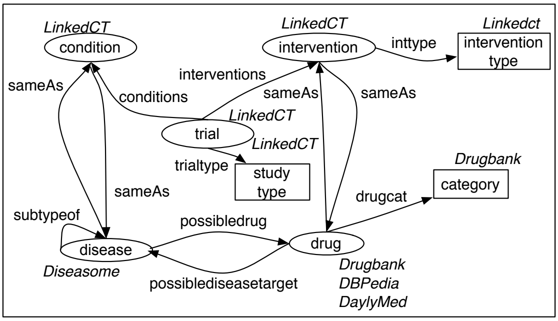


Fig. 1. Datasets in the Life Sciences

DBPedia datasets; this suggests that not all of the redundant Alemutuzumab interventions share the same links. Thus, tasks of pattern discovery or query answering may be affected by the lack of links between interventions and drugs.

Once again, the approach proposed by Hassas et. al [12] could provide support to detect some of these ambiguities and inconsistencies. First, appropriate string similarity metrics and similarity thresholds could be set up to identify that different interventions labelled with Alemtuzumab are duplicated. Additionally, in presence of misspellings of this name, also a service as the one provided by the Google spell checker API[5], could decide if the labels are wrongly spelled.

Nevertheless, consider two interventions in the LinkedCT dataset, one combines the drugs Alemtuzumab and Rituximab[6] while the other is just the intervention Rituximab[7]. Using a sub-string similarity metric and appropriate thresholds, the approach by Hassas et al. [12] would classify these two interventions as duplicates. However, this combination of drugs has been tested for specific types of immune system diseases and leukemia, while Rituximab has been also tested for a wide variety of other diseases that include different types of neoplasms, psoriatic arthritis, and scleritis. Additionally, the intervention that corresponds to the combination of Alemtuzumab and Rituximab is associated with the drug `drugbank:DB00087` in Drugbank that corresponds to Alemtuzumab, and the intervention Rituximab is associated with the drug `drugbank:DB00073` that is Rituximab. In consequence, these two interventions should not be considered the same even though their names are similar. We propose a Bayesian network based approach that models the entities of a knowledge domain, associations between entities as well as their conditional dependencies, and relies on a statistical inference process to infer the probability of duplicates and inconsistencies. In this particular case, our approach inference process will realize that these two interventions are associated with different drugs and conditions, and will infer a very low probability of duplicates; suggesting thus, they are not the same intervention.

3 Our Approach

In this section we present the LiQuate Bayesian network (LBN), a probabilistic model of RDF linked datasets, and the Process Model that represents the measure of link (in)completeness that can be encountered in the LOD cloud.

3.1 LiQuate Bayesian Network (LBN)

An LBN is a probabilistic model of a network of linked RDF datasets. It represents all the conditional dependencies among property subjects and objects in single RDF datasets and in linked datasets. The analysis of these dependencies is used to detect linked data quality problems.

[5] `https://code.google.com/p/google-api-spelling-java/`

[6] Intervention identifier `linkedct:resource/91d6bfb1b01cb7e26847ec38f3601f71`

[7] `linkedct:resource/cf01d4494c00870f75f757d2c8d1bcba`

The LBN model is based on the Bayesian network model developed for relational domains in [8] but adapted to the RDF data model. Getoor et al [8] apply Bayesian networks to the problem of imprecise estimates of the selectivity of a relational query; this framework is known as the Probabilistic Relational Model (PRM). This imprecision stems from the assumption of uniform distribution of values for attributes in a table, attribute independence in one table, and attribute independence in tables that are semantically related. Although an LBN resembles a PRM, its nodes and arcs have a particular semantics based on the linked RDF graph semantics. Nodes represent property object and subject values, and intra- and inter-links.

Definition 1 (LiQuate Bayesian network). *Given an RDF directed graph* $O_R = (V_R, E_R)$ *where* V_R*, and* E_R *are the nodes and arcs in the RDF graph. A* **LiQuate Bayesian network** R_B *for* O_R*, is a pair* $R_B = \langle O_B, CPT_B \rangle$*, where* $O_B = (V_B, E_B)$ *is a DAG.* V_B *are the nodes in* O_B*, and* E_B *are the arcs in* O_B*, and an homomorphism* $f : \mathbb{P}(E_R) \Rightarrow \mathbb{P}(V_B)$ *establishes a mapping between the power set of sets of graph edges (sets of triples) in* O_R *and sets of nodes in* O_B*.* CPT_B *are the Conditional Probability Tables for each node.*

There are three types of nodes:

1. **Value** nodes: `s-<property>` and `o-<property>` represent property subjects or objects in a single dataset. For example, node `o-hasintervention` represents the object values of interventions in clinical trials.
2. **Join** nodes: s-s-<pro$_1$>-<pro$_2$>, o-s-<pro$_1$>-<pro$_2$> and o-o-<pro$_1$>-<pro$_2$> correspond to boolean variables, and represent the matching of subjects or objects in related properties in a single dataset.
   For example, node `s-s-hascondition-hasintervention` represents the "join" over a trial, that is, a condition and an intervention are part of a trial.
3. **Link** nodes :b-<linkprop>-<typeres$_1$>-<typeres$_2$> corresponds to a boolean variable, and represents the existence of links among related resources. For example, node `b-sameas-condition-disease` represents the existence of *owl:sameAs* links among conditions in Clinical Trials (**LinkedCT**) and diseases in **Diseasome**.

The first two types of nodes represent data items and intra-dataset links [20], respectively, while the third type of nodes corresponds to inter-dataset links. Arcs represent dependencies between nodes. For each modeled dataset, there is a set of nodes annotated with the URI of the dataset.

Example 1. Suppose we want to study if the relationship that establishes that the drug *Paclitaxel* has as possible disease target *Leukemia*, is supported by at least one clinical trial. To achieve this goal we have built the LBN of Figure 2 with data from **LinkedCT**, **Diseasome**, **Drugbank** *circa* September 2010.

In this network, node `s-s-hascondition-hasintervention` represents the event that a condition and intervention are part of a clinical trial. This event is conditioned by the values of condition, intervention (nodes `o-hascondition` and `o-hasintervention`), by the existence of an *owl:sameAs* link among the condition and a disease, and among the intervention and a drug (nodes `b-sameAs-condition-disease` and `b-sameAs-intervention-drug`), and finally by a possible disease target

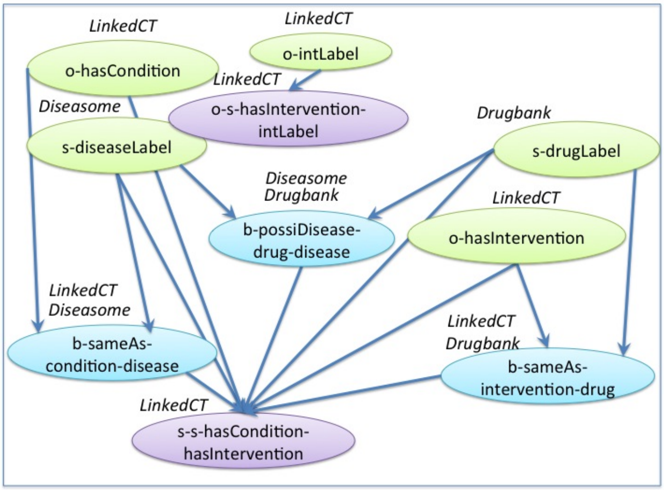


Fig. 2. LiQuate Bayesian network for the Life Sciences Domain

(node `b-possiblediseasetarget-drug-disease`. A query against the network may be a marginal posterior probability query where the marginal variable is `s-s-hascondition-hasintervention`, and where the evidence is set up according to the particular query. In this particular study, the query will check if given - the evidence - that the drug *Paclitaxel* has as possible disease target *Leukemia*, there is a clinical trial that backs this relationship with an equivalent - *owl:sameAs* - condition and intervention. The answer to this probability query is 1.0, so we can consider that this relationship is backed by at least one clinical trial. Tables 1(a) and 1(b) show a portion of the conditional probability tables (CPT) of this LBN.

Table 1. a) CPT o-hascondition; b) CPT s-s-hascond-hasint

(a)

o-hascondition	**prob(o-hascondition)**
Leukemia	0.00023736
Paget disease of bone, 602080	0.00047472
Pheochromocytoma, 171300	0.00047472
...	...

(b)

s-s-hascond-hasint	**o-hascondition**	**o-hasintervention**	**s-diseaselabel**	**s-druglabel**	**prob(s-s-hascond-hasint)**
false	asthma	f4d6498384f4f2b1becba117026f0d84	116	DB00043	0.0
false	prostate-cancer	5aedad01bdb8377fac9a76926e546309	960	DB01128	0.0
false	leukemia	94034c65834138c8ac01ef1bc18b4e37	673	DB01229	0.0
true	hypercholesterolemia	b6145d418fef2bf7f9d70ea844eff0f9	539	DB01076	1.0
true	prostate-cancer	5aedad01bdb8377fac9a76926e546309	3658	DB01128	1.0
true	leukemia	94034c65834138c8ac01ef1bc18b4e37	673	DB01229	1.0
...	...	...	...	...	...

Definition 2. *The homomorphism $f : \mathbb{P}(E_R) \rightarrow \mathbb{P}(V_B)$ establishes a mapping between O_R and O_B. f defines the set of nodes V_B as follows:*

$$f(\{(sub, pro, obj)\}) = \{s\text{-}pro, o\text{-}pro\} \tag{1}$$

$$f(\{(sub_1, pro_1, obj), (sub_2, pro_2, obj)\}) = \{o\text{-}o\text{-}pro_1\text{-}pro_2, o\text{-}o\text{-}pro_2\text{-}pro_1\} \tag{2}$$

$$f(\{(sub, pro_1, obj_1), (sub, pro_2, obj_2)\}) = \{s\text{-}s\text{-}pro_1\text{-}pro_2, s\text{-}s\text{-}pro_2\text{-}pro_1\} \tag{3}$$

$$f(\{(sub, pro_1, obj_1), (sub_2, pro_2, sub)\}) = \{s\text{-}o\text{-}pro_1\text{-}pro_2, o\text{-}s\text{-}pro_2\text{-}pro_1\} \tag{4}$$

$$f(\{(sub, linkprop, obj), (sub, rdf\text{:}type, typeres_1), (obj, rdf\text{:}type, typeres_2)\} = \tag{5}$$

$$\{b\text{-}linkprop\text{-}typeres_1\text{-}typeres_2\} \tag{6}$$

Definition 1 establishes *Value* nodes, definitions 2, 3 and 4 establish *Join* nodes, and definition 5 establishes *Link* nodes. The CPT_B are multidimensional histograms ordered by value. If a node v is a source node, the histogram will be one-dimensional, because in this case the CPT_B only represents the distribution of values taken up by the variable represented by the node.

The size of a CPT for one node depends on the number of predecessors, and on the number of possible values for the node and its predecessors. In our case the structure of the Bayesian network is related to the number of properties in the RDF datasets; in general, the RDF datasets are not complex in terms of their classes and properties. As to the set of possible values, these were aggregated in an indexed histogram that represents the CPT values.

Example 3. *Next, we illustrate the use of the homomorphism f. Figure 3 is a portion of the RDF graph (O_R) and its corresponding LBN graph (O_B).*

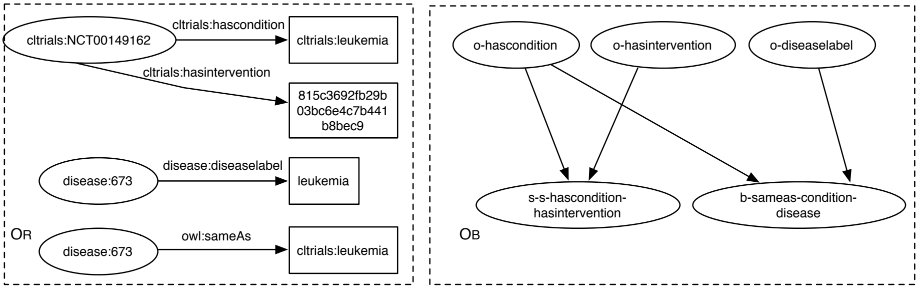


Fig. 3. Example Mapping RDF - Linked Bayesian Network

Definition 1 is applied to the sets of RDF arcs {`(NCT00149162, hasCondition, leukemia)`} and {`(673, diseaselabel, leukemia)`}:

f(`{(NCT00149162, hasCondition, leukemia)}`)=`{s-hascondition,o-hascondition}`

f(`{(673, diseaselabel, leukemia)}`)=`{s-diseaselabel,o-diseaselabel}`

Then, Definition 3 is applied to the set of RDF arcs {(NCT00149162,hasCondition, leukemia), (NCT00149162,hasIntervention,815c3692fb29b03bc6e4c7b441b8bec9)}

```
f({(NCT00149162,hasCondition,leukemia)),
   (NCT00149162,hasIntervention,815c3692fb29b03bc6e4c7b441b8bec9)})=
    {s-s-hascondition-hasintervention,s-s-hasintervention-hascondition}
```

Following, Definition 5 is applied:

```
f({(673,owl:sameAs,leukemia)),(673,rdf:type,Disease),
   (leukemia,rdf:type,Condition)})={b-sameas-condition-disease}
```

Intuitively, an LBN is semantically valid if its arcs have been established between nodes that map to properties whose subjects and objects are of the same type, i.e., have some type of matching instantiations, subject-subject, subject-object or object-object. Given the symmetry property of the combinations between triple patterns, for example the set V_B may contain only one of the nodes in the set {o-o-pro1-pro2, o-o-pro2-pro1} that has been defined with expression 2. Similarly, V_B may contain only one node in the sets defined with expressions 3 and 4; thus, the resulting LBN is minimal.

3.2 A Process-Based Model to Assess Link Completeness

Naumann et. al. [19] present several information quality dimensions which are categorized as: content-related, technical, intellectual, and instance-related. Our work is focused on content-related dimensions, in particular, on completeness of links. Additionally, information quality may be represented as a Data or as a Process model. LiQuate describes completeness of links as a process model; thus, the decision problem of link completeness is modeled as a workflow that detects possible missing links and proposes new links. We model two incompleteness problems: (1) link incompleteness and ambiguities between labels of resources, and (2) link incompleteness and ambiguities between sets of links.

Figure 4 presents the workflow for the first incompleteness problem. It represents the process that detects redundant resources that do not share the same set of links. First, the property subject or object redundancy is detected through probability queries to the property *value* node (either subject or object) or to a "join" node . Then, for each set of redundant values, conditional probability queries on the link, e.g. *owl:sameAs*, are generated, and the incompleteness of links is detected through these queries. Finally, links are added to incomplete redundant values and could be validated against the original data sources.

Figure 5 presents the workflow for the second incompleteness problem. This workflow represents the process used by LiQuate to detect incompleteness of links that occurs when certain relationships between sets of links do not hold. In this case, the query is a conditional probability query where the *Join* or *Link* node represent the marginal variable, and the evidence is set up according to the relationships between links that should hold. Once the incompleteness is detected, the user can intervene to set up other workflows that allow to detect the source of the inconsistency.

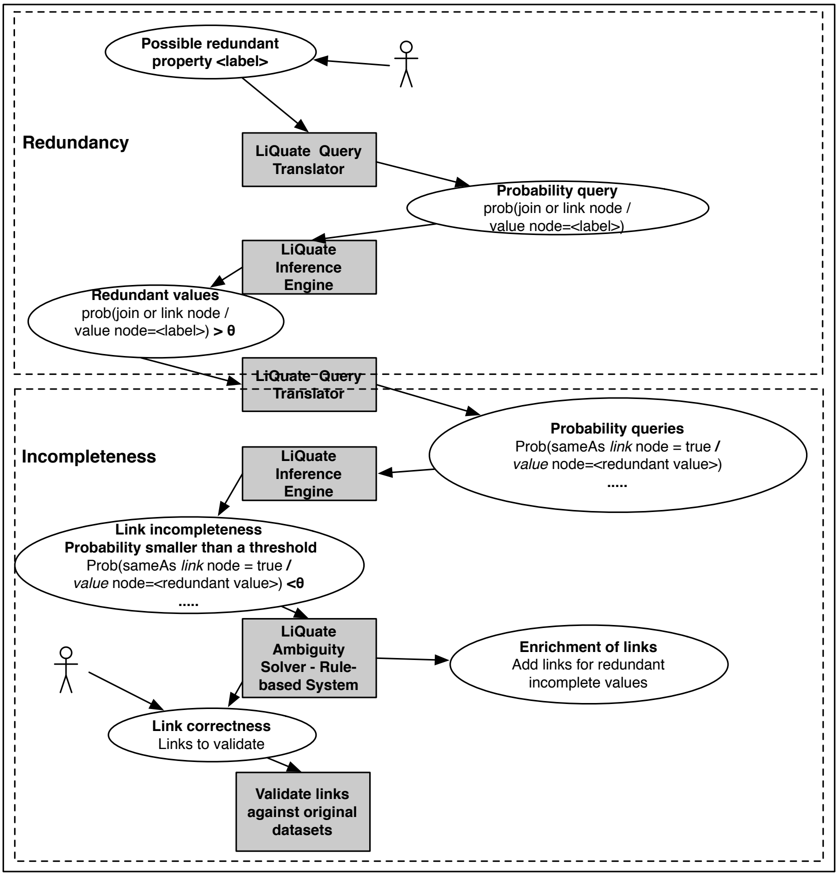

Possible redundant property <label>
Redundancy
LiQuate Query Translator
Probability query prob(join or link node / value node=<label>)
LiQuate Inference Engine
Redundant values prob(join or link node / value node=<label>) > θ
LiQuate Query Translator
Incompleteness
Probability queries Prob(sameAs link node = true / value node=<redundant value>) .....
LiQuate Inference Engine
Link incompleteness Probability smaller than a threshold Prob(sameAs link node = true / value node=<redundant value>) <θ .....
LiQuate Ambiguity Solver - Rule-based System
Enrichment of links Add links for redundant incomplete values
Link correctness Links to validate
Validate links against original datasets

Fig. 4. Workflow to Detect Incomplete Links and Uncontrolled Redundancy

4 The LiQuate System

In this section we describe the architecture of the LiQuate system, and the steps involved in constructing a LiQuate Bayesian network. We also describe Probabilistic Soft Logic (PSL) programs [15] which are part of the system.

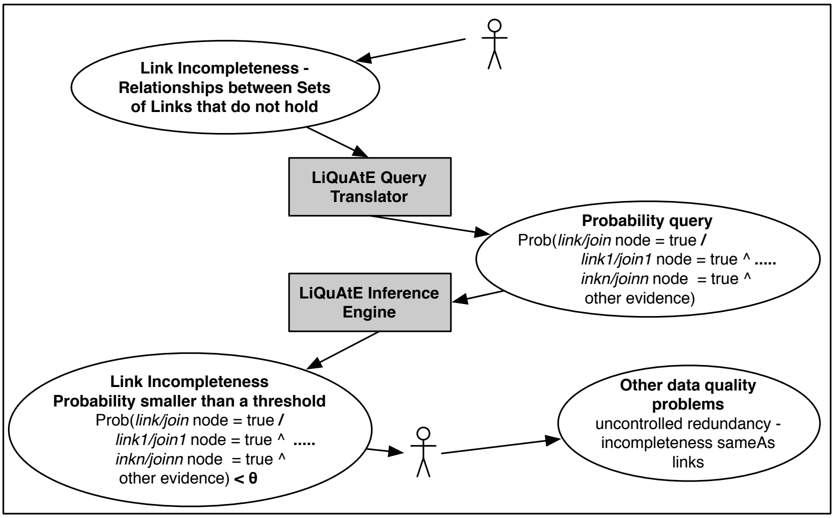

Link Incompleteness - Relationships between Sets of Links that do not hold
LiQuAtE Query Translator
Probability query
Prob(*link/join* node = true / *link1/join1* node = true ^ ..... *inkn/joinn* node = true ^ other evidence)
LiQuAtE Inference Engine
Link Incompleteness Probability smaller than a threshold
Prob(*link/join* node = true / *link1/join1* node = true ^ ..... *inkn/joinn* node = true ^ other evidence) < θ
Other data quality problems
uncontrolled redundancy - incompleteness sameAs links

Fig. 5. Workflow to Detect Incomplete Links w.r.t. Relationships Between Sets of Links

4.1 Architecture

Figure 6 shows the LiQuate architecture. LiQuate receives a *quality validation request* which is expressed as one or more evidence queries against the Bayesian network. The answer of a *quality validation request* is a number in the range [0.0:1.0] that indicates the probability that a given quality problem occurs among the data. Currently, three types of quality validation requests can be expressed: i) probability that labels or names of a given (type of) resource are redundant, ii) probability of incomplete links among a given set of resources, and iii) probability of inconsistent links. LiQuate follows a semi-automatic approach and is comprised of three components:

- The `Liquate Bayesian network Builder`: a semi-automatic off-line process that relies on an expert's knowledge about the properties in the RDF linked datasets that are going to be represented in the Bayesian network. It receives an RDF document and creates the LBN structure using mappings that establish the correspondence between the RDF graph and the nodes and arcs of the LBN structure. Once the LBN structure has been defined, relevant data is retrieved from SPARQL endpoints, and stored in a relational database to compute the histograms that implement the conditional probability tables (CPTs) associated with the nodes of the network; an aggregated CPT is generated for each node in the LBN structure. Both, the LBN structure and CPT's are fed to the *SamIam* network editor[8], and a Bayesian network is generated in one of *SamIam* internal formats.
- The `Ambiguity Detector`: is a probabilistic model that supports the analysis of the the linked data quality problems. The `Ambiguity Detector` is in turn comprised of three components: *1)* the `Quality Validation Request Analyzer`,

[8] `http://reasoning.cs.ucla.edu/samiam/help/recursiveconditioning.html`

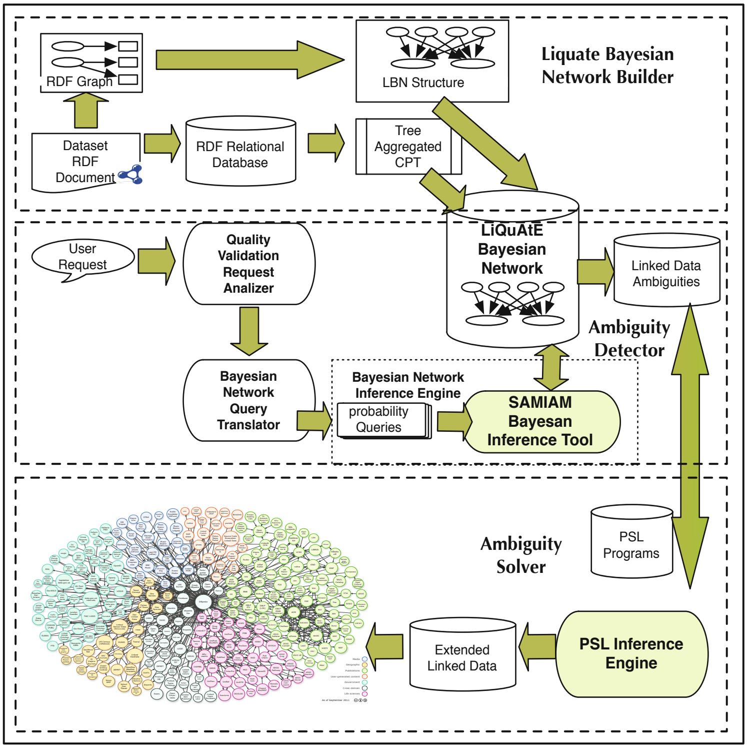


Fig. 6. The LiQuate Architecture

2) the `Bayesian network Query Translator`, and *3*) the `Bayesian network Inference Engine`.

The `Quality Validation Request Analyzer` receives a user request and determines if it can be satisfied with the existing Bayesian network.

The `Bayesian network Query Translator` considers the user request and generates the set of probability queries that must be posted against the Bayesian network. To this effect, SQL queries retrieve the marginal and evidence variables, values of interest, and probability queries are generated. It also gathers the answers of these queries and generates an answer to the user request.

The `Bayesian network Inference Engine` is responsible of performing the inference process required to answer each of the queries posted against the Bayesian network; this engine is implemented by the *SamIam* Bayesian Inference Tool. The Shenoy-Shafer exact inference algorithm [5] is used.

- The `Ambiguity Solver`: a probabilistic rule-based system that suggests new links for ambiguity resolution. It has been implemented on top of the Probabilistic Soft Logic tool (PSL)[9]. The PSL Inference Engine receives a set of weighted rules and

[9] `http://psl.umiacs.umd.edu/`

infers with a certain degree of uncertainty when two terms are duplicates, incomplete or incorrect. Currently, the degree of uncertainty ϕ is 0.5 but this value may be parametrized. PSL similarity metrics are implemented to decide when two labels or complex data items are similar. Inferred data quality problems between two terms are used to suggest RDF intra- and inter-links. The output of this component is the Extended Linked Data. This paper focuses on the `Ambiguity Detector` component.

4.2 LBN Construction

A **LiQuate Bayesian network** R_B corresponding to one or more linked RDF datasets, is a pair $R_B = \langle O_B, \mathrm{CPT}_B \rangle$. O_B, the LBN structure, is defined using the the mappings defined in Section 3. The intuition behind the mappings is that for each triple in the RDF dataset ($\{(sub, pro, obj)\}$), nodes `s-<property>` and `o-<property>` are created in O_B. For each pair of triples $\{(sub_1, prop_1, obj), (sub_2, prop_2, obj)\}$, where there exists a match on the object for the two properties, a node `o-o-<prop1>-<prop2>` is created in O_B. Similarly, nodes `o-s-<prop1>-<prop2>` and `s-s-<prop1>-<prop2>` may be created. Also, for each triple $\{(res_1, linkprop, res_2)\}$, a node `b-<linkprop>-<typeres1>-<typeres2>` is created, indicating a link among two types of resources, e.g., the *owl:sameAs* link among interventions and drugs

Arcs are created semi-automatically from "object" and "subject" nodes towards "join" and "link" nodes, and represent conditional dependencies among nodes. Arcs created by default include arcs from the non-matching "subject" or "object" nodes towards a "join" node, for example:

```
{(o-hascondition,s-s-hascondition-hasintervention),
 (o-hasintervention,s-s-hascondition-hasintervention)}
```

Similarly, arcs are created from "object" or "subject" nodes towards "link" nodes, for example:

```
{(o-hascondition,b-sameas-condition-disease),
 (s-diseaselabel,b-sameas-condition-disease)}
```

The CPT is represented as a variable width frequency histogram which aggregates values that have the same frequency into buckets, and the representative of each bucket is its highest value (endpoint). This type of histograms is useful when there is a small number of different probability values in the node's CPT, i.e., a large number of node values share the same probability value.

Figure 7 presents the workflow that is followed for the construction of the LBN CPTs. The steps are the following:

1. RDF datasets are loaded as vertically partitioned relational tables [1] (one table per property with subject and object columns).
2. A first version of the CPT for root nodes `s-<property>` and `o-<property>` is created in O_B through queries to the property tables which count and group the different object or subject values.

3. A first version of the CPT for link nodes, `b-<linkprop>-<typeres1>- <typeres2>` is created through queries to the link property tables which count and group the different linked resources.
4. A first version of the CPT for join nodes, `s-s-<property1>-<property2>` and other join nodes is created through join queries to the property tables and their parent property tables which count and group the parent object/subject values.
5. All of these initial CPT tables are ordered by their probability value, and the frequency histogram is generated. Some auxiliary data structures have been created in order to speed-up the lookup of the CPT histogram values: (1) the *Corr* structure that registers the correspondence of each CPT value with a sequential number, and (2) the *CptIndex* structure for non-root nodes, where for each sequential number representing a parent CPT value, there is a reference to the corresponding entry in the parent's aggregated histogram.

Figure 8 illustrates the auxilliary data structures used to build the CPT histograms. For both, root and non-root nodes, the initial CPT is ordered by probability value. Following the ordering given by the initial CPT, the *Corr* structure is built as a table of correspondence of each value (root nodes), or set of values (non-root nodes) with a sequence number. In this manner, we aggregate values that have the same probability into buckets, and the representative of each bucket is its highest sequence number (endpoint). For example, the root node has an endpoint of 2 for probability 0.2.

Additionally, there is a multi-level index, *cptIndex*, where for each parent of a non-root node, there is a reference to the endpoint of the bucket that corresponds to this parent's value. In this example, values $V1$ and $V2$ refer to the root node bucket with endpoint 2.

4.3 Probabilistic Soft Logic Programs

As indicated before, the **Linked Data Ambiguity Solver** is implemented as a Probabilistic Soft Logic [2] program, that is comprised by weighted rules of the form:

$$Body(X) \Rightarrow Head(Y);\ \phi$$

where, each rule is associated with a score ϕ greater than zero that represents the relative importance of the rule in the program. Additionally, *Body(X)* corresponds to a conjunction of predicates whose truth values can be in the interval [0,1]; soft predicates may correspond to similarity functions between explicit or implicit facts, while *Head(Y)* is a single predicate. Consider the following rule that illustrates the PSL semantics:

$$\begin{aligned} sameAs(X1,X2) \wedge conditionName(X1,C1) \wedge diseaselabel(X2,C2) \wedge \\ subType(X3,X2) \wedge diseaselabel(X3,C3) \wedge sameName(C1,C3) \Rightarrow \\ sameAsExtended(X1,X3). \end{aligned} \tag{7}$$

Rule 7 states that an *owl:sameAs* link will be added between a condition *X1* and a disease *X3*, depending on how similar are the names of *X3* and a more general disease *X2* that is already linked to *X1*.

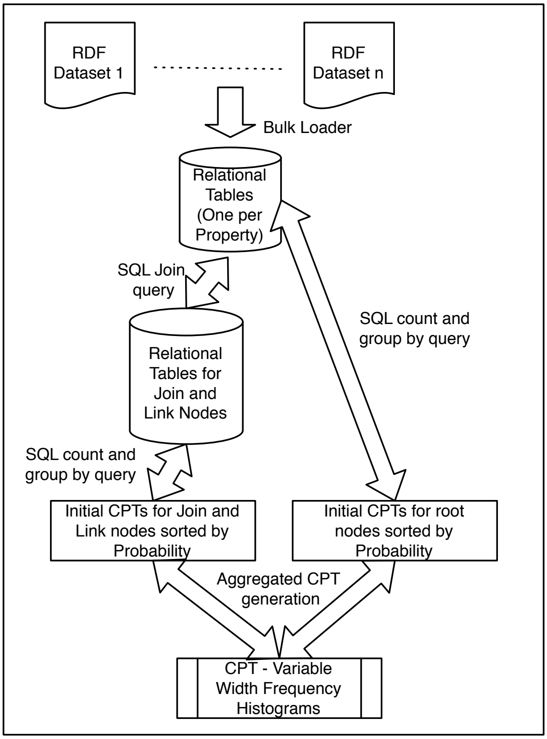

RDF Dataset 1
RDF Dataset n
Bulk Loader
Relational Tables (One per Property)
SQL Join query
Relational Tables for Join and Link Nodes
SQL count and group by query
SQL count and group by query
Initial CPTs for Join and Link nodes sorted by Probability
Initial CPTs for root nodes sorted by Probability
Aggregated CPT generation
CPT - Variable Width Frequency Histograms

Fig. 7. A Workflow to Build an LBN Given Several RDF Datasets

In the PSL engine [2], the truth degree of the *sameAsExtended(X1,X3)* corresponds to the Lukasiewicz t-norm of the truth values of the predicates *sameAs(X1,X2)*, *conditionName(X1,C1)*, *diseaselabel(X2,C2)*, *subType(X3,X2)*, *diseaselabel(X3,C3)* and *sameName(C1,C3)*, considering that the conjunction of two predicates is computed as the maximal value between 0 and the sum of the truth values of these predicates minus 1. In general the interpretation *In* of the logical operators conjunction $\wedge$, disjunction $\vee$, negation $\neg$, and $\Rightarrow$ is computed as follows:

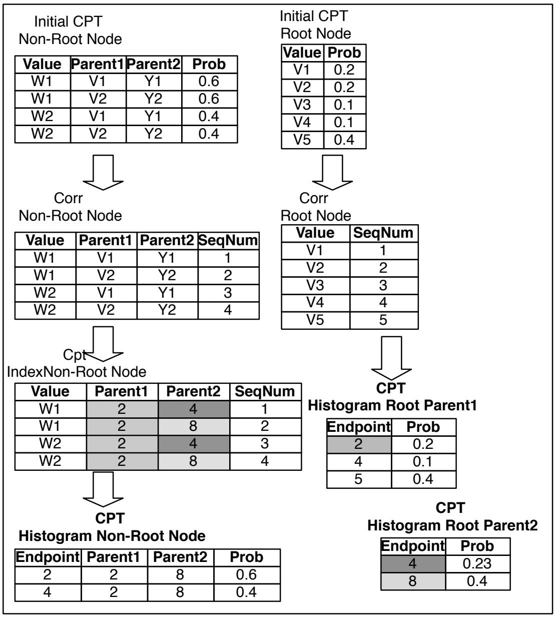
$$In(S_i \wedge S_j) = max\{0, In(S_i) + In(S_j) - 1\} \tag{8}$$

$$In(S_i \vee S_j) = min\{In(S_i) + In(S_j), 1\} \tag{9}$$

$$In(\neg S_i) = 1 - In(S_i) \tag{10}$$

$$In(S_i \Rightarrow S_j) = In(\neg S_i \vee S_j) \tag{11}$$

Once all the variables in a rule are instantiated, i.e., the rule is ground, the interpretation of the rule is computed according to rule 11. A rule is satisfied if and only if,



Initial CPT
Non-Root Node

Value	Parent1	Parent2	Prob
W1	V1	Y1	0.6
W1	V2	Y2	0.6
W2	V1	Y1	0.4
W2	V2	Y2	0.4

Initial CPT
Root Node

Value	Prob
V1	0.2
V2	0.2
V3	0.1
V4	0.1
V5	0.4

Corr
Non-Root Node

Value	Parent1	Parent2	SeqNum
W1	V1	Y1	1
W1	V2	Y2	2
W2	V1	Y1	3
W2	V2	Y2	4

Corr
Root Node

Value	SeqNum
V1	1
V2	2
V3	3
V4	4
V5	5

Cpt
IndexNon-Root Node

Value	Parent1	Parent2	SeqNum
W1	2	4	1
W1	2	8	2
W2	2	4	3
W2	2	8	4

CPT
Histogram Root Parent1

Endpoint	Prob
2	0.2
4	0.1
5	0.4

CPT
Histogram Non-Root Node

Endpoint	Parent1	Parent2	Prob
2	2	8	0.6
4	2	8	0.4

CPT
Histogram Root Parent2

Endpoint	Prob
4	0.23
8	0.4

Fig. 8. LBN Auxiliary Structures for CPT Histograms

the $In(S_i) \leq In(S_j)$, i.e., the truth value of the head ($In(S_j)$) is at least the same truth value than the body($In(S_i)$). Note that the interpretation of a PSL rule does not coincide with the traditional interpretation of Horn clauses implemented by the refutation inference rule performed in Programming Logic languages as Prolog or Datalog [3]. Given the truth values of a rule r under an interpretation In, the distance to satisfaction of $In(r)$, $d(In(r))$, is defined as how far are the truth values of $In(r)$ to 1, i.e.,

$$d_r(In) = max\{0, In(S_i) - In(S_j)\} \tag{12}$$

Finally, an interpretation In is a model of a PSL program P, if In is the interpretation that satisfies with the highest probability, the majority of the ground rules of P. Given the interpretation In, the probability of satisfaction of a rule r weighted with the score $\phi(r)$ is computed as $\phi(r) \times In(r)$.

It is important to notice that the conditional probabilistic approach is used to suggest possible ambiguities in linked datasets, while the PSL approach is just used to suggest new links for ambiguity resolution. The t-norms are performed by the PSL engine in conjunction with optimization techniques to identify the optimal model for a PSL program; and they cannot be configured or selected. Nevertheless, PSL allows to infer with a certain degree of uncertainty if a given new link can be included to resolve an ambiguity identified by the `Ambiguity Detector`, and this degree can be configured.

Thus users can decide appropriate level uncertainty for a given domain. Additionally, in the current version of LiQuate, the `Ambiguity Solver` does not perform any prediction task; it just uses information inferred from the Bayesian network to suggest with a certain degree of uncertainty, a link between two possible redundant concepts. However, it is important to highlight that PSL features can be also exploited to implement prediction techniques that rely on graph analysis algorithms, to determine the density of the linked datasets and based on this, suggest potential missing links. This problem is out of the scope of this paper.

5 Quality Process Workflows for the Life Sciences

As a proof of concept, we model two quality problems in the Life Sciences domain; for each problem, different hypotheses are considered.

1. **Incompleteness** *owl:sameAs* **Links and Uncontrolled Redundancy of Diseases, Drugs, Conditions or Interventions.** This quality problem consists in having uncontrolled redundancy of these entities, i.e., exactly the same name or label, but different ids, and (in)completeness of *owl:sameAs* condition-disease and intervention-drug links. If these type of data items are linked to or from other data items, it is possible that portions of the redundant labels are not being considered in the links, and in consequence, ignored by the link prediction or pattern discovery tools such as the one described in [22]. It should be noted that once the problem is corrected, there will still be redundancy, but it will be controlled. For example, if diseases in **Diseasome** are linked using *owl:sameAs* links, to conditions in **LinkedCT**, then all of the diseases that are redundant, i.e., have the same name, should be linked to the same conditions. This is the case of *Leukemia, acute lymphoblastic* in **LinkedCT** *circa* September 2011, where identical labels appear for ids 2908, 2909, and 2910. The query is posed to the LiQuate Bayesian network requiring the probability of occurrence of the disease label. It is of the form:

   ```
   prob(o-diseaselabel=<label>)
   ```

   If there are N distinct object labels, and the probability is greater than $1/N$, then the label is redundant. In this case, the query generator will produce probability queries of the form:

   ```
   prob(b-sameas-condition-disease=true/s-diseaselabel=<diseaseid>)
   ```

   for all the ids of the duplicate label, Again, If the probability is less than a threshold, the condition has not been linked to a disease.
2. **Incompleteness - Lack of disease target support by an intervention (drug) in a clinical trial.** We assume that the relationship between a drug that targets a disease should be supported by at least one clinical trial. In the dataset **LinkedCT** *circa* September 2011, *Leukemia, acute myeloid, 601626* has a possible drug treatment, *Sorafenib*, for ids 2921 and 2922. This disease has no *owl:sameAs* link to a

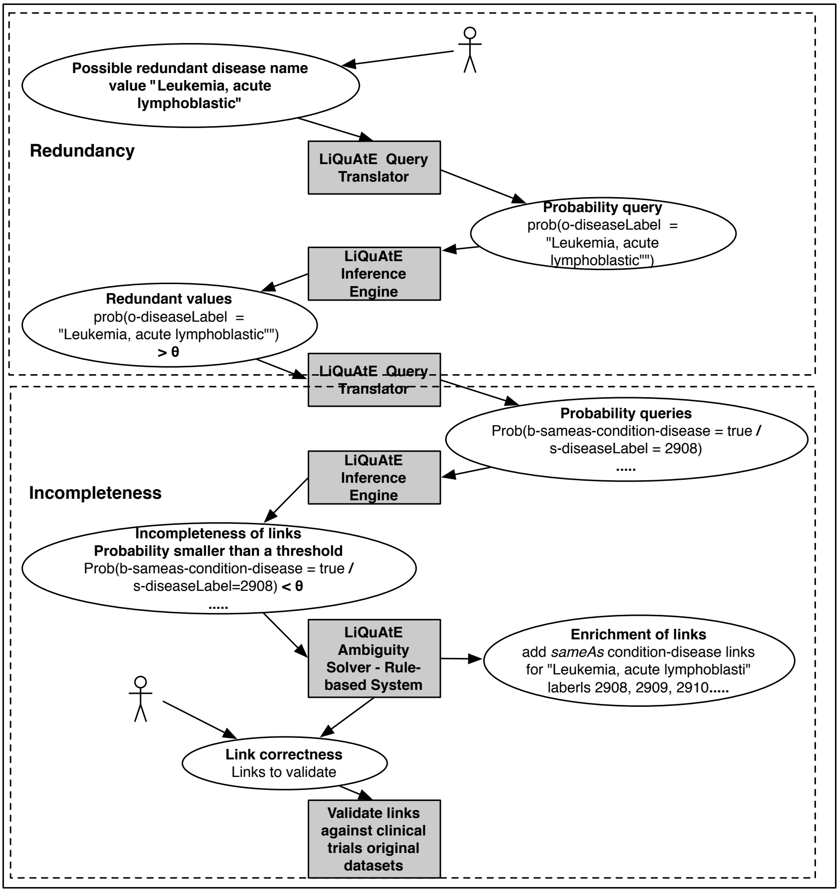

Possible redundant disease name value "Leukemia, acute lymphoblastic"
Redundancy
LiQuAtE Query Translator
Probability query prob(o-diseaseLabel = "Leukemia, acute lymphoblastic"")
LiQuAtE Inference Engine
Redundant values prob(o-diseaseLabel = "Leukemia, acute lymphoblastic"") > θ
LiQuAtE Query Translator
Probability queries Prob(b-sameas-condition-disease = true / s-diseaseLabel = 2908) .....
LiQuAtE Inference Engine
Incompleteness
Incompleteness of links Probability smaller than a threshold Prob(b-sameas-condition-disease = true / s-diseaseLabel=2908) < θ .....
LiQuAtE Ambiguity Solver - Rule-based System
Enrichment of links add sameAs condition-disease links for "Leukemia, acute lymphoblasti" laberls 2908, 2909, 2910.....
Link correctness Links to validate
Validate links against clinical trials original datasets

Fig. 9. Workflow to Detect Incompleteness *owl:sameAs* links and uncontrolled redundancy of diseases in clinical trials of **LinkedCT** *circa* September 2011

condition in clinical trials, so it is not possible to determine the support by clinical trials (there are few *owl:sameAs* links, 497 for conditions and diseases in general), even in the case where there is the corresponding clinical trial. This problem could also arise when there are missing *owl:sameAs* links for interventions and drugs. Furthermore, it could be the case that there is really no trial that supports the relationship among the drug and its possible disease target.

To detect this problem, several probability queries may be formulated to the LBN. The marginal variable corresponds to the `s-s-hascondition-hasintervention` node which represents that there is a matching trial (subject in the RDF triple) for properties condition and intervention; the evidence may be set up with the drug and

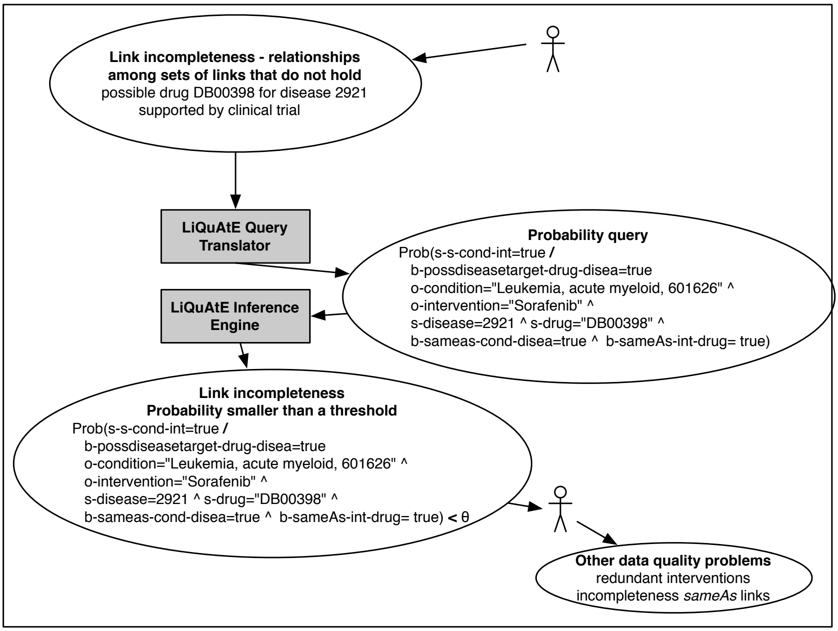

Link incompleteness - relationships among sets of links that do not hold
possible drug DB00398 for disease 2921 supported by clinical trial
LiQuAtE Query Translator
LiQuAtE Inference Engine
Probability query
Prob(s-s-cond-int=true / b-possdiseasetarget-drug-disea=true o-condition="Leukemia, acute myeloid, 601626" ^ o-intervention="Sorafenib" ^ s-disease=2921 ^ s-drug="DB00398" ^ b-sameas-cond-disea=true ^ b-sameAs-int-drug= true)
Link incompleteness
Probability smaller than a threshold
Prob(s-s-cond-int=true / b-possdiseasetarget-drug-disea=true o-condition="Leukemia, acute myeloid, 601626" ^ o-intervention="Sorafenib" ^ s-disease=2921 ^ s-drug="DB00398" ^ b-sameas-cond-disea=true ^ b-sameAs-int-drug= true) < θ
Other data quality problems
redundant interventions
incompleteness sameAs links

Fig. 10. Workflow to Detect Incompleteness - lack of disease target support by drug (intervention) in clinical trials of **LinkedCT** *circa* September 2011

disease, and the corresponding intervention and condition. The user may also set up the evidence on the existence of *owl:sameAs* links. The steps of this workflow are illustrated in Figure 10.

It should be mentioned that this disease has twelve redundant labels, but only two of them have Sorafenib as possible drug treatment; thus, there is a "possible drug" link incompleteness.

In this work, quality problems have been studied for a portion of linked datasets in the Life Sciences domain. The evidence is set for each query according to the particular data problem that needs to be studied; a subset of the parents of the node that represents the marginal variable may be set as evidence. Table 2 lists some of the quality problems that have been studied for a portion of linked datasets in the Life Sciences domain. The evidence is set for each query according to the particular data problem that needs to be studied; a subset of the parents of the node that represents the marginal variable may be set as evidence. For example, if the goal is to detect the probability of the existence of an *owl:sameAs* link among a certain condition and any disease, the probability query is `prob(b-sameas-condition-disease/o-hasCondition=<condition>)`; in this case the evidence on the disease has not been set.

Table 2. Quality Problems and Probability Queries for Life Sciences Linked Datasets-A Use Case

Data Quality Problem	**Probability Query**
Redundant disease labels	`prob(o-diseaselabel=<value>)`
Redundant condition labels	`prob(o--s-hasCondition-conditionLabel / o-conditionLabel = <label>)`
Redundant drug labels	`prob(o-druglabel=<value>)`
Redundant intervention labels	`prob(o--s-hasIntervention-intLabel / o-intLabel = <label>`
Existence of *owl:sameAs* among condition and disease	`prob(b-sameas-condition-disease/<evidence parents>)`
Existence of *owl:sameAs* among intervention and drug	`prob(b-sameas-intervention-drug/<evidence parents>)`
Existence of trial for properties condition and intervention	`prob(s-s-hascondition-hasintervention/<evidence parents>)`
Existence of possible disease target	`prob(b-possdiseasetarget-drug-disease/<evidence parents>)`
Existence of possible drug	`prob(b-possdrug-disease-drug/<evidence parents>)`

6 Experimental Study

The purpose of the experimental study conducted in the Life Sciences domain is two-fold: (1) evaluate the datasets in general, in order to detect quality problems related to link incompleteness, and (2) evaluate the quality of particular scenarios. Two linked data sets which are published at the `LinkedCT.org` website [11], were considered: (1) *Datasets 1*: **LinkedCT**, **Diseasome** and **Drugbank** *circa* September 2010, and (2) *Dataset 2*: **LinkedCT**, **Diseasome**, **Drugbank**, and **DBPedia** *circa* September 2011.

We use the Bayesian inference tool, *SamIam* to build the LBN based on the mappings defined in Section 3. Bayesian inference queries are posed to the network through the *SamIam* tool, and one of the algorithms implemented by *SamIam*, the Shenoy-Shafer exact inference algorithm [5] is used.

The rule-based component was built on top of PSL. The following studies were conducted during our evaluation:

1. **Redundant entities (conditions, interventions, diseases, and drugs), and incompleteness in** *sameAs/seeAlso* **links**.
   Hypothesis 1: All redundant entities, i.e., entities with same labels, share the same properties, in particular they have the same *sameAs/seeAlso* links.

   Similarly to the workflow that describes a specific case, this experiment evaluates the whole dataset, i.e., the redundant conditions, interventions, drugs and diseases. Information that is stored in the LBN CPT structure allows us to retrieve the percentage of labels in each dataset that are redundant. Then, for all the redundant labels, the LiQuate query generator produces LBN probability queries for drugs and interventions and for conditions and diseases, e.g.:
   `prob(b-sameas-condition-disease/s-diseaselabel=<diseaseid>)`.

   For each dataset, we partition the entities according to their label, e.g., diseases are partitioned according to their name. The metrics that are considered in the experimental study are described in Table 3.

   The results for *Dataset 1* are presented in Table 4. For these datasets, experiments on conditions and diseases were executed also with the extended version of the *owl:sameAs* links among conditions and diseases. The following set of rules were used to extend *owl:sameAs* links between diseases and conditions. The predicate *sameName* represents a string similarity function and suggests how similar are

Table 3. Metrics of Experimental Study

Metric	**Formula**
% of redundant partitions	# partitions size > 1/#*partitions*
% of redundant labels with *owl:sameAs* (resp. *rdfs:seeAlso*) links	# labels in partitions w/sameAs / # labels in partitions
% partitions with at least one *owl:sameAs* (resp. *rdfs:seeAlso*) link	# partitions with at least one sameAs / # partitions
% partitions with one *owl:sameAs* (resp. *rdfs:seeAlso*) link	# partitions with one sameAs / # partitions
% partitions with all labels with *owl:sameAs*(resp. *rdfs:seeAlso*) links	# partitions with all labels w/sameAs / # partitions

two strings; a value close to 0.0 suggests that the strings are different while a value close to 1.0 means that the strings are the same.

(1) sameAs(X1,X2) & conditionName(X1,C1) & diseaselabel(X2,C2) &
subType(X2,X3) & diseaselabel(X3,C3) & sameName(C1,C3)
⇒ extendedSameAs(X1,X3)
(2) conditionName(X1,C1) & diseaselabel(X2,C2) & sameName(C1,C2)
⇒ extendedSameAs(C1,C2)

Rule (1) extends *owl:sameAs* links between conditions and diseases, with super types of the diseases that have similar names to the condition name. Rule (2) combines conditions and diseases that have similar names. Two similarity metrics were used to implement the predicate *sameName*: Levenshtein and Jaro-Winkler [4]. PSL was used to implement the rules and the probabilistic model that computes the uncertainty degree of the membership of a pair condition-disease, to *extendedSameAs*. Currently, the degree of uncertainty for suggested links is 0.5, but this value may be parametrized. We analyzed the quality of the inferred links *by hand* to ensure that only valid links where included in the set of *extendedSameAs* when the degree of 0.5 was considered.

Discussion. For conditions and diseases, less than 11.6% of partitions of redundant labels have *owl:sameAs* "original" links; this situation improves for interventions with 26.2%. This is due to the fact that the percentage of *owl:sameAs* condition-

Table 4. *Dataset 1*: Redundant Conditions and Diseases (Diseasome), Interventions and Drugs (Drugbank), and sameAs Completeness. Redundancy is measured in terms of % of redundant partitions, % of redundant labels with *owl:sameAs* links, % partitions with at least one *owl:sameAs* link, % partitions with one *owl:sameAs* link, and % partitions with all labels with *owl:sameAs* links. Interesting values are highlighted in **bold**.

Property	**% redundant partitions**	**% redundant labels w/sameAs**	**% redundant partitions w/sameAs**	**%redundant partitions w/one label w/sameAs**	**%redundant partitions w/all labels w/sameAs**
Conditions and Diseases					
condition w/sameAs	3.49	11.5	10.7	4.7	0
condition w/sameAs Ext.	3.49	12.4	11.0	7.6	0.19
disease w/sameAs	14.2	19.8	**11.6**	**6.1**	**5.9**
disease w/sameAsExt	14.2	55.8	23.3	23.16	**23.16**
Interventions and Drugs					
intervention w/sameAs	**17.8**	89.9	**26.2**	**12.3**	1.21
drug w/sameAs	0	–	–	–	–

disease links is very small with respect to the total number of conditions and diseases, 1.3% and 5.6%, whereas for interventions it is 11.18%. The other metrics also suggest the poor quality of the original links: the percentage of partitions with over one link is in the best of cases, interventions, only 12.3%, and the percentage of partitions with links for all its members is at most 5.9% for diseases. When the links are enriched using the LiQuate ambiguity solver, there is a noticeable improvement of the quality of links for diseases, from 6.1 to 23.16, not so for conditions. The explanation of this lies in the proportion of condition links, it has barely increased to 1.4%.

Results for *Dataset 2* are presented in Table 5.

Table 5. *Dataset 2*: Redundant Interventions and Drugs (Drugbank), and sameAs/seeAlso Completeness. Redundancy is measured in terms of % of redundant partitions, % of redundant labels with *sameAs* or *seeAlso* links, % partitions with at least one *sameAs* or *seeAlso* link, % partitions with one *sameAs* or *seeAlso* link, and % partitions with all labels with *sameAs* or *seeAlso* links. Interesting values are highlighted in **bold**.

Property	**% redundant partitions**	**% labels w/sameAs-seeAlso**	**% redundant partitions w/sameAs-w/seeAlso**	**%redundant partitions w/one label w/sameAs-seeAlso**	**%redundant partitions w/all labels w/sameAs-seeAlso**
Diseases and Interventions					
disease w/sameAs	11.88	14.24	14.78	0	**14.78**
intervention w/sameAs	**15.74**	22.82	9.47	**5.64**	3.23
disease w/seeAlso	**11.88**	14.44	15.01	0	15.01
intervention w/seeAlso	**15.74**	25.59	9.56	0	**9.56**

Discussion. For diseases, 11.88% of partitions have a size > 1, i.e. are redundant. For interventions, this number is slightly higher, 15.74%. For diseases, all of the labels in 14.78% of the redundant partitions have *sameAs* links, whereas in the case of interventions, 5.64% of partitions have only one *sameAs* link, and all of the labels in 9.56% of the redundant partitions have *seeAlso* links. These results suggest the incompleteness of *sameAs* and *seeAlso* links to the Drugbank dataset in the case of interventions. It should be noted that the percentage of redundant conditions is only 0.07. Although this suggests that the redundancy is low, the number of *seeAlso* and *sameAs* links among conditions in trials and diseases in Diseasome is roughly 2.3% of the total number of distinct conditions. Additionally, when comparing both datasets (2010 and 2011), we can see that the percentage of redundant partitions for interventions decreases, from 17.8 to 15.74.

2. **Possible Disease Target Support by an Intervention (Drug) in a Clinical Trial**. **Hypothesis 2:** Diseases targeted by a drug with a link *possibleDiseaseTarget*, are supported by at least one clinical trial with a condition and drug intervention.

   This experiment was conducted on *Dataset 1*. Approximately 10,000 probability queries were generated for each drug and disease and all the combinations of linked (through *owl:sameAs*) conditions and interventions. The marginal node is `s-s-hascondition-hasintervention`, and the evidence is a disease, drug, condition, intervention and the existence of *owl:sameAs* links among them. The result is that 13,5% of the drugs and targeted diseases are supported by clinical trials. This experiment was executed again with the enrichment of the *owl:sameAs* links generated by the LiQuate ambiguity solver; in this case, approximately 11, 400 queries were generated; the result is that there is 13, 9% of disease targets that are supported by clinical trials. Similarly, another hypothesis is that drugs that can possibly treat diseases (*possibleDrug* links) are supported by the same percentage of clinical trials than possible disease targets. The result is 13, 5%. This result suggested that both links *possibleDiseaseTarget* and *possibleDrug* are the inverse of each other. We confirmed this result by querying the RDF datasets.

Discussion. The percentage of possible diseases targeted by drugs that are supported by clinical trials suggests the incompleteness of links. This situation does not improve greatly when the study is performed with the extended links. This is because the coverage of conditions in this extended version is still very low.

3. **Redundant interventions and completeness of** *owl:sameAs* **and** *rdfs:seeAlso* **links (specific scenario)**.

   Hypothesis 3: For a certain group of drugs used in the treatment of different types of cancer, all of the redundant labels are linked to the same entities. This experiment was conducted on datasets *Dataset1* and *Dataset 2*. 1, 186 probability queries were generated for all redundant labels:

   ```
   prob(b-sameas-intervention-drug=true/o-hasintervention=<interventionid>)
   ```

   Similarly, probability queries were issued for *owl:sameAs* links to DBPedia, and *rdfs:seeAlso* links for both Drugbank and DBPedia. The results can be observed in Table 6.

Table 6. *Dataset 1 and Dataset2*: Redundant Interventions - sameAs and seeAlso Completeness. Redundancy and Incompleteness are measured in terms of # duplicates; **% prob** *sameAs* > 0: probability of having sameAs links and **% prob** *seeAlso* > 0: probability of having seeAlso links. Interesting results are highlighted in **bold**.

Intervention	**# redundant labels**	**% prob** *sameAs* > 0 **Drugbank**	**% prob** *seeAlso* > 0 **Drugbank**	**% prob** *sameAs* > 0 **DBPedia**	**% prob** *seeAlso* > 0 **DBPedia**
Alemtuzumab	47	78.72	100	78.72	100
Bevacizumab	295	88.47	100	88.47	100
Brentuximab vedotin	2	**0**	**0**	**0**	**0**
Catumaxomab	4	**0**	**0**	100	100
Cetuximab	162	78.4	100	78.4	100
Ipilimumab	51	**0**	**0**	49.02	100
Ofatumumab	39	**0**	**0**	64.10	100
Panitumumab	48	79.10	100	79.10	100*s*
Rituximab	283	82.33	100	82.33	100
Trastuzumab	58	74.14	100	74.14	100
dexamethasone	112	77.68	100	77.68	100
doxycycine	18	94.44	100	0	0
exemestame	24	91.67	100	91.67	100
haloperidol	11	90.91	100	90.91	100
mercatopurine	17	76.47	100	76.47	100
tamoxifen	13	**100**	**100**	**100**	**100**

Discussion. In *Dataset 2*, we can observe that for all of these interventions (except tamoxifen) there is a percentage of redundant labels that is not linked with *owl:sameAs* or *rdfs:seeAlso*. We can distinguish several cases:

- A percentage of redundant labels is not linked through *owl:sameAs* to neither Drugbank or DBPedia, but 100% of the labels are linked through *rdfs:seeAlso*, e.g., Bevacizumab.
- None of the redundant labels are linked to Drugbank or DBPedia, e.g., Brentuximab vedotin. In this case, the drug is not present in Drugbank .
- All of the redundant labels are linked to DBPedia and none to Drugbank, e.g., Catumaxomab.
- A percentage of redundant labels is linked to DBPedia through *owl:sameAs*, all of them are linked to DBPedia through *rdfs:seeAlso* and none to Drugbank, e.g., Ipilimumab.
- Almost all redundant labels are linked to Drugbank and none to DBPedia, e.g., doxycycine.

These results suggest in general the incompleteness of links. We can conclude that *rdfs:seeAlso* links are more complete than *owl:sameAs* links, and that links to DBPedia are more complete than links to Drugbank. It should be mentioned that we have further explored the cases where the same percentage of labels link through *owl:sameAs* to Drugbank and DBPedia, and exactly the same labels are linked to both datasets. Additionally, in cases of interventions like Brentuximab vedotin, where there are no links, although the drug is not registered in Drugbank, it does appear in DBPedia.

7 Related Work

The publication of clinical trials as linked RDF data is described in [11]. In this work, the authors emphasize the challenges of linking resources in the trials data, and linking different datasets. The authors demonstrated how state-of-the-art approximate string matching and ontology-based semantic matching can be used for discovery of such semantic links between several data sources. Differently from our work, the emphasis is on link discovery while LiQuate's focus is on the detection of linked data quality problems and the enrichment of the links among the data.

The framework xCurator proposed by Hassas et al [12] aims to produce high quality linked RDF data from semi-structured sources where unique URIs are generated, duplicates are merged, resources are linked to vocabularies using entity type extraction techniques, and links are established to external RDF sources. The framework was applied to clinical trial and to bibliographic data, and the quality of the data was improved with respect to previous transformations that were done manually. As illustrated in Section 2, xCurator relies on a set of ontology matching techniques, string similarity metrics and word spell checker services to detect duplicates, link incompleteness and inconsistencies. Nevertheless, knowledge of this domain is not considered during the linking and cleaning process, and some ambiguities may be wrongly identified. Contrary, LiQuate exploits semantics encoded in the Bayesian network during a statistical inference process, and is able to suggest possible ambiguities and inconsistencies not only by considering the names of the entities, but also by looking at the different entities that are related to the studied entities.

In [14], the applicability and benefits of using linked data in the Life Sciences domain is studied, specifically for clinical trials, drugs, and related sources. The authors present several challenges and among them, the need of progress in finding links between data items where no commonly used identifiers exist, and the need to develop techniques for record linkage and duplicate detection with methods from the database and knowledge representation communities. The described challenges summarize some of the data quality problems that we detected in our experimental study.

The Silk Linking Framework [24] is focused on the discovery of links by establishing linkage rules that include similarity functions. Our approach is not focused on discovering new links, but on detecting link incompleteness, and proposing the enrichment of the datasets, with triples that represent the ambiguities that have been found.

Demartini et. al. [6] develop a probabilistic framework in combination with crowdsourcing techniques, in order to improve the quality of links in the LOD cloud. The system, *ZenCrowd* combines algorithmic and manual matching techniques to link entities. It exploits probabilistic models using factor-graphs to represent probabilistic variables. This approach is based on evaluating several alternative links, whereas our system proposes evaluating the quality of the current links in some specific domain.

Network approximate measures are used to analyze the quality of linked data in [9] using the LINK-QA framework. An original local network and extended networks are constructed around resources to be evaluated by querying the Web of Data. Five metrics are used. degree, clustering coefficient, number of *owl:sameAs* chains, centrality and richness of description. Finally, Fürber et. al. [7] propose a conceptual model for data quality management that allows to formulate a set of data quality and data cleaning rules, classification of data quality problems, and the computation of quality scores for Semantic Web data sources. The authors present several use cases and competency questions, but these are all related to the quality of the classes, properties and instances on the datasets, but not in the consistency or quality of the links. A set of quality and cleaning rules is established, but none of these refer to links among datasets.

Memory et. al. [18] present a work on summarization of annotation graphs where PSL is the framework used. The work integrates the multiple types of evidence from the annotation links, the various similarity metrics, and two graph summarization heuristics: a similarity heuristic and a summarization heuristic within a probabilistic model, using PSL. This approach uses PSL to model the summarization graph whereas in our work, a PSL system is used to propose additional links with a certain degree of uncertainty.

8 Conclusions and Future Work

LiQuate is a semi-automatic tool that supports the process of detecting data quality problems and enriching linked data, recommending a set of additional links. Although a large number of triples and links is available in the LOD cloud, the data is not reliable. Initial experiments in the Life Sciences domain show link incompleteness in the *possibleDiseaseTarget* and *possibleDrug* links together with their corresponding trials data, incompleteness in the *owl:sameAs* links, and uncontrolled redundancy in conditions, diseases, and interventions. LiQuate, a combination of Bayesian networks and a rule-based system, may be used to detect incompleteness and extend the linked datasets in order to improve the quality of links using the semantics of the relationships among the RDF concepts, encoded in the LBN.

The focus of the conducted evaluation study was the analysis of the different quality problems presented in the studied datasets, i.e., the quality of the `Ambiguity Detector` component. In this current version, we did not conduct a formal evaluation of the quality of the links inferred by the `Ambiguity Solver`; we just analyzed the inferred links *by hand* to ensure that only valid links where included in both sets of *extendedSameAs* and *extendedSeeAlso*. Thus, future work includes validating the links that have been generated, developing additional rules to produce different sets of extended *owl:sameAs* links, and extending the reported experimental study with these sets of links.

Provenance information of the datasets will be taken into account in order to not only enrich the datasets with the solution to the quality problems, but to be able to modify them.

Furthermore, the generation of the CPT tables for the LBN can be developed using the RDFStats [16] statistics generator, or use their API for accessing statistics including several estimation functions that also support SPARQL filter-like expressions. Finally, experimental studies will be developed for other domains in order to do a thorough evaluation of the system and validate the proposed links. Particularly, we plan to validate links to terms of Geonames[10].

References

1. Abadi, D.J., Marcus, A., Madden, S.R., Hollenbach, K.: Scalable semantic web data management using vertical partitioning. In: Proceedings of VLDB 2007 (2007)
2. Broecheler, M., Mihalkova, L., Getoor, L.: Probabilistic similarity logic. In: Conference on Uncertainty in Artificial Intelligence (2010)
3. Ceri, S., Gottlob, G., Tanga, L.: What you always wanted to know about datalog (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering 1(1) (1989)
4. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb, pp. 73–78 (2003)
5. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
6. Demartini, G., Difallah, D.E., Cudré-Mauroux, P.: Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: WWW (2012)
7. Fürber, C., Hepp, M.: Towards a vocabulary for data quality management in semantic web architectures. In: EDBT/ICDT Workshop on Linked Web Data Management (2011)
8. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. SIGMOD Record 30(2), 461–472 (2001)
9. Guret, C., Groth, P., Stadler, C., Lehmann, J.: Linked data quality assessment through network analysis. In: ISWC 2011 Posters and Demos (2011)

[10] `http://www.geonames.org/`

10. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl: sameas isn't the same: An analysis of identity in linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 305–320. Springer, Heidelberg (2010)
11. Hassanzadeh, O., Kementsietsidis, A., Lim, L., Miller, R.J., Wang, M.: Linkedct: A linked data space for clinical trials. CoRR, abs/0908.0567 (2009)
12. Hassanzadeh, O., Yeganeh, S.H., Miller, R.J.: Linking semistructured data on the web. In: WebDB (2011)
13. Isele, R., Jentzsch, A., Bizer, C.: Silk server - adding missing links while consuming linked data. In: 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai (2010)
14. Jentzsch, A., Andersson, B., Hassanzadeh, O., Stephens, S., Bizer, C.: Enabling Tailored Therapeutics with Linked Data. In: Proceedings of the WWW 2009 Workshop on Linked Data on the Web, LDOW 2009 (2009)
15. Kimmig, A., Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: A short introduction to probabilistic soft logic. In: NIPS Workshop on Probabilistic Programming: Foundations and Applications (2012)
16. Langegger, A., Wolfram, W.: Rdfstats - an extensible rdf statistics generator and library. In: DEXA Workshops (2009)
17. Maali, F., Cyganiak, R., Peristeras, V.: Re-using cool uris: Entity reconciliation against lod hubs. In: Proceedings of the Linked Data on the Web Workshop 2011 (LDOW 2011), WWW 2011 (2011)
18. Memory, A., Kimmig, A., Bach, S.H., Raschid, L., Getoor, L.: Graph summarization in annotated data using probabilistic soft logic. In: URSW (2012)
19. Naumann, F., Sattler, K.-U.: Information quality: Fundamentals, techniques, and use (2006)
20. Ruckhaus, E., Vidal, M.-E.: The BAY-HIST Prediction Model for RDF Documents. In: Proceedings of the 2nd ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web-CEUR, vol. 611, pp. 30–41 (2010)
21. Stankovic, M., Jovanovic, J., Laublet, P.: Linked data metrics for flexible expert search on the open web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 108–123. Springer, Heidelberg (2011)
22. Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., Zhang, X.-N.: Link prediction for annotation graphs using graph summarization. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 714–729. Springer, Heidelberg (2011)
23. Villazón-Terrazas, B., Vilches-Blázquez, L., Corcho, O., Gómez-Pérez, A.: Methodological guidelines for publishing government linked data linking government data. In: Wood, D. (ed.) Linking Government Data, ch. 2, pp. 27–49. Springer, New York (2011)
24. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 650–665. Springer, Heidelberg (2009)
25. W3C. OWL Web Ontology Language Reference (2004)

Web Service Composition Based on Petri Nets: Review and Contribution*

Yudith Cardinale[1], Joyce El Haddad[2],
Maude Manouvrier[2], and Marta Rukoz[2,3]

[1] Universidad Simón Bolívar, Dep. de Computación y T.I.
Caracas, Venezuela
[2] Université Paris-Dauphine, LAMSADE, CNRS UMR 7243
Paris, France
[3] Université Paris Ouest Nanterre La Défense
Nanterre, France
`yudith@ldc.usb.ve`,
`{elhaddad,manouvrier,marta.rukoz}@lamsade.dauphine.fr`

Abstract. Web Services (WSs) are the most used implementation of service-oriented architectures allowing the construction and the sharing of independent and autonomous software. WS composition consists in combining several WSs into a Composite one, which becomes a value-added service, in order to satisfy complex users queries. Thus, the WS composition process may imply several phases to identify *how* and *which* WSs will conform the Composite WS, including specification, verification, evaluation, WSs selection, and execution. As it is known, Petri Nets are the main formal models used to describe static vision of a system and dynamic behavior of processes. Then, Petri Nets are well suited to model internal operations of WSs and interactions among them as well as to model the processes in all phases of the WS composition process. In this article we present a review of approaches using Petri Nets for WS composition. Moreover, we describe our experiences in this field: a transactional-QoS-driven WS selection approach and a framework for reliable execution of Composite WSs based on Colored Petri Nets.

Keywords: Transactional Composite Web Services, Automatic Composition, Execution of Composite Web Services, Petri-Nets.

1 Introduction

Large computing infrastructures, like Internet, increase the capacity to share information and services across organizations. Web Services (WSs) have gained significant research interest from both research and industry sectors, motivated by the offer of a language-neutral, loosely-coupled, platform independent, and

* This work was supported by the Franco-Venezuelan CNRS-FONACIT project N°22782.

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 83–122, 2013.

standardized fashion for linking applications within organizations [56]. A WS is a software system designed to support interoperable machine-to-machine interaction over a network, using open standards, such as XML, SOAP[1], JSON[2], WSDL[3], and UDDI[4]. XML is used to represent and tag the data, SOAP and JSON are data transfer protocols, WSDL describes the interface of services, and UDDI, called the *Registry*, is used for registering and listing available services.

WSs allow organizations to process and communicate data without deep knowledge of each other system technologies. Many organizations have engaged their core scientific or business competencies with collections of WSs over the Internet, allowing users to dynamically resolve complex problems by combining available WSs. This combination of several WSs is called WS composition. It concerns *how* and *which* WSs will be combined to obtain a Composite Web Service satisfying the users needs. Users requests can express functional and non-functional requirements. Functional requirements specify the functions that a system or its components must be capable of performing. These are software requirements that define the system behavior, i.e., the fundamental process to produce outputs from inputs. In contrast, non-functional requirements are constraints that allow the description of an application in terms of quality attributes, such as performance-related, reliability, and availability issues. These types of requirements are often related to Quality of Services (QoS) criteria (e.g., response time and price) and transactional properties (e.g., reliable execution, compensatable or not). QoS criteria allow efficient/optimal solutions, while transactional properties define the support of failures during the execution; a reliable execution ensure consistent state of the whole system even in presence of fails. Hence, a Composite WS is a service with more complex structure and more powerful functionality. Accordingly, recently there has been a growing interest in WS composition and the related issues, including efforts in:

- Specification languages and graphical tools, which allow users specify *how* and *which* WSs will be part of the composition. Sometimes, users only decide *how* to combine abstract WSs, indicating the combination of functionalities and indicating some desired non-functional properties in the Composite WS. In this case, the system decides *which* WSs match with every abstract functionality and satisfy non-functional requirements.
- Analysis, verification, and evaluation techniques, which enable designers to test and repair design errors before the actual execution of the Composite WS or evaluate the Composite WS in terms of QoS criteria, if they are considered.

[1] Simple Object Access Protocol, 2007, `http://www.w3.org/TR/soap12-part1/` - Extracted on April 2012.

[2] JavaScript Object Notation 1999 - `http://www.json.org/` - Extracted on March 2013.

[3] Web Services Description Language, 2001, `http://www.w3.org/TR/wsdl` - Extracted on April 2012.

[4] Universal Description Discovery and Integration, 2003, `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec` - Extracted on April 2012.

- Automatic specification and selection techniques and tools, which are capable to automatically decide *how* and *which* WSs should be combined. In general, such techniques and tools are based on Semantic Web technology, such as OWL-S[5], a language that allows to express semantic knowledge of WSs based on ontologies. In this context, WSs functionality can be interpreted by machines and users can resolve complex problems that require the interaction among different tasks without the need of programming expertise [21].
- Execution engines, which actually execute Composite WSs. Once the Composite WS is correctly obtained, it has to be executed, i.e., component WSs are invoked according to the execution control flow depicted by the Composite WS. In a high dynamic infrastructure like Internet, fault tolerant execution has also to be provided.

Composite WSs can be represented in structures such as workflows, graphs, or Petri Nets indicating, for example, the control flow, data flow, WSs execution order, and/or WSs behavior. Somehow, these structures can also support the phases of the WS composition process (i.e., analysis, verification, evaluation, selection, and execution). As it is known, Petri Nets are the main formal models used to describe static vision of a system and dynamic behavior of processes. Hence, Petri Nets are well suited to model internal operations of WSs and interactions among them, as well as huge theoretical investigations with a wide range of efficient algorithms are directly applicable in all phases of the WS composition problem [25].

Some works define, study, validate, and check the compatibility, usability, and behavioral equivalence of WSs by transforming their specifications into Petri Nets [2,29,45]. There exist also frameworks through which users can manually express Composite WSs by using Petri Nets and validate their properties (e.g., reachability, safety, and deadlock free) [39]. While classical Petri Nets are well suited for capturing flows in WSs [20], different flavors of Petri Nets (e.g., Colored, Hierarchical, and Stochastic Petri Nets) are needed to consider more complex information. For example, colors in places and tokens allow to consider the different data types or WS properties [9,11,12,45,57]. Time in the firing rules of Petri Nets allows to take into account the execution time of the WSs or to control WSs synchronization [46,47]. Recent works propose automatic WS composition approaches by modeling with Petri Nets the selection and execution phases [6,9,11,12,35].

This paper presents an overview of how Petri Nets have been used in all phases of WS composition and our contribution in this field. The rest of this article is organized as follows. Sections 2 introduces the concepts and terms in the WS composition context. A review of the more important recent works in the field of WS composition based on Petri Nets are presented in Section 3. In [9,11,12] we have extended the Colored Petri Net (CPN) formalism to incorporate description

[5] OWL-S: Semantic Markup for Web Services, 2004, `http://www.w3.org/Submission/OWL-S/` - Extracted on April 2012.

of non-functional WS properties and model the WS composition phases. All phases of the composition process are directed by CPN unrolling algorithms. A summary of these works is presented in Section 4. Finally, the conclusions of the article are presented in Section 5.

2 Web Service Composition: The Background

WS composition consists in combining several existent WSs to produce more complex services that satisfy more complex users requests. This integration of WSs is called a Composite WS. In this context, the composition problem consists in deciding *how* and *which* WSs will be assembled to satisfy users needs. Accordingly, there has been a growing interest in WS composition including efforts in all phases of the WS composition problem (i.e., specification, verification, selection, and execution) and in transverse issues, such as functional and non-functional properties managing. Normally, approaches focused on each issue need intermediate structures to model the problem. In this section, we briefly describe the most important existing approaches treating the different fields of WS composition process and identify wherein Petri Nets have been used as the model to represent the different associated problems.

2.1 Composite WS Specification and Selection Phases

We call specification, the phase defining *how* the different WSs functionalities will be combined to satisfy the query requirements. The selection indicates *which* particular WSs will be executed to carry out each functionality. These phases can be performed manually by the user or (semi)automatically by the system.

Manual Specification and Selection. When specification and selection phases are manually performed, user indicates *how* and *which* services will be used. A landscape of Web Service-oriented Architecture Languages (WSADLs) to specify Composite WSs, such as Business Process Execution Language for WSs (BPEL4WS)[6], WS Choreography Interface (WSCI)[7], and WS Choreography Description Language (WS-CDL)[8] have emerged and are continuously being enriched with new proposals. WSADLs allow users/designers define *how* and *which* WSs will conform the Composite WS. In this sense, with these languages, WS composition need designers manual coding to define the interaction among WSs. Some graphical tools using workflows or graphs have been created to help users in the description of Composite WS. Afterward, the proper specification

[6] Business Process Execution Language for Web Services, 2001, `http://www.ibm.com/developerworks/library/specification/ws-bpel/` - Extracted on April 2012.

[7] Web Service Choreography Interface, 2002, `http://www.w3.org/TR/wsci/` - Extracted on April 2012.

[8] Web Services Choreography Description Language, 2004, `http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/` - Extracted on April 2012.

using a WSADL is automatically generated from the graphical representation of the Composite WS; YAWL[9] is an example of this kind of tools.

Manual Specification and Automatic Selection. To help designers on the selection phase, several works have been proposed allowing the user specify *how* to combine the functionality of abstract WSs and what non-functional requirements are requested. Then, the system automatically decides *which* WSs match with every abstract functionality and satisfy non-functional requirements [17,19,37]. Generally, these works are based on workflow specifications, some of them also provide control patterns (e.g., sequence, choice, parallel, XOR, AND) to help the user in the specification phase [23].

Analysis, Verification, and Evaluation. Practical experience indicates that the definition of real world Composite WSs, made by users/designers, is a complex and error-prune process. All these languages and tools remain at the descriptive level, without providing any kind of mechanism support for verifying the composition specified in the proposed notation. Therefore, it is needed analysis, verification, validation, and evaluation techniques which: (i) enable designers to test and repair specification errors before actual execution of the Composite WS, or (ii) allow designers to detect erroneous properties (such as deadlock and livelock) and formally verify whether the service process design does have certain desired properties (such as consistency with the conversation protocols of partner service- i.e., interoperability), or (iii) allow designers to evaluate the Composite WS in terms of QoS criteria, if they are considered (see survey [38]). Normally, this phase implies a cyclic process between verification and re-design steps to ensure a correct final Composite WS before it could be executed.

Automatic Specification and Selection. The aforementioned approaches do not provide any methodology to how select WSs, how satisfy the local/global functional and non-functional requirements, or how the business rules should be captured in the Composite WS. The rapid proliferation of available WSs in Internet imposes on designers the composition process as a very complex problem. Hence, an emergent need is claiming for approaches capable to automatically decide *how* and *which* WSs should be combined to satisfy users requests. The ability to perform automatic composition is the next step in the evolution of WS composition area [9,41]. With the advent of Web 3.0, machines should contribute to users needs, by searching for, organizing, and presenting information from the Web which means, users needs can be fully automated on the Internet. Besides, with machine intelligence, users can resolve complex problems that require the interaction among different tasks with minimal technical knowledge [21]. A current trend in this area proposes approaches based on the

[9] Yet Another Workflow Language, `http://www.yawlfoundation.org/` - Extracted on April 2012.

Semantic Web which makes part of the Web 3.0. Semantic Web Service technology aims to provide for rich semantic specifications of WSs through several semantic specification languages such as OWL for Services (OWL-S), the Web Services Modeling Ontology (WSMO)[10], and Semantic Annotations for WSDL and XML Schema (SAWSDL)[11]. Automatic WS composition requires the system to automatically select and assemble suitable WSs to satisfy users requests, while the user only need to specify its requirements on a high level specification. Requirements may consist of functional requirements (i.e., the set of input attributes provided in the query and the attributes that will be returned as output), QoS criteria (e.g., response time and price), and/or transactional properties (e.g., compensatable or not). Because the Composite WS is automatically generated by the system, the validation/verification phase is not necessary in these approaches.

2.2 Composite WS Execution

Once the Composite WS is correctly obtained, it has to be executed, i.e., all component WSs have to be invoked according to the control and execution flow imposed by the Composite WS. Composition execution engines, such as the IBM framework BPWS4J[12] or the open source Orchestra[13] solutions actually execute Composite WSs specified with BPEL4WS. The execution control can be centralized, in which a coordinator manages the whole execution process [43,58], or distributed, in which the execution process proceeds with collaboration of several participants without a central coordinator [4,8]. On the other hand, the execution control could be attached to the WS [5,27] or independent of its implementation [12]. Some execution engines are capable to manage fails during the execution. Ones are based on exception handling [5], others are based on transactional properties [12], and some others use a combination of both approaches [27]. Exception handling normally is explicitly specified at design time, regarding how exceptions are handled and specifying the behavior of the Composite WS when an exception is thrown. In contrast, transactional properties implicitly describe the behavior in case of failures. When transactional properties are not considered, the system consistence is a responsibility of users/designers.

2.3 Functional and Non-functional Requirements

Functional requirements are always provided by users in their queries. They can be expressed according to the specification approach, in terms of explicit

[10] Web Service Modeling Ontology, 2004, `http://www.wsmo.org/2004/d2/v1.0/` - Extracted on April 2012.

[11] Semantic Annotations for WSDL and XML Schema, 2007, `http://www.w3.org/TR/sawsdl/` - Extracted on April 2012.

[12] Business Process for Web Services JavaTM, `http://www.alphaworks.ibm.com/tech/bpws4j` - Extracted on April 2012.

[13] Orchestra, `http://orchestra.ow2.org/xwiki/bin/view/Main/WebHome` - Extracted on April 2012.

specifications for WSADL (such as BPEL4WS or WSCI), abstract functionalities for workflows or graphs, or input/output attributes for automatic selection. Some approaches allow non-functional requirements along the query, such as QoS criteria and transactional user preferences [6,9,23,24]. Heterogeneity, which means different QoS values in terms of response time, cost, reliability, throughput, trust, etc., granted by multiple distributed WSs delivering the same functionality, must be adequately managed to ensure efficient Composite WS. In this sense, knowledge about the parameters whose values describe the execution quality of available WSs, play an important role during the composition process in order to satisfy particular QoS user requirements. This means that sufficiently rich, machine-readable descriptions will be required to aid with a solution that composes a diversity of heterogeneous WSs. Generally, each WS is described in terms of functional properties and also its QoS criteria by using semantic technology (such as WSDL and OWL-S documents). This information is registered in a *Registry*, in order to discover and select WSs to be part of a Composite WS. When QoS criteria guide the composition process, the idea is to obtain an optimal Composite WS. WSs with the highest quality will produce more efficient compositions.

When transactional properties are taken into account, the Composite WS also allows reliable and fault tolerant execution. In this sense, WS transactional properties guarantee integrity, continuity, and data consistency of business processes, even in presence of failures [3]. The execution of a Transactional Composite WS has to adapt to the open, dynamically changing environment, and unpredictable conditions of distributed applications, where remote services may be affected by failures and availability of resources. The most used definition of individual WS transactional properties (TPs) is as follows (see survey [10]).

Let s be a WS: (i) s is **pivot** (p), if once s successfully completes, its effects remains forever and cannot be semantically undone (compensated), if it fails, it has no effect at all; (ii) s is **compensatable** (c), if it exists another WS s', which can semantically undo the execution of s, even after s successfully completes; (iii) s is **retriable** (r), if s guarantees a successfully termination after a finite number of invocations; (iv) the **retriable** property can be combined with properties p and c defining **pivot retriable** (pr) and **compensatable retriable** (cr) WSs.

In [17], the following TPs of Transactional Composite WSs have been derived from the TP of its component WSs and their execution order (sequential or parallel).

Let tcs be a Transactional Composite WS: (i) tcs is **atomic** ($\boldsymbol{a}$), if once all its component WSs complete successfully, they cannot be semantically undone, if one component WS does not complete successfully, all previously successful component WSs have to be compensated; (ii) tcs is **compensatable** (c), if all its component WSs are compensatable; (iii) tcs is **retriable** (r), if all its component WSs are retriable; (iv) the retriable property can be combined with properties $\boldsymbol{a}$ and c defining **atomic retriable** ($\boldsymbol{a}r$) and **compensatable retriable** (cr) Transactional Composite WSs.

According to these TPs, we can establish three possible recovery techniques in case of failures [9,11,12,19,27,37]:

- *Backward* recovery: it consists in restoring the state (or a semantically closed state) that the system had at the beginning of the *tcs* execution; i.e., all the successfully executed WSs, before the fail, must be compensated to undo their produced effects. All TPs (p, $\boldsymbol{a}$, c, pr, $\boldsymbol{a}r$, and cr) allow backward recovery, because by definition all these properties ensure that if it is a failure, there is no effect at all. Rollback and compensation are some of the most used techniques to ensure this characteristic;
- *Forward* recovery: it consists in repairing the failure to allow the failed WS to continue its execution. TPs pr, $\boldsymbol{a}r$, and cr allow forward recovery, because by definition these properties allow to retry until success;
- *Semantic* recovery: after the successful end of a *tcs* execution, a semantic recovery consists in reaching a state, which is semantically closed to the state the system had before the *tcs* execution. Only TPs c and cr allow semantic recovery, because by definition the compensatable property allows to semantically undo the effects produced by the execution.

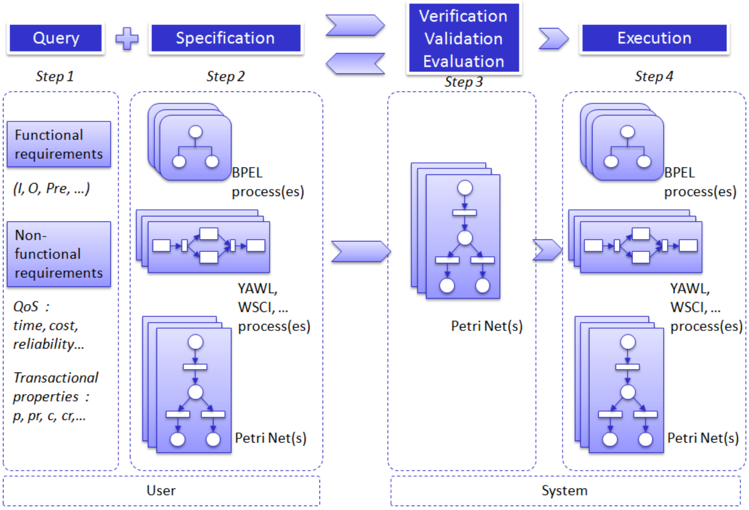


Fig. 1. Manual specification and manual/automatic selection phases of WS composition

2.4 Petri Nets in the WSs Composition Process

The use of Petri Nets in the phases of WS composition process has played an important role. Indeed, different flavors of Petri Nets have been used directly as a specification tool or by transforming specifications in a WSADL or workflows,

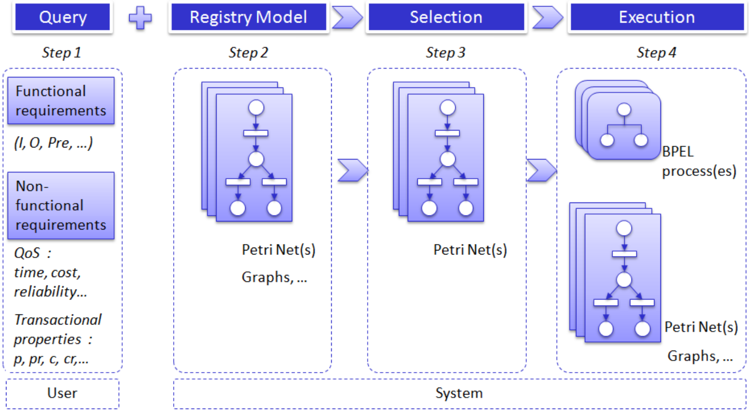


Fig. 2. Automatic selection phase of WS composition

to model the behavior of the Composite WS. Petri Nets also allow automatic validation. Once the validation is accomplished, the resulting Composite WS, represented by a Petri Net, is transformed again in a control execution language (such as BPEL4WS or WSCI), or used to direct the execution. Figure 1 illustrates the use of Petri Nets on those phases. Currently, new emergent approaches, such as [6,9,11,12], responding Web 3.0 challenges, use Petri Nets not only to model the Composite WS (which is automatically generated) but also the set of available WSs (WS registry). Semantic Web, which make part of Web 3.0, allows to use semantic knowledge to provide automatic WS composition. Therefore, automatic WS composition does not need a verification phase and Petri Nets can be used to guide selection and execution phases. Figure 2 illustrates this case.

3 Composite Web Services Based on Petri-Nets: A Review

This section presents state-of the-art on how the different Petri Nets have been used in each phase of the WS composition problem.

3.1 Specification Phase (see Step 2 in Fig. 1 and 2)

A Petri Net-Based Model of WS Composition. As far as we know, one of the first approaches using Petri Nets to model WS composition is [20]. This work has proposed a Petri Net-based algebra to represent a WS by a Petri Net, called *Service Net*, which is formally defined as a tuple $SN = (P, T, W, i, o, \ell)$, where:

- P is a finite set of places representing the state of the service,
- T is a finite set of transitions representing the operations of the service,

- $W \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs (flow relation),
- i is the input place with ${}^{\bullet}i = \{x \in P \cup T \mid (x, i) \in W\} = \emptyset$,
- o is the output place with $o^{\bullet} = \{x \in P \cup T \mid (o, x) \in W\} = \emptyset$, and
- $\ell : T \longrightarrow \mathcal{A} \cup \{\tau\}$ is a labeling function where $\mathcal{A}$ is a set of operations names. $\tau \notin \mathcal{A}$ is a silent (or empty) operation.

A WS execution starts when a token is in place i and terminates when a token reaches the place o. As explained in [20], Service Net contains only one input place and one output place in order to facilitate the definition of composition operators. For each WS composition operator (e.g., sequence, alternative, iteration), a formal Petri Net-based definition is given, providing a direct mapping from each operator to a Petri Net construction.

For example, let consider two WSs s_1 and s_2, respectively represented by Petri Nets as Figure 3(a) and 3(b) show: $SN_1 = (P_1, T_1, W_1, i_1, o_1, \ell_1)$ and $SN_2 = (P_2, T_2, W_2, i_2, o_2, \ell_2)$. The Composite WS that performs s_1 before s_2, using sequence operator, is represented by $SN = (P, T, W, i, o, \ell)$ (see Figure 3(c)), where:

- $P = P_1 \cup P_2$;
- $T = T_1 \cup T_2 \cup \{t\}$, where t represents the transition added between places o_1 and i_2;
- $W = W_1 \cup W_2 \cup \{(o_1, t), (t, i_2)\}$;
- $i = i_1$; $o = o_2$; $\ell = \ell_1 \cup \ell_2 \cup \{(t, \tau)\}$.

In the same way, the Composite WS performing either s_1 or s_2, but not both, using an alternative operator is represented by $SN = (P, T, W, i, o, \ell)$, as shown in Figure 3(d), where:

- $P = P_1 \cup P_2 \cup \{i, o\}$, where i represents the input place of SN and o represents the output place of SN, such that i is the input place of two transitions t_{i_1} and t_{i_2}, which output places are respectively i_1 and i_2, and o is the output place of two transitions t_{o_1} and t_{o_2}, which input places are respectively o_1 and o_2.
- $T = T_1 \cup T_2 \cup \{t_{i_1}, t_{i_2}, t_{o_1}, t_{o_2}\}$, where t_{i_1} and t_{i_2} respectively represent the transitions between place i and places i_1 and i_2; t_{o_1} and t_{o_2} respectively represent the transitions between output places o_1 and o_2 and place o;
- $W = W_1 \cup W_2 \cup \{(i, t_{i_1}), (i, t_{i_2}),$ $(t_{i_1}, i_1), (t_{i_2}, i_2), (o_1, t_{o_1}), (o_2, t_{o_2}), (t_{o_1}, o), (t_{o_2}, o)\}$;
- $\ell = \ell_1 \cup \ell_2 \cup \{(t_{i_1}, \tau), (t_{i_2}, \tau), (t_{o_1}, \tau), (t_{o_2}, \tau)\}$.

While such a Petri Net-based model allows the detection of inconsistencies both within and among WSs by analyzing or verifying certain well-known Petri Net properties (e.g., reachability, deadlock, and liveness), this approach does not include the management of time and resources of WS. Therefore, it has been extended, for example by [57], to Colored Petri Net, where color of token represents the type of information managed by WSs. To model and analyze time

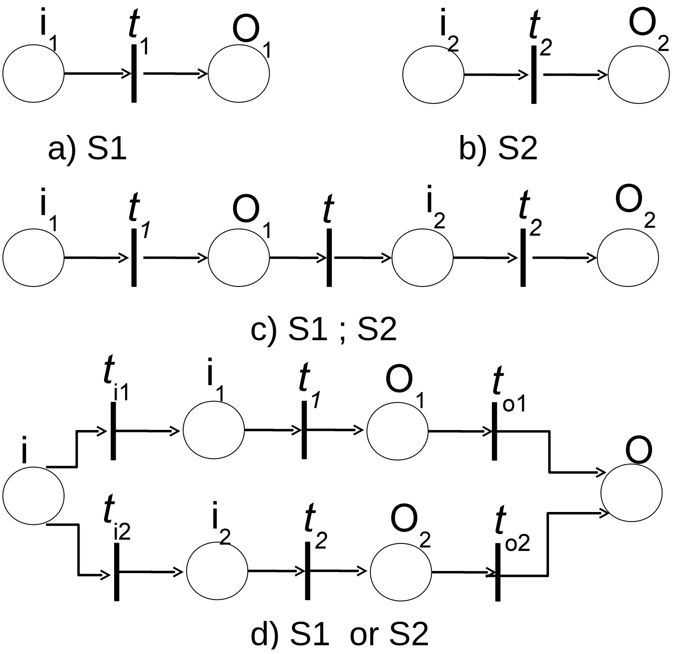


Fig. 3. Example of Service Net

constrained WS composition, a Timed Petri Net, called Service Composition Timed-constrained Petri Net (SCT-Net), is proposed in [55] which takes into account besides time constraints, WS priority and transactions properties.

A SCT-Net is a 4-tuple, (PN, IO, C, Pr), where PN is a Petri Net, IO is a special type of place in PN representing the interface of the SCT-Net, C is a function assigning firing time of tokens, and Pr a priority function of transitions. This work also considers WS transactional properties (p, r, etc.) modeled by a SCT-Net representing patterns for the composition model.

A requirement model of service composition is a 3-tuple (WS, RL, RT), where WS is a set of services, RL is the relation function among services, and RT is the QoS of services such that RT(WS_i) is (ST_i, d_i, SP_i, NP_i, Sr_i) representing the running time, deadline, transactional property, the preventive property, and the priority respectively for each WS i. The composition is performed by connecting the SCT-Net representing each WS, adding priorities to each transition, and adding transitions to connect input and output places.

Manual Specification and Automatic WS Selection. In [18], a generalized Associative Petri Net (APN) [44], derived from fuzzy Petri Nets, has been used to model automatic WS selection. In this approach, the user request is expressed as a tuple composed by abstract tasks (or functional requirements manually specified by the user), constraints over attributes of WSs, and QoS requirements. This approach proposes an APN to describe multi-attribute multi-constraint relations, and associate relationships among component WSs. The APN contains three types of nodes: places representing all candidate WSs for each abstract task, support nodes representing relationship constraints among component WSs in adjacent tasks, and transitions representing the trust values among component WSs in adjacent tasks. Based on the APN, a genetic algorithm has been proposed to find the optimal Composite WS according to user requirements. A similar approach is proposed in [28] using an extended Colored Petri Net, where places represent WSs, tokens represent relation constraints, and transitions represent trust values among component WSs in adjacent tasks. For each abstract task, the system performs the selection process by dynamically identifying from candidates, instances of WSs that match its functionality. Colors are used to represent global constraints. Finally, a genetic algorithm is used to obtain the firing sequences representing possible Composite WSs. The optimal ones are the firing sequences having the highest trust value in the Colored Petri Net model.

3.2 Verification, Validation, and Evaluation Phase (see Step 3 in Fig. 1)

Verification and Validation. Many approaches transform business processes specified in some Web Service-oriented Architecture language (WSADL), such as BPEL4WS, WSCI, and WS-CDL, into Petri Nets in order to analyze their functionality and/or verify the WSs interoperability or compatibility [30]. In approaches working with BPEL4WS, BPEL structures are divided into internal behavior (representing internal operations of WSs, e.g., assign, terminate, wait, sequence, flow, while, switch) and external behavior (representing the interface of the WSs, e.g., receive, reply, invoke). To achieve this transformation, specific types of Petri Nets can be used, like workflow net [48] in [33,40,45] or open WorkFlow Net(s) (oWFN) [34] in [16,29]. A workflow net has two special places representing the begin and the end of the process and allows dividing the relation flow into internal and communication ones. Open workflow net is a generalized version of workflow net where the interface of the process is represented by a set of input and output places. A GNU Software, BPEL2oWFN [14] has been proposed to transform BPEL process to oWFN, based on [22,29].

In order to analyze the Composite WS, all BPEL processes (representing component WSs) are transformed into a Petri Net and are composed into one Petri Net (representing the whole composition) [13,14,16,32,33,45,52]. The analysis, generally considers verification of compatibility and similarity of processes or verification of reliability including reachability, safety, and deadlock. Different kind of methods based on reachability graph as in [32,33] or based on structural properties like Petri Net siphons as in [52] are used in order to identify WSs incompatibilities. Some approaches try to resolve the incompatibilities

[14] GNU BPEL2oWFN, 2007, `http://www.gnu.org/software/bpel2owfn/` - Extracted on April 2012.

by adding mediators/adapters to glue partially compatible services [16,45,52]. Each mediator is represented by a specific Petri Net transition correcting the incompatibilities. The approach presented in [16] is based on open Workflow net, while approach presented in [45] has used a combination of Colored Petri Net and Workflow net to verify behavioral compatibility of processes. Recently, the latter approach was extended in [26] to check behavioral similarity between processes.

Considering that WS-CDL [49] specifies the interactions of component WSs of a composition from a more global point of view than BPEL4WS, prioritized-timed extensions of Colored Petri Nets have been proposed in [47]. The idea is to transform WS-CDL specifications into Colored Petri Net in order to consider timed or prioritized interactions among WSs to verify reachability. In [46], a Timed Petri Net representation of WSs flow is derived from WSDL to verify correctness and deadlock freeness. In [7], a formal encoding of OWL-S into an Open Consume-Produce Read (OCPR) net has been proposed. OCPR is a variant of classical Petri Net having two kinds of places, control and data places, modeling the control and the data flow of a WS. The goal of this model is to verify the WS specification and check if the replacement of a part of the WS by another WS can be done without changing the behavior described in the initial specification.

Hierarchical Colored Petri Nets (HCPN) for verification and analysis of functional correctness, behavioral and performance of Composite WSs are used in [53,54]. These works translate BPEL [54] and WSCI [53] processes into two HCPNs, called Interface-Nets (I-Nets) and Composition-Nets (C-Nets), to verify reachability, boundness, dead transitions, dead markings, liveness, fairness, etc. I-Nets describe WS choreography in a specific scenario, while C-Nets connect at least two I-Nets among different organizations. In I-NETs, places model the states of the system by markings that represent distribution of data values (tokens) in places. Each place has a color set, which represents the types of values it can hold; places can also represent input/output messages or control messages. On the other hand, transitions model auxiliary transitions (sequence, loop, while, fork, etc.), substitution transitions (subpages modeling subprocesses), and activity transition. In C-Nets, places are input/output messages, transitions represent organization transitions (abstract representation of an I-Net). Beside the verification for correctness of BPEL or WSCI workflows, they also verify that the transformation process should be complete, unique, syntactically correct, semantically correct, and could terminate.

The model in [20] has been extended by [35,36,42,50] into paired Petri Net to model the compensation flow. The compensation flow implies the execution in the reverse execution order in case of failures. Approaches presented in [35,36] propose aggregation of classical QoS criteria (response time, execution time, reliability, availability, cost, and reputation) taking into account the compensation flow. Verification of reachability, deadlock-free, and liveness of compensation paired Petri Net is done by [42,50].

Evaluation. Some works transform WSADL specifications into Petri Nets to evaluate aggregated QoS criteria of the Composite WS. In [31], two complexity metrics are proposed. A count-based metric to measure static features of the Composite WS by counting, for example, the number of places, number of transitions, average degree of places, average degree of transitions, and number of transfers per service; and an execution path-based metric to measure dynamic execution complexity, to which a weight is assigned to each BPEL structure according to its execution complexity. Then the aggregated weight is calculated according to the paths the Composite WS could execute. In [15,51], WSCI and BPEL processes are transformed into Stochastic Petri Nets to make analytical evaluation of QoS metrics. Each timed transition has an execution probability, an execution rate, and the overhead per unit of time. Based on these parameters, QoS metrics such as expected-process-normal-execution-time, process-normal-completion-probability, and expected-overhead-of-normal-completion can be obtained.

3.3 Automatic Selection Phase (See Step 3 in Fig. 2)

Automatic WS selection approaches based on Petri Net are proposed in [6,24]. In both works, the *Registry* is modeled as a Petri Net, in which transitions represent WSs and places represent WS input/output attributes. Semantic of places are deviated from an external ontology. In [6], the selection process is implemented by an unfolding algorithm over the Petri Net, guided by a quality function that considers functional, QoS, and transactional requirements provided in the user query to generate a Transactional Composite WS. From the initial marking, denoted by inputs in the query, a final marking, denoted by desired outputs, is reached. The quality function allows to obtain an optimal solution. The approach proposed in [24] is based on Petri Net coverability. The coverability tree and coverability graph (both representing all possible reachable markings) are built from the user query, which contains the values of inputs, list of desired outputs, and QoS preferences. Places with the same semantic meaning are merged and its tokens are summed. For each query, the coverability tree and coverability graph are generated to find all coverability paths (those that reach the target marking denoted by the desired outputs), from them they choose the one with the best aggregated QoS. This work considers the following QoS criteria: response time, reliability, usability, and cost. The WSs quality is calculated according to user preferences, given in the query.

In [41], a Colored Petri Net (called Moap for Message Oriented Activity based Petri Net model) has been proposed to model statefull services[15] and to allow automatic WS selection. Places in the Petri Net correspond to the state of the services and transitions denote the action of the services. Tokens correspond to meta-message and the color of tokens correspond to the different types of

[15] where the behavior of services are described by state transition model – see `http://xml.coverpages.org/statefulWebServices.html` for a definition.

meta-messages. Given a set of input parameters, a set of desired output parameters (both representing meta-messages) and a set of behaviors, the proposed algorithm find a composite process.

In [9], another Colored Petri Net has been proposed to transactional-QoS driven Web Service composition. This approach is presented in detail in Section 4.

3.4 Execution Phase (See Step 4 in Fig. 1 and 2)

Research in execution engines or frameworks based on Petri Nets is scarce. However, it is easy to realize that all works proposing selection approaches that automatically generate Petri Nets, as works described in previous section, can execute the resulting Composite WS in two fashions. By transforming the Petri Net in a business process represented in some WSADL or by implementing algorithms that execute the resulting Petri Net. Based on compensation paired Petri Nets, in [35] an algorithm to execute the Petri Net is proposed. During the execution, the state of transitions execution is kept in a log. When a fail occurs, the compensation process is based on the log and on the compensation flow. A framework for reliable execution of Composite WSs based on Colored Petri Net is proposed in [11,12]. This approach is presented in detail in Section 4.

Table 1 gives a summary of all aforementioned approaches of this survey section. Table 2 recalls the type of Petri Net used by each approach and the tackled composition phase.

4 Transactional Composite WSs Based on Colored Petri Net: Our Contribution

We have extended the Colored Petri Net (CPN) formalism to incorporate description of non-functional WS properties and model the automatic WS composition [9,11,12].

In [9], the WSs and their execution flow are automatically selected and represented by a CPN [9]. The election process, implemented by a COMPOSER, follows an unrolling algorithm guided by QoS and transactional parameters to prune the search space and produce an optimal and reliable solution. It builds automatically two CPNs: (i) a CPN representing a Transactional Composite Web Service, which satisfies the user query (expressed as functional conditions, QoS criteria, and transactional property requirements) and specifies the execution flow, and (ii) a CPN to represent its corresponding compensation flow, which should be followed in case of failures during the execution. In these CPNs, WSs inputs and outputs are represented by places and WSs are represented by transitions. In the Transactional Composite WS colors represent the transactional properties of WSs, while in the CPN representing the compensation flow, colors indicate the WS execution state, needed to decide what to do in case of failures.

In [11,12], the execution of the Transactional Composite WS, represented by a CPN, is the responsibility of an EXECUTER. The EXECUTER actually invokes the WSs in the composition and provides recovery techniques in case of

Table 1. Summary of Petri Net-based WS Composition approaches

Approach	Objective
[20]	Propose a Petri Net-based algebra to capture the semantics of WS composition and to formally model a Composite WS, which is the first step to allow the verification of the composition and the detection of inconsistencies within and among WSs.
[57]	Propose a Colored Petri Net to model types of resources managed by WSs.
[55]	Propose a Time-constrained Petri Net to model and analyze time constrained WS composition.
[18]	Define automatic WS selection based on manual user specifications and using fuzzy Petri Net.
[22,29,40]	Transform a BPEL process to a Petri Net in order to allow process verification.
[33,52]	Transform two or more BPEL processes to Petri Nets and compose Petri Nets in order to detect WSs incompatibility.
[16,26,45]	Transform two or more BPEL processes to Petri Nets, compose Petri Nets in order to detect WSs incompatibility, and add mediator transitions to correct partial incompatibilities among WSs.
[13,14,53,54]	Transform BPEL or WSCI processes into Petri Nets in order to verify reachability, safety, and deadlock.
[15,31,51]	Transform WSADL specifications into Petri Nets to evaluate aggregated QoS criteria of the Composite WS.
[7]	Generate Petri Net from OWL-S definition of WS for checking the correctness of WS specifications and the replace-ability of (sub)services.
[46]	Define Timed Petri Net representation of WSs flow from WSDL specification.
[47]	Generate Petri Net from WS-CDL definition of Composite WS for simulating timed or prioritized interactions among component WSs.
[6]	Propose an automatic QoS-transactional WS selection based on classical Petri Nets.
[24]	Propose an automatic QoS WS selection based on Petri Net coverability.
[41]	Propose an automatic WS selection based on Colored Petri Nets.
[9]	Propose an automatic QoS-transactional WS selection based on Colored Petri Nets.
[11,12]	Propose framework for reliable execution of transactional Composite WS based on Colored Petri Nets.
[35,36,42,50]	Propose compensation paired Petri Net to model, evaluate QoS, verify reachability and deadlock, and control the execution .

Table 2. Comparison of the Petri Net-based WS Composition

Approach	Type of PN	Composition phase
[13,20]	Classical	Modelisation
[57]	Colored Petri Net	Modelisation
[55]	Timed-constrained Petri Net	Modelisation
[18]	Generalized Associative Petri Net model (APN) [44] (fuzzy Petri Net)	Manual Specification and automatic selection
[26,45]	Combination of Colored Petri Net and Workflow net	Verification, validation, and evaluation
[33,40]	Workflow net [48]	Verification, validation, and evaluation
[16,29]	open Workflow net [34]	Verification, validation, and evaluation
[7,14,22,31]	Adaptation of classical Petri Net	Verification, validation, and evaluation
[15,46,51]	Time Petri Net	Verification, validation, and evaluation
[47]	Prioritized-timed extension of colored Petri Net	Verification, validation, and evaluation
[36,42,50]	Compensation Paired Petri Net	Verification, validation, and evaluation
[53,54]	Hierarchical Colored Petri Net	Verification, validation, and evaluation
[6,24]	Classical Petri Net	Automatic selection
[9,41]	Colored Petri Net	Automatic selection
[11,12]	Colored Petri Net	Execution
[35]	Compensation Paired Petri Net	Execution

failures, by algorithms that execute the CPNs representing the execution flow and its corresponding compensation flow. It provides a correct and fault tolerant execution of Transactional Composite WSs by: (i) ensuring that sequential and parallel WSs will be executed according to the execution flow depicted by the corresponding CPN; (ii) in presence of failures, trying a forward recovery by replacing the faulty WS by another WS which satisfies the substitution properties; and (iii) if replacement is not possible, leaving the system in a consistent state by executing a backward recovery by a compensation process.

Our framework leverages on the formalism of CPN to efficiently implement a QoS-and-Transactional driven composition process and on the advantages of portability, modularity, and flexibility of Web 3.0 techniques (semantic Web and ontologies) in order to provide totally transparent use to users and developers; i.e., user only provides the functional and non-functional requirements and no instrumentation/modification is needed for WSs participating in the composition. In this section, we present our extended CPN formalism. Table 3 summarizes all notations used in our definitions.

Formally, we define a query in terms of input and output attributes, QoS constraints, and the required global transactional property as follows.

Definition 1 Query.
Let $Onto_A$ be the integrated ontology[16]*. A Query Q is a 4-tuple (I_Q, O_Q, W_Q, T_Q), where $I_Q = \{i \mid i \in Onto_A$ is an input attribute$\}$, $O_Q = \{o \mid o \in Onto_A$ is an*

[16] Many ontologies could be used and integrated.

Table 3. Notation of the article

Variable	Description
Q	Query
I_Q	Input attributes
O_Q	Output attributes
W_Q	Set of QoS weights
T_Q	Required transactional property
T_0	Semantic recovery guarantee
T_1	No guarantee of the result compensation
$WSDN$	Web Service Dependency Net (WS registry)
A, A_Q, A'	Finite non-empty set of places
S, S_Q, S'	Finite set of transitions
F, F_Q, F^{-1}	Flow relation
ξ, ξ_Q, ζ	Color function
I	Color of the tokens in places I_Q
p, pr	Color of WSDN transition corresponding to resp. pivot or pivot retriable WS
a, ar, c, cr	Color of transition or tokens corresponding to resp. atomic, atomic retriable, compensatable or compensatable retriable (composite) WS
C_A	Color function such that $\Sigma_A = \{I, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$
C_S	Color function such that $\Sigma_S = \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$
M	Function assigning tokens to places
M_Q	Initial Marking
$(^\bullet x)$	Set of x's predecessors
$(x^\bullet)$	Set of x's successors
$WSDN_Q$	Transactional Composite WS corresponding to query Q
BR_WSDN_Q	Backward Recovery Net
BR_WSDN_Q	Backward Recovery Net
σ	Firing sequence
SC	Service class
In, Ru, Ex, Co, Fa, Ab	Colors of the transition of BR_WSDN_Q representing respectively initial, running, executed, compensated, failed, or abandoned state of a WS
$\equiv_F$	Functional Substitute
$\equiv_{EF}$	Exact Functional Substitute
$\equiv_T$	Transactional Substitute

output attribute whose value has to be produced by the system$\}$, $W_Q = \{(w_i, q_i) \mid w_i \in [0,1]$ *with* $\sum_i w_i = 1$ *and* q_i *is a QoS criterion*$\}$, *and* T_Q *is the required transactional property:* $T_Q \in \{T_0, T_1\}$. *If* $T_Q = T_0$, *the system guarantees that a semantic recovery can be done by the user. If* $T_Q = T_1$, *the system does not guarantee the result can be compensated. In both cases, if the execution is not successful, nothing is changed on the system.*

4.1 Automatic Selection Phase: The COMPOSER

We model the *Registry*, in which all WSs are registered and described by WSDL and OWL-S, by a *Web Service Dependency Net* (WSDN) defined as:

Definition 2 WS Dependency Net (WSDN).
A WSDN is a 4-tuple (A, S, F, ξ)*, where:*

- A *is a finite non-empty set of places, corresponding to input and output attributes of the WSs in the Registry such that* $A \subset Onto_A$*;*
- S *is a finite set of transitions corresponding to the set of WSs in the Registry;*
- $F : (A \times S) \cup (S \times A) \rightarrow \{0, 1\}$ *is a flow relation indicating the presence (*1*) or the absence (*0*) of arcs between places and transitions defined as follows:* $\forall s \in S$*, (*$\exists a \in A \mid F(a, s) = 1$*)* $\Leftrightarrow$ *(a is an input attribute of s) and* $\forall s \in S$*, (*$\exists a \in A \mid F(s, a) = 1$*)* $\Leftrightarrow$ *(a is an output attribute of s);*
- ξ *is a color function such that* $\xi : C_A \cup C_S$ *with:* $C_A : A \rightarrow \Sigma_A$*, a color function such that* $\Sigma_A = \{I, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ *representing, for* $a \in A$*, either the transactional property of the Composite WS that can produce it or the user input (*I*), and* $C_S : S \rightarrow \Sigma_S$*, a color function such that* $\Sigma_S = \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ *representing the transactional property of* $s \in S$*.*

The firing of a transition of a WSDN corresponds to the selection of a WS, which will participate in the Composite WS allowing to answer the user query Q. We define the *marking* of a WSDN, the $fireable$ property of a transition, and the $firing\ rules$ in such way we obtain, at the end, a Transactional Composite WS. Thus, given a user query Q and a WSDN, the selection process will create a CPN, called $WSDN_Q$, sub-part of WSDN, which satisfies Q and its corresponding compensation CPN, called BR_WSDN_Q, representing the backward recovery process.

Definition 3 Marked WSDN.
A marked WS Dependency Net is a pair $(WSDN, M)$*, where* M *is a function which assigns tokens (values) to places such that* $\forall a \in A$*,* $M(a) \subseteq \{\emptyset, Bag(\Sigma_A)\}$*, where Bag corresponds to a set which can contain several occurrences of the same element.*

Definition 4 Initial Marking (M_Q).
The initial marking M_Q *depends on the user query* Q *and is defined as:* $\forall a \in (A \cap I_Q)$*,* $M_Q(a) = \{I\}$ *and* $\forall a \in (A - I_Q)$*,* $M_Q(a) = \emptyset$*.*

According to CPN notation, we have that for each $x \in (A \cup S)$, $({}^\bullet x) = \{y \in A \cup S : F(y, x) = 1\}$ is the set of its predecessors, and $(x^\bullet) = \{y \in A \cup S : F(x, y) = 1\}$ is the set of its successors.

Transactional properties impose restrictions in the WS execution order (sequential or parallel) [17]. To control this restrictions, from a state of $WSDN$ defined by a marking M, it is possible identifying which transitions can be executed in sequential or parallel. In this context, if WS s is selected, then colors of the tokens in places $a \in (A - {}^\bullet s)$ correspond to the transactional properties of WSs that could be executed in parallel with WS s, whereas colors of tokens in places $x \in ({}^\bullet s)$ correspond to the transactional properties of WSs that could be executed in sequential order with WS s. Table 4 presents these restrictions.

For example, rule 1 means that only **compensatable** or **compensatable retriable** WS can be sequentially executed before a **pivot** one. Rule 2 means that a **pivot** WS can only be executed with a **compensatable retriable** one. Hence, to determine if a transition $s \in S$ is fireable, we have to analyze (i) its own color ($C_S(s)$), (ii) the color of the other attribute places ($a \in (A - {}^{\bullet}s)$) to consider the parallel fireable transitions, and (iii) the color of its input places ($x \in {}^{\bullet}s$) to consider the sequentially fired transitions. More precisely, the fireable conditions are deduced from transactional properties of individual WSs and their execution order, using Table 4, as follows.

Table 4. Transactional rules of [17]

Transactional property of a WS	**Sequential compatibility** (${}^{\bullet}s$)	**Parallel compatibility** ($A - {}^{\bullet}s$)
p, $\boldsymbol{a}$	$c \cup cr$ (rule 1)	cr (rule 2)
pr, $\boldsymbol{ar}$	Σ_S (rule 3)	$pr \cup \boldsymbol{ar} \cup cr$ (rule 4)
c	$c \cup cr$ (rule 5)	$c \cup cr$ (rule 6)
cr	Σ_S (rule 7)	Σ_S (rule 8)

Definition 5 Fireable Transition.
A marking M enables a transition s iff all its input places contain a token ($\forall x \in ({}^{\bullet}s)$, $M(x) \neq \emptyset$) and at least one of the following conditions is verified:

1. No transition has been fired yet: ($\forall a \in A$, $M(a) \in \{I, \emptyset\}$)
2. The color of transition s is cr: ($C_S(s) = cr$)
3. The color of transition s is pr or ar and all its parallel fireable transitions are not fired or have attributes with tokens colored by ar or cr (see rules 3 and 4):
$(C_S(s) \in \{pr, \boldsymbol{ar}\}) \wedge [\forall a \in (A - {}^{\bullet}s),\ M(a) \in \{\emptyset, Bag(\{I, \boldsymbol{ar}, cr\})\}]$
4. The color of transition s is c and all its parallel and sequential fireable transitions are not fired or have attributes with tokens colored by c or cr (see rules 5 and 6):
$(C_S(s) = c) \wedge [\forall a \in (A - {}^{\bullet}s),\ M(a) \in \{\emptyset, Bag(\{I, c, cr\})\}] \wedge [\forall x \in ({}^{\bullet}s),\ M(x) \in Bag(\{I, c, cr\})]$
5. The color of transition s is p or a and all its parallel fireable transitions are not fired or have attributes with tokens colored by cr (see rule 2) and all its sequential ones are not fired or have attributes with tokens colored by c or cr (see rule 1):
$(C_S(s) \in \{p, \boldsymbol{a}\}) \wedge [\forall a \in (A - {}^{\bullet}s),\ M(a) \in \{\emptyset, Bag(\{I, cr\})\}] \wedge [\forall x \in ({}^{\bullet}s),\ M(x) \in Bag(\{I, c, cr\})]$

We illustrate these definitions in Figures 4 and 5. Figure 4 shows a $WSDN$ with nine WSs. Figure 5 shows all possible fireable transitions when query is $Q = (I_Q = \{A_1\}, O_Q = \{A_7, A_8\}, W_Q, T_Q = T_0)$.

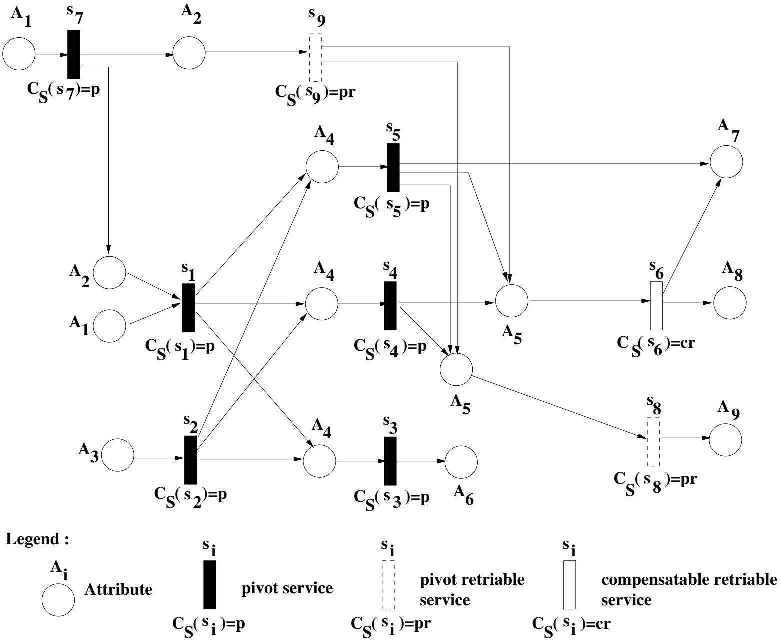


Fig. 4. Example of WSDN

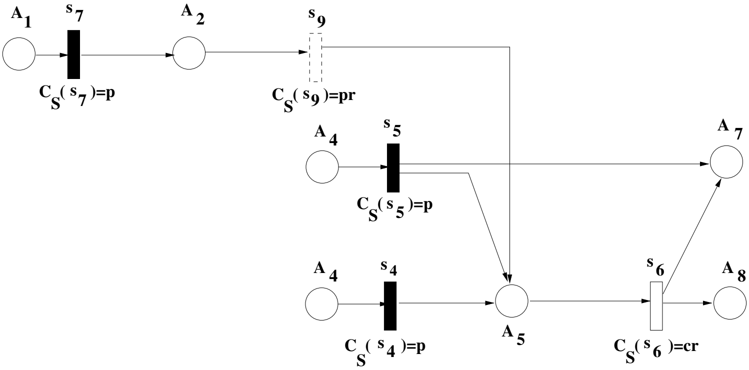


Fig. 5. Fireable transitions for WSDN of Figure 4 when $Q = (I_Q = \{A_1\}, O_Q = \{A_7, A_8\}, W_Q, T_Q = T_0)$

To know the aggregated transactional property of the resulting Transactional Composite WS (composed by all the WSs corresponding to the fired transitions), we defined the color associated with a marked WSDN by:

Definition 6 Color of a Marked WSDN.
Let (WSDN,M) be a marked WSDN. Its color is: $\mathcal{C}_M \in \{I, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$. $\mathcal{C}_M = I$*, when no transition has been fired (i.e., when no WS has been selected and there is no resulting Transactional Composite WS). Otherwise,* $\mathcal{C}_M$ *represents the aggregated transactional property of the resulting Transactional Composite WS and is updated each time a transition is fired.*

Definition 7 Transactional Composite WS for Q. *A* $WSDN_Q$ *is a 4-tuple* (A_Q, S_Q, F_Q, ξ_Q)*, where:*

- $A_Q \subseteq A \mid I_Q \subseteq A_Q \wedge O_Q \subseteq A_Q$;
- $S_Q \subseteq S$;
- $F_Q : (A_Q \times S_Q) \cup (S_Q \times A_Q) \rightarrow \{0,1\}$ *is a flow relation indicating the presence* (1) *or the absence* (0) *of arcs between places and transitions defined as follows:*
  $\forall s \in S_Q$, $(\exists a \in A_Q \mid F_Q(a,s) = 1$ *if* $F(a,s) = 1)$ *and* $\forall s \in S_Q$, $(\exists a \in A_Q \mid F_Q(s,a) = 1$ *if* $F(s,a) = 1)$;
- ξ_Q *is a color function such that* $\xi_Q : S_Q \rightarrow \Sigma_S$ *and* $\Sigma_S = \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ *represents the TP of* $s \in S$.

The global transactional property of $WSDN_Q$ ensures that if a component WS, whose transactional property does not allow forward recovery fails, then all previous executed WSs can be semantically recovered by a backward recovery. For modeling the compensation flow of a Transactional Composite WS, we formally define BR_WSDN_Q as follows.

Definition 8 Backward Recovery Net.
A BR_WSDN_Q*, associated to a given* $WSDN_Q$=$(A_Q$, $S_Q, F_Q, \xi_Q)$*, is a 4-tuple* $(A'$, S', F^{-1}, $\zeta)$*, where:*

- A' *is a finite set of places corresponding to the* $WSDN_Q$ *places such that:* $\forall a' \in A'$ $\exists a \in A_Q$ *associated to* a' *and* a' *has the same semantic of* a.
- S' *is a finite set of transitions corresponding to the set of compensation WSs in* $WSDN_Q$ *such that:* $\forall s \in S_Q$, $\xi_Q(s) \in \{c, cr\}, \exists s' \in S'$ *which compensate* s.
- F^{-1}:$(A_Q \times S_Q) \cup (S_Q \times A_Q) \rightarrow \{0,1\}$ *is a flow relation establishing the restoring order in a backward recovery defined as:* $\forall s' \in S'$ *associated to* $s \in S_Q$, $\exists$ $a' \in A'$ *associated to* $a \in A_Q \mid F^{-1}(a', s') = 1 \Leftrightarrow F(s,a) = 1$ *and* $\forall s' \in S'$, $\exists a' \in A' \mid F^{-1}(s', a') = 1 \Leftrightarrow F(a,s) = 1$.

- *ζ is a color function such that $\zeta : S' \to \Sigma'_S$ and $\Sigma'_S = \{In, Ru, Ex, Co, Fa, Ab\}$ represents the execution state of $s \in S_Q$, and $s' \in S'$ is its compensation WS (In: initial, Ru: running, Ex: executed, Co: compensated, Fa: Failed, and Ab: abandoned).*

We consider that a WS should be selected at most once, therefore the corresponding transition in $WSDN$ should be fired only one time. As a consequence, when a transition s is fired, tokens are added to its output places and all tokens are deleted from its input places (except from places that belong to O_Q). If s is compensatable ($C_S(s) \in \{c, cr\}$), its corresponding compensation WS, s', has to be added in the compensation CPN. Each WS s includes in its semantic description the corresponding reference to its compensatable service s'; s and s' are registered in the same *Registry*. Figure 6 represents the transactional rules of Composite WSs defined by [17]. Its states model all the possible transactional property (TP) of a composite WS and whose arc labels model the TP of the component WSs. Label "$;t$" represents a sequential execution and label "$//t$" represents a parallel execution with a WS whose TP is t ($t \in \Sigma_S$). For example, in condition 3, the marking of places $a \in (A - {}^\bullet s)$ is deduced from the arcs labeled by "$//t$", with $t \in \{\boldsymbol{a}r, cr\}$, leaving from state $\boldsymbol{a}r$ (a pr or $\boldsymbol{a}r$ WS can only be executed in parallel with retriable WSs). On the other hand, there is no condition on the marking of places ($x \in {}^\bullet s$), when the color of s is pr or $\boldsymbol{a}r$, because there is an arc with label "$;pr$" or "$;\boldsymbol{a}r$" leaving from all the final states of the automaton (any WS can be executed in sequential with a (previous) pr or $\boldsymbol{a}r$ WS). Following rules formalize these actions.

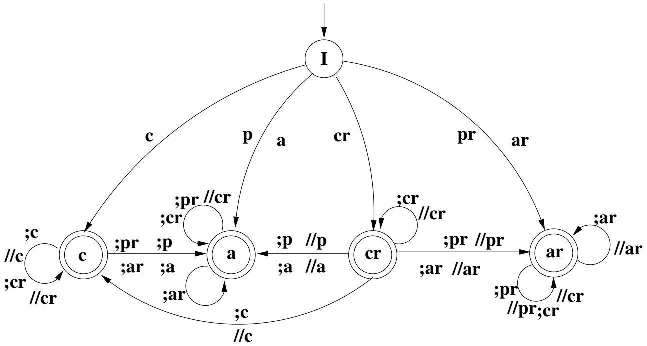


Fig. 6. Transactional rules of Composite WSs defined by [17]

Definition 9 Firing Rules.

The firing of a fireable transition s for a marking M defines a new marking M', denoted as $M \xrightarrow{s} M'$, such that:

1. *Tokens are added to the output places of s depending on the color of s and on the color of the tokens contained by the input places of s, according to the following rules (see Figure 6):*

```
if ( $\exists x \in ({}^\bullet s) \mid \boldsymbol{a} \in M(x)$ ), then $\forall y \in (s^\bullet),\ M'(y) \leftarrow M'(y) \cup \{\boldsymbol{a}\}$
  else if ( $\exists x \in ({}^\bullet s) \mid \boldsymbol{ar} \in M(x)$ ),
      then $\forall y \in (s^\bullet),\ M'(y) \leftarrow M'(y) \cup \{\boldsymbol{ar}\}$
    else if [ $(\exists x \in ({}^\bullet s) \mid c \in M(x)) \wedge (C_S(s) \in \{p, pr, \boldsymbol{a}, \boldsymbol{ar}\})$ ],
        then $\forall y \in (s^\bullet),\ M'(y) \leftarrow M'(y) \cup \{\boldsymbol{a}\}$
      else if [ $(\exists x \in ({}^\bullet s) \mid c \in M(x)) \wedge (C_S(s) \in \{c, cr\})$ ],
          then $\forall y \in (s^\bullet),\ M'(y) \leftarrow M'(y) \cup \{c\}$
        else /*in this case: $\forall x \in ({}^\bullet s),\ M(x) \in Bag(\{I, cr\})$*/
          $\forall y \in (s^\bullet),\ M'(y) \leftarrow (M'(y) \cup C_S(s))$
            if $C_S(s) \in \{\boldsymbol{a}, \boldsymbol{ar}, c, cr\}$,
              $M'(y) \leftarrow (M'(y) \cup \{\boldsymbol{a}\})$ if $C_S(s) = p$,
              and $M'(y) \leftarrow M'(y) \cup \{\boldsymbol{ar}\}$ if $C_S(s) = pr$
```

2. *Tokens are deleted from input places of s, if they do not belong to O_Q:*
   $\forall x \in ({}^\bullet s - O_Q),\ M(x) \leftarrow \emptyset$,
3. *Color $\mathcal{C}_{M'}$ of the resulting $(WSDN, M')$ (see Def. 6) is updated, according to the following rules (see Figure 6):*

```
if $(\mathcal{C}_M \in \{I, cr\})$ and $C_S(s) = p$ then $\mathcal{C}_{M'} \leftarrow \boldsymbol{a}$
  else if $(\mathcal{C}_M \in \{I, cr\})$ and $C_S(s) = pr$ then $\mathcal{C}_{M'} \leftarrow \boldsymbol{ar}$
    else if $(\mathcal{C}_M \in \{I, cr\})$ and $C_S(s) \in \{\boldsymbol{a}, \boldsymbol{ar}, c, cr\}$
        then $\mathcal{C}_{M'} \leftarrow C_S(s)$
      else if $(\mathcal{C}_M = c\})$ and $C_S(s) \in \{p, pr, \boldsymbol{a}, \boldsymbol{ar}\}$
          then $\mathcal{C}_{M'} \leftarrow \boldsymbol{a}$
        else $\mathcal{C}_{M'} \leftarrow \mathcal{C}_M$
```

4. *s and its input and output places are added to $WSDN_Q$=(A_Q, S_Q, F_Q, ξ_Q) (Def. 7) as:*
   $A_Q \leftarrow A_Q \cup {}^\bullet s \cup s^\bullet$; $S_Q \leftarrow S_Q \cup \{s\}$; $F_Q(a, s) \leftarrow 1,\ \forall a \in {}^\bullet s$; $F_Q(s, a) \leftarrow 1,\ \forall a \in s^\bullet$
5. *If $C_S(s) \in \{c, cr\}$, its compensation WS s' and its input and output places are added to $BR\_WSDN_Q = (A', S', F^{-1}, \zeta)$ (see Def.8), according to the following rules:*
   $A' \leftarrow A' \cup \{a' \mid \exists\ a \in A_Q \wedge a \in {}^\bullet s \vee a \in s^\bullet\ \}$; $S' \leftarrow S' \cup \{s'\}$; $F^{-1}(a', s') = 1 \Leftrightarrow F_Q(s, a) = 1$; $F^{-1}(s', a') = 1 \Leftrightarrow F_Q(a, s) = 1$

Associated to $WSDN_Q$, there exists a firing sequence $\sigma = \{s_1, \ldots, s_n \mid s_i \in S_Q\}$, such that: $M_Q \xrightarrow{\sigma} M_F$, where M_Q is the initial marking (see Def. 4) and M_F denotes the desired marking in which $\forall o \in O_Q, M_F(o) \neq \emptyset$.

When several transitions are fireable, to select which transition has to be fired, we propose a quality measure of a transition s which depends on the user query Q such that:

Definition 10 Quality associated with a transition. *The quality of a transition $s_i \in S$, called $Quality_Q(s_i)$, depends on the user query Q and is defined as:*

$$Quality_Q(s_i) = Score(s_i) \times g(C_S(s_i)) \times (card(O_Q \cap s_i^\bullet) + 1) \times \left(1 + \frac{card((s_i^\bullet)^\bullet)}{card(S)}\right)$$

with $Score(s_i) = \sum_j w_j \times q_j(s_i)$ *with* $(w_j, q_j) \in W_Q$ *and* $q_j(s_i)$ *the value of the QoS criterion* q_j *for the WS corresponding to* s_i*, and with* $g : \sum_S \to \mathbb{N}$*, a function such that:* $g(p) = g(\boldsymbol{a}) < g(pr) = g(\boldsymbol{a}r) < g(c) < g(cr)$.

Value $Score(s_i)$ *allows to evaluate the QoS of the WS corresponding to a transition s (the higher the score, the better WS QoS). Function g allows to select a transition whose transactional property is the less restrictive. An example of g is:* $g(p) = g(\boldsymbol{a}) = 1$*,* $g(pr) = g(\boldsymbol{a}r) = 2$*,* $g(c) = 3$*, and* $g(cr) = 4$*.* $(card(O_Q \cap s_i^\bullet) + 1)$ *gives more chance to select transitions producing more required outputs.* $\left(1 + \frac{card((s_i^\bullet)^\bullet)+1}{card(S)}\right)$ *increases the quality to those transitions which will allow more transitions to be fireable. If several transitions have the same* $Quality_Q$ *value, they can be randomly selected to be fired.*

Finally, the selection problem consists in discovering and selecting the WSs of the *Registry* whose composition satisfies the functional, QoS, and transactional requirements of the user, such that:

Definition 11 The WS Selection Problem (WSS Problem): *Given a user query* $Q = (I_Q, O_Q, W_Q, T_Q)$ *and a* $WSDN = (A, S, F, \xi)$*, the WSS Problem consists in creating a Colored Petri Net* $WSDN_Q = (A_Q, S_Q, F_Q, \xi_Q)$*, sub-part of* $WSDN$*, which satisfies Q and its corresponding compensation Colored Petri Net* $BR\_WSDN_Q = (A', S', F^{-1}, \zeta)$*, which provides the compensation flow, from the firing sequence* σ*, such that:* $M_Q \overset{\sigma}{\to} M_F$*, where* M_Q *is the initial marking and* M_F *is a reachable marking such that:* $\forall a \in (A_Q \cap O_Q)$*,* $M_F(a) \in \{c, cr\}$ *if* $T_Q = T_0$*, and* $\forall a \in (A_Q \cap O_Q)$*,* $M_F(a) \in \{\boldsymbol{a}, \boldsymbol{a}r, c, cr\}$ *if* $T_Q = T_1$*, and such that transitions of* σ *represents a Transactional Composite WS whose components locally optimize the QoS and* $\forall s_i \in \sigma \mid \xi_Q(s) \in \{c, cr\}$*, its corresponding compensation* s_i' *is in* BR_WSDN_Q *(*$s_i' \in S'$*).*

Based on this formalism, we have implemented the unrolling algorithm to automatically generate $WSDN_Q$ and BR_WSDN_Q[9].

4.2 Execution Phase: The EXECUTER

Once $WSDN_Q$ and BR_WSDN_Q are obtained, the Transactional Composite WS have to be executed ensuring a consistent state of the system in presence of failures. The execution process is managed by algorithms that execute the CPNs. We formally describe the execution process in following definitions.

The marking of a $WSDN_Q$ or BR_WSDN_Q represents the current values of attributes that can be used for some component WSs to be invoked or control values indicating the compensation flow, respectively. A Marked CPN denotes which transitions can be fired.

Definition 12 Marked Executable WSDN.
A marked executable WSDN =(A, S, F, ξ) *is a pair (CPN,M), where M is a function which assigns tokens (values) to places such that* $\forall a \in A$*,* $M(a) \in \mathbb{N}$*.*

Definition 13 Fireable Executable Transition.
A marking M enables a transition s iff all its input places contain tokens such that: $\forall x \in ({}^{\bullet}s), M(x) \geq card({}^{\bullet}x)$

During the execution, in $WSDN_Q$ and BR_WSDN_Q, a transition is fireable (its corresponding WS can be invoked) only if all its predecessor transitions have been fired. Note that execution only concerns with the number of tokens in places, instead of their colors. For both, $WSDN_Q$ and BR_WSDN_Q, colors are considered in transitions but not in places. Considering only number of tokens in places is enough to respect sequential and parallel executions, which in turn keep the global transactional property. We illustrate this definition with the example shown in Figure 7. Note that ws_3 needs two tokens in a_3 to be invoked; this data flow dependency indicates that it has to be executed in sequential order with ws_1 and ws_2, and can be executed in parallel with ws_4. Note that a_3 is produced by ws_1 and ws_2, ws_1 was already executed and it produced a token on a_3, and ws_2 is still running. Even if ws_3 could be invoked with the values produced by ws_1, if ws_3 is fired, it will be executed in parallel with ws_2; however, it could be possible that transactional properties of ws_2 and ws_3 dictates that they have to be executed in sequential order as the data flow indicates. Then, ws_3 has to wait all its predecessors transitions finish to be invoked. Once ws_2 finishes, ws_3 and ws_4 can be executed in parallel. This example illustrates a scenario in which the data flow controls the execution flow in order to maintain the global transactional property. In other scenarios, data and execution flows coincide.

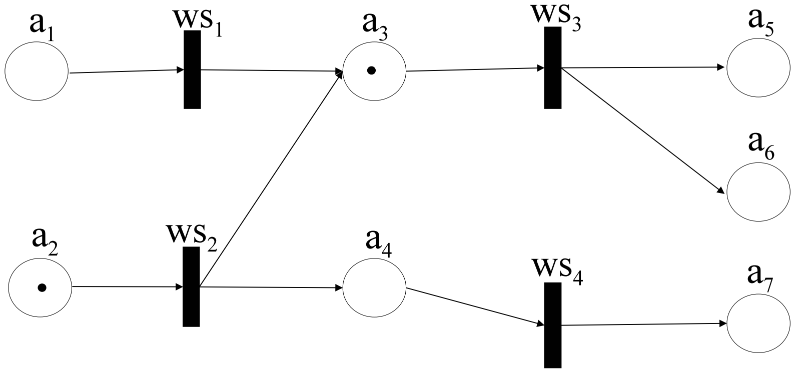


Fig. 7. Example of Fireable Transitions

The execution control of a Transactional Composite WS is guided by an algorithm executing its corresponding $WSDN_Q$=(A_Q, S_Q, F_Q, ξ_Q).

To support backward recovery, it is necessary to keep the trace of the execution on the BR_WSDN_Q=(A', S', F^{-1},ζ).

To start the execution algorithm, the $WSDN_Q$ is marked with the *Initial Marking*: an initial token is added only to places representing inputs of Q ($\forall a \in (A_Q \cap I_Q)$, $M(a) = 1$, $\forall a \in (A_Q - I_Q)$, $M(a) = 0$) and the state of all transitions in BR_WSDN_Q is set to *initial* ($\forall s' \in S'$, $\zeta(s') \leftarrow In$).

The firing of a transition of a $WSDN_Q$ corresponds to the execution of a WS, let say s, which participates in the composition. While s is executing, the state of its corresponding s' in BR_WSDN_Q is set on *running* ($\zeta(s') \leftarrow Ru$). When s finishes its s' is set to *executed* ($\zeta(s') \leftarrow Ex$), others transitions become fireable, and the following firing rules are applied.

Definition 14 $WSDN_Q$ **Firing Rules.**
The firing of a fireable transition s for a marking M defines a new marking M', such that: all tokens are deleted from its input places ($\forall x \in {}^\bullet s$, $M(x) = 0$), if the $\xi_Q(s) \in \{c, cr\}$, the state of its corresponding s' in BR_WSDN_Q is set to running ($\zeta(s') \leftarrow Ru$), and the WS *s is invoked. After s finishes, tokens are added to its output places ($\forall x \in (s^\bullet)$, $M(x) = M(x) + card(x^\bullet)$), and the state of its corresponding s' in BR_WSDN_Q (if it exists) is set to executed ($\zeta(s') \leftarrow Ex$).*

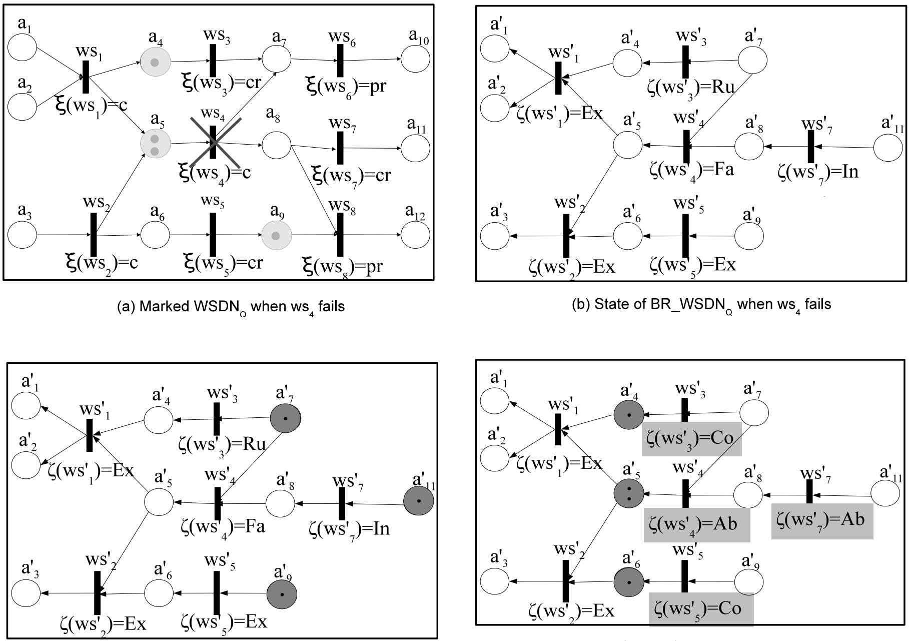
In case a WS s fails, if $\xi_Q(s) \in \{pr, \boldsymbol{a}r, cr\}$, s is re-invoked until it successfully finishes (forward recovery); otherwise, its corresponding s' in BR_WSDN_Q (if it exists) is set to *failed* ($\zeta(s') \leftarrow Fa$) and a backward recovery is needed, i.e., all executed WSs must be compensated in the inverse order they were executed; for parallel executed WSs, the order does not matter. Backward recovery implies the execution of $WSDN_Q$ is halted and the compensation process is initiated over BR_WSDN_Q with its *Initial Marking*: tokens are added to places representing inputs of BR_WSDN_Q ($\forall a' \in A' \mid {}^\bullet a' = \emptyset$, $M(a') = card(a'^\bullet)$) and other places has no tokens. The execution of BR_WSDN_Q is guided by Def. 15 and Def. 16.

Definition 15 Fireable Compensation Transition.
A marking M enables a transition s' iff all its input places contain tokens such that $\forall a' \in ({}^\bullet s')$, $M(a') \neq 0$, $\wedge$ $\zeta(s') \notin \{Co, Ab\}$.

Definition 16 BR_WSDN_Q **Firing Rules.**
The firing of a fireable transition (see Def. 15) s' for a marking M defines a new marking M', such that:

- *if $\zeta(s') = In$, $\zeta(s') \leftarrow Ab$ (i.e., the corresponding s is abandoned before its execution),*
- *if $\zeta(s') = Fa$, $\zeta(s') \leftarrow Ab$ (i.e., the corresponding s is abandoned, it has failed),*
- *if $\zeta(s') = Ru$, $\zeta(s') \leftarrow Co$ (in this case s' is executed after s finishes, then s is compensated),*
- *if $\zeta(s') = Ex$, $\zeta(s') \leftarrow Co$ (in this case s' is executed, i.e., s is compensated),*
- *tokens are deleted from its input places ($\forall x \in {}^\bullet s'$, $M(x) = M(x) - 1$) and tokens are added to its output places as many successors it has ($\forall x \in (s'^\bullet)$, $M(x) = card(x^\bullet)$).*

Figure 8 illustrates a backward recovery. The marked $WSDN_Q$ depicted in Figure 8(a) and the BR_WSDN_Q depicted in Figure 8(b) represent the execution state when ws_4 fails, the execution of $WSDN_Q$ is halted, and the initial marking on BR_WSDN_Q is set to start its execution process (Figure 8(c)), after ws'_3 and ws'_5 are fired to compensate ws_3 and ws_5 respectively, and ws_4 and ws_7 are abandoned before its invocation (ws_4 has failed and ws_7 was not invoked before the failure), a new marking is produced (Figure 8(d)), in which ws'_1 and ws'_2 are both fireable and can be invoked in parallel. Note that only compensatable transitions have their corresponding compensation transitions in BR_WSDN_Q; in this example ws_6 and ws_8 are not compensatable, in consequence, they and their corresponding output attributes (a_{10} and a_{12}) do not appear in BR_WSDN_Q.



(a) Marked $WSDN_Q$ when ws_4 fails

(b) State of BR_WSDN_Q when ws_4 fails

(c) Initial marking of BR_WSDN_Q

(d) Marked BR_WSDN_Q after ws'_3 and ws'_5 were invoked and ws_4 and ws_7 were abandoned

Fig. 8. Example of Backward Recovery

If a failure occurs in an advanced execution point, a backward recovery may incur in high wasted resources. On the other hand, it is hard to provide a **retriable** Transactional Composite WS, in which all its components are **retriable** to guaranty forward recovery. We proposed an approach based on WS substitution in order to try forward recovery [11]. In this context, we deal with *service classes* [2], which group WSs with the same semantic functionality, i.e., WSs providing the same operations but having different WSDL interfaces (input and output attributes), transactional support, and QoS. We suppose that *service classes* are previously defined and are specified in the semantic descriptions in the *Registry.* When a WS fails, if it is not **retriable**, instead of backward recovery, a substitute WS is automatically searched to be executed on behalf of the faulty WS.

Definition 17 Functional Substitute.
Let SC be a service class, if s, $s^* \in SC$, we say that s is a functional substitute of s^* (denoted as $s \equiv_F s^*$), if $({}^\bullet s^*) \subseteq ({}^\bullet s) \wedge (s^*)^\bullet \supseteq (s^\bullet)$.

Definition 18 Exact Functional Substitute.
Let SC be a service class, if s, $s^* \in SC$, we say that s is an exactly functional substitute of s^* (denoted as $s \equiv_{EF} s^*$), if $({}^\bullet s^*) = ({}^\bullet s) \wedge (s^*)^\bullet = (s^\bullet)$.

In a *service class*, the functional equivalence is defined according to the WSs input and output attributes. A WS s is a Functional Substitute of another WS s^*, if s^* can be invoked with at most the input attributes of s and s^* produces at least the same output attributes produced by s. They are exactly functional substitutes if they have the same input and output attributes. Figure 9 illustrates several examples: $ws_1 \equiv_F ws_2$, however $ws_2 \not\equiv_F ws_1$, because ws_1 does not produce output a_5 as ws_2 does. $ws_1 \equiv_F ws_3$, $ws_3 \equiv_F ws_1$, and also $ws_1 \equiv_{EF} ws_3$.

In order to guarantee the global transactional property of the Transactional Composite WS, a WS s can be replaced by another WS s^*, if s^* can behave as s in the recovery process. Hence, if $\xi_Q(s)=p$, in which case s only allows backward recovery, it can be replaced by any other WS because all transactional properties allow backward recovery. A WS with $\xi_Q(s) = pr$ can be replaced by any other **retriable** WS (pr,$\boldsymbol{a}r$,cr), because all of them allow forward recovery. An $\boldsymbol{a}$ WS allows only backward recovery, then it can be replaced by another WS which provides backward recovery. A c WS can be replaced by a WS that provides semantic recovery as c and cr WSs. While a cr WS can be only replaced by another cr WS because allows forward, backward, and semantic recovery. Then, we have:

Definition 19 Transactional Substitute.
Let SC be a service class, if s, $s^* \in SC$, we say that s is a transactional substitute of s^* (denoted as $s \equiv_T s^*$):

- *if* $(\xi_Q(s) = p, \xi_Q(s^*) \in \{p, pr, \boldsymbol{a}, \boldsymbol{a}r, c, cr\}) \wedge (s \equiv_F s^*)$,
- *if* $(\xi_Q(s) = pr, \xi_Q(s^*) \in \{pr, \boldsymbol{a}r, cr\}) \wedge (s \equiv_F s^*)$,
- *if* $(\xi_Q(s) = \boldsymbol{a}, \xi_Q(s^*) \in \{\boldsymbol{a}, \boldsymbol{a}r, c, cr\}) \wedge (s \equiv_F s^*)$,
- *if* $(\xi_Q(s) = \boldsymbol{a}r, \xi_Q(s^*) \in \{\boldsymbol{a}r, cr\}) \wedge (s \equiv_F s^*)$,
- *if* $(\xi_Q(s) = c, \xi_Q(s^*) \in \{c, cr\}) \wedge (s \equiv_{EF} s^*)$,
- *if* $(\xi_Q(s) = cr, \xi_Q(s^*) = cr) \wedge (s \equiv_{EF} s^*)$.

In Figure 9, $ws_1 \equiv_T ws_2$, because $ws_1 \equiv_F ws_2$ and $\xi_Q(ws_2) = cr$, then ws_2 can behave as a pr WS; however $ws_1 \not\equiv_T ws_3$, even $ws_1 \equiv_F ws_3$, because as $\xi_Q(ws_3) = p$, w_3 cannot behave as a pr WS. Transactional Substitute definition allows WSs substitution in case of failures.

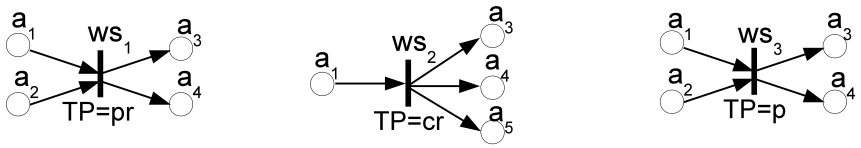


Fig. 9. Example of Equivalent WSs

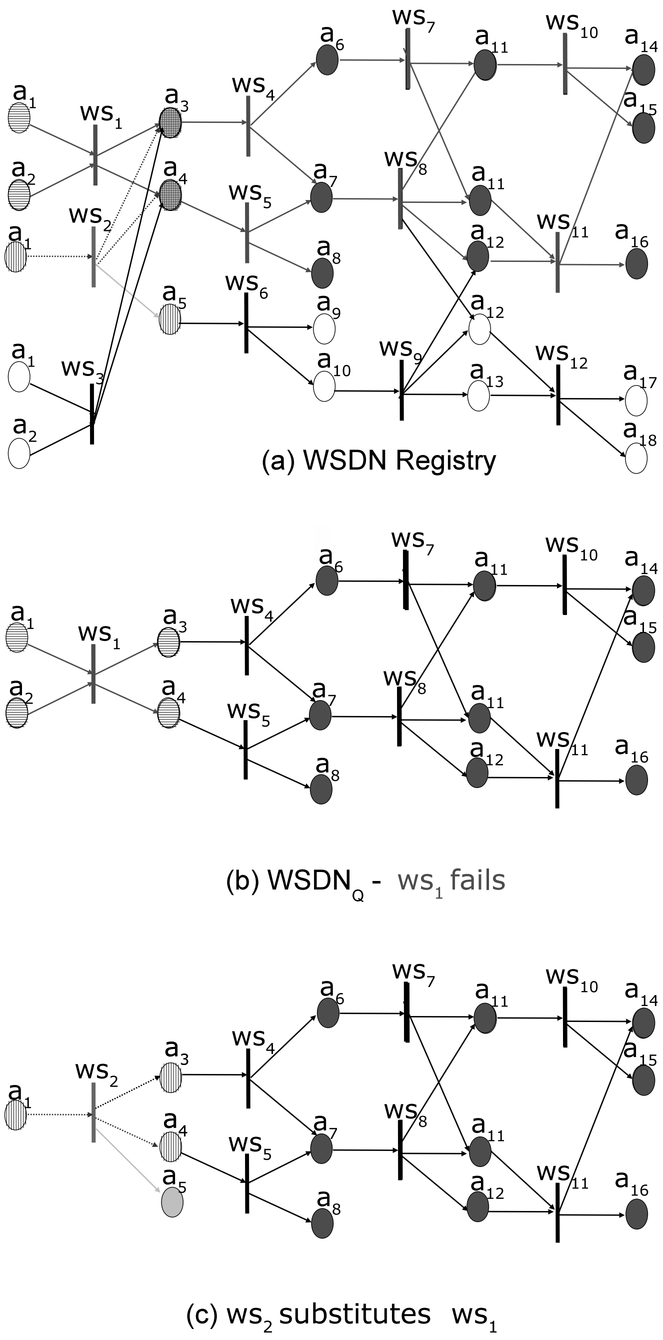
Definition 20 WSs Substitution.
Let $WSDN_Q$=(A_Q, S_Q, F_Q, ξ_Q) be the CPN allowing the execution of a Transactional Composite WS that satisfies the Query $Q = (I_Q, O_Q, W_Q, T_Q)$, and BR_$WSDN_Q$=(A', S', F^{-1}, ζ) its corresponding backward recovery CPN. In case of a WS $s \in S_Q$ fails, it can be replaced by another s^*, if $s \equiv_T s^*$, and the following actions proceed:

1. $S_Q \leftarrow S_Q \cup \{s^*\}$;
2. $\forall a \in {}^\bullet(s^*)$, $F(a, s^*) = 1 \wedge$ $\forall a \in {}^\bullet s$, $F(a, s) = 0$;
3. $\forall a \in s^\bullet, F(s^*, a) = 1, F(s, a) = 0$;
4. $S_Q \leftarrow S_Q - \{s\}$;
5. *if $\xi_Q(s) \in \{c, cr\}$, $s' \in S'$ is replaced by s'^* (it compensates s^*) applying 1, 2, 3, and 4 on BR_$WSDN_Q$.*

When a substitution occurs, the faulty WS s is removed from the $WSDN_Q$, the new s^* is added, but we keep the original sequential relation defined by the input and output attributes of s. In that way, the $WSDN_Q$ structure, in terms of sequential and parallel WSs, is not changed. For compensatable WSs, it is necessary exact functional substitutes to do not change the compensation control flow in the respective *BR_*$WSDN_Q$. The idea is to try to finish the Transactional Composite WS execution with the same properties of the original one. In Figure 10(a), there is a CPN representing a *Registry*, Figure 10(b) shows a $WSDN_Q$, and Figure 10(c) illustrates the resulting $WSDN_Q$ after ws_1 was replaced by ws_2. Note that the output a_5 produced by ws_2 does not belong to the $WSDN_Q$, because it is not produced by the original ws_1, and the execution order is not modified.

When in a *service class* there exist several WSs candidates for replacing a faulty s, it is selected the one with the best quality measure.The quality of a transition depends on the user query Q and on its QoS values. WSs Substitution is done such that the substitute WS locally optimize the QoS. A similar quality measure used by the COMPOSER is used during the execution, in order to keep the same heuristic to select substitutes.

Summarily, in case of failure of a WS s, depending on the color of transition ($\xi_Q(s)$), the following actions could be executed: (i) if $\xi_Q(s)$ is **retriable** (pr, $\boldsymbol{ar}$, cr), s is re-invoked until it successfully finish (forward recovery); (ii) otherwise, another Transactional Substitute WS, s^*, is selected to replace s,



(a) WSDN Registry

(b) WSDN_Q - ws_1 fails

(c) ws_2 substitutes ws_1

Fig. 10. Example of of WSs substitution

and the execution algorithm goes on (trying a forward recovery); (iii) if there not exist any substitute s^*, its corresponding s' in BR_WSDN_Q (if it exists) is set to *failed* ($\zeta(s') \leftarrow F$) and a backward recovery is needed, i.e., all executed WSs must be compensated in the inverse order they were executed; for parallel executed WSs, the order does not matter.

4.3 Experimental Results

We have evaluated our COMPOSER and EXECUTER with some experiments. Following sections describe the obtained results.

a. COMPOSER

In order to evaluate our COMPOSER [9], experiments were executed on a PC with 2 Dual Core P8600 with 2.4GHz each one, and 3GB RAM, Windows Vista, JDK 1.6 virtual machine was used to develop and run the programs. We have conducted experiments in two scenarios. The first scenario compares the global QoS of the TCWS returned by our algorithm with the best solution which is obtained by an exhaustive implementation of the algorithm. The second one evaluates the execution time of our solution. In the first scenario, due to the execution time of the exhaustive algorithm, the number of WSs in the *Registry* varies from 10 to 50, while in the second one it varies from 100 to 500. In both scenarios, each WS has between 1 and 5 inputs and between 1 and 3 outputs, randomly generated from an ontology containing 20 generated elements. Each value of each QoS criterion and of each transactional property have been randomly generated for each WS. In order to model the fact that a WS can appear or disappear, each WS Registry is generated independently from the previous generated one. In both experiments, 10 user queries are randomly generated by varying the number of inputs and the number of outputs between 1 and 3 and by randomly generating the weights over QoS criteria. These queries were executed on each scenario for each number of services with both required transactional property values T_0 and T_1.

The results of our first scenario indicate that our algorithm finds the best solution in 82.86%, considering all *Registry* sizes. For *Registry* with 10 to 40 services our algorithm finds solution for all queries where at least one solution exists. While in *Registry* with 50 services in 10% of queries our algorithm did not found the existing solution (representing less then 1% of total number of queries). Figure 11 shows the results of the second scenario, when the required transactional property is T_0 or T_1, and when the transactional properties are not taken into account (NT in the graph for Non Transactional). As shown in the figure, taking into account the transactional properties does not have a real impact on the execution time. From a theoretical point of view, the complexity of our algorithm is $\mathcal{O}(card(S)^2)$, where $card(S)$ is the number of WSs in the *Registry*. This is confirmed by our experimental results.

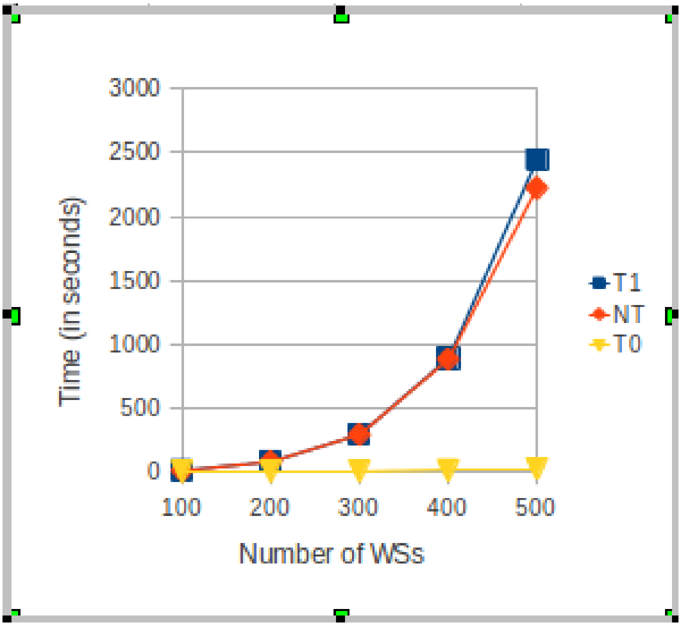


Fig. 11. *Execution time of CPN-TCWS selection algorithm [9]*

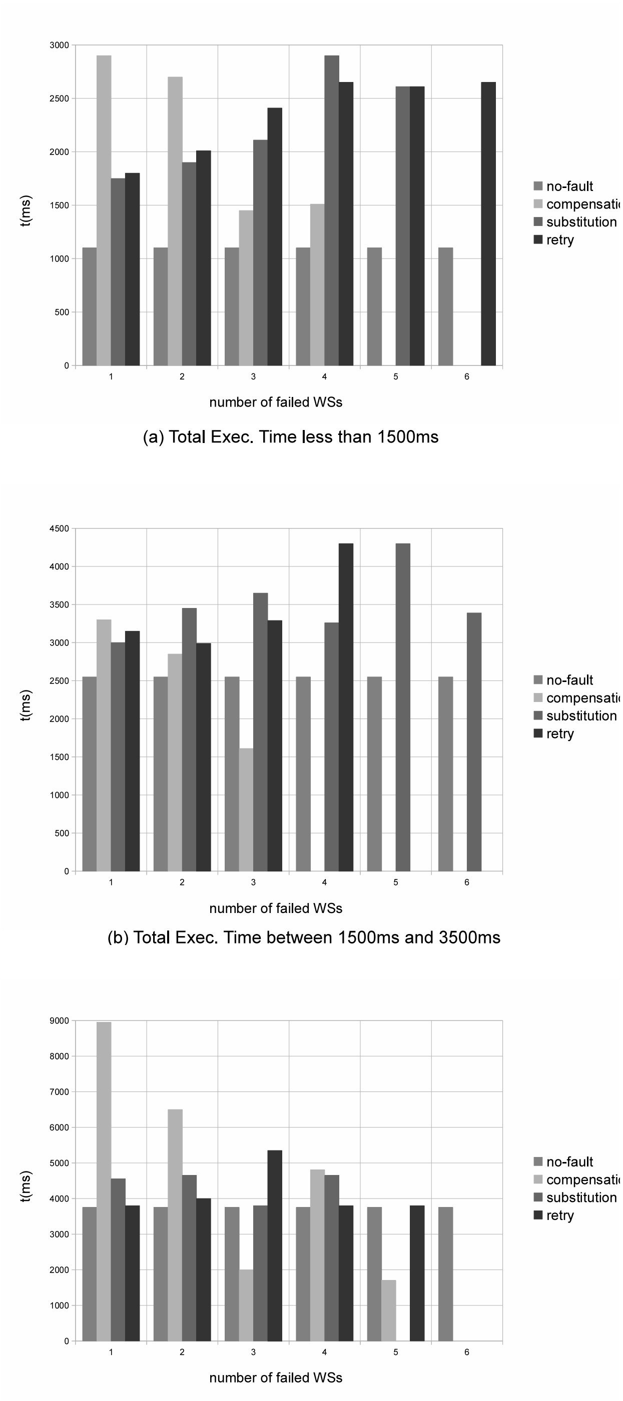
b. EXECUTER

We developed a prototype of our EXECUTER [1], using Java 6 and MPJ Express 0.38 library to allow the execution in distributed memory environments. We deployed our EXECUTER in a cluster of PCs: one node for the Execution Engine and one node for each Engine Thread needed to execute the TCWS. All PCs have the same configuration: Intel Pentium 3.4GHz CPU, 1GB RAM, Debian GNU/Linux 6.0, and Java 6. They are connected through a 100Mbps Ethernet interface.

We generated 10 **compensatable** TCWSs. All those TCWSs were automatically generated by our COMPOSER [9], from synthetic datasets comprised by 6400 WSs. Each WS is annotated with a set of *QoS* parameters, however for our experiments we only consider the response time as the *QoS* criteria. Replicas of WSs have different response times. The OWLS-API 3.0 was used to parse the WS definitions and to deal with the OWL classification process.

The first group of experiments were focused on a comparative analysis of the recovery techniques. The second group of experiments evaluates the overhead incurred by our framework in control operations to perform the execution of a TCWS and to execute the fault tolerant mechanisms.

To simulate unreliable environments, we define five different conditions wherein all WSs have the same probability of failure: 0.2, 0.15, 0.1, 0.005, and 0.001. The executions on these unreliable environments were done in three scenarios to support the fails: *(i)* backward recovery (compensation, second bars on each group



(a) Total Exec. Time less than 1500ms

(b) Total Exec. Time between 1500ms and 3500ms

(c) Total Exec. Time more than 3500ms

Fig. 12. Executions on the unreliable environments

in Figure 12), *(ii)* forward recovery because all WSs are retriable (retry, fourth bars on each group in Figure 12), and *(iii)* forward recovery (substitution, third bars on each group in Figure 12). On each scenario all TCWSs were executed 10 times.

Each TCWS was also executed 10 times in a reliable environment, in which all WSs have 0 as probability of failures (no-faulty, first bars on each group in Figure 12) in order to classify them according to their average total execution time in three groups: less than 1500ms (Figure 12(a)), (ii) between 1500ms and 3500ms (Figure 12(b), and (more than 3500ms (Figure 12(c)).

In Figure 12 we plot the average of the total execution time according to the number of failed WSs, in order to compare all recovery techniques. The results show that when the number of failed WSs is small (i.e., the probability of failures is less than 20%) backward recovery (compensation) is the worst strategy because almost all component WSs had been executed and have to be compensated. Moreover, when the average of the total execution time of TCWSs is high (greater than 1500ms) forward recovery with retry strategy is better than forward recovery with substitution due to the substitute normally has a bigger response time than the faulty WS. By the other side, in cases in which the probability of failure is greater than 30%, backward recovery whit compensation behaves better than the other ones (even the final results is not produced) because there are many faulty services and only few have to be compensated.

Another issue that can be observed it is the number of outputs received before the backward recovery mechanism has to be executed. In this experiment, the average percentage of outputs received before compensation was 37%. All these outputs are lost or delivered as a set of incomplete (and possibly meaningless and useless) outputs to the user. This percentage is related to the average percentage of compensated services, which is 80%, confirming the overhead, the possible unfulfillment of *QoS* requirements, and the lost outputs. Definitely, backward recovery should be executed only in absence of another alternative, at early stages of execution of a TCWS, or high faulty environments.

To measure the intrusiveness of our EXECUTER incurred by control activities, we execute the same set of experiments describe above, but we set to 0 the response time of all WSs. Table 5 shows the average overhead under all different scenarios.

Table 5. Average overhead incurred by the EXECUTER

	Average Overhead (ms)	% overhead increased
No Fault	611.7	
Compensation	622.38	2%
Substitution	612.82	0.2%
Retry	612.01	0.05%

The average overhead of the EXECUTER only depends on the number of components WSs in a TCWS. It does not depend on the total response time of TCWS. It means that while the total response time is higher the overhead

% will decrease. It is clear that the reason behind the backward recovery strategy overhead (increased by 2%) is the amount of coordination required to start the compensation and the fact that a new WS execution (the compensation WS execution) has to be performed for each successfully executed WS, in order to restore the consistent system state. Additionally, the compensation process has to be done following the unfolding algorithm of the respective BRCPN-$TCWS_Q$.

We do not consider to wait before the retry of a failed WS execution; therefore, the increased overhead of retry a WS is almost imperceptible.

As the *service class* for each WS is sent by the Execution Engine in the *Initial* phase, each Engine Thread has the list of the equivalent WSs sorted according to their quality, then there is virtually no overhead when searching for an equivalent WS to replace a faulty one.

Based on the results presented above, we can conclude that our EXECUTER efficiently implements fault tolerant strategies for the execution of TCWSs with admissible small overhead.

5 Conclusions

WS composition problem consists in combining several available WSs to satisfy complex user queries, taking into account functional requirements, QoS criteria, and transactional properties.

It implies several phases to decide *how* and *which* WSs will participate in the Composite WS. These phases gain advantages of using formal models as Petri Nets. Contributions of this article are twofold: a survey and a classification of Petri Nets-based WS composition approaches and a description of our contribution in transactional and QoS driven WS composition based on Colored Petri Nets. Petri Nets are a powerful tool for modeling and analyzing static vision and dynamic behavior of a wide variety of systems. The most broadly use of Petri Nets in WS composition is motivated by their analytical power to analyze model properties, e.g., reachability, safety, deadlock freeness, liveness. Indeed, the classification underlines the use in the literature of Petri Nets essentially to validate compatibility and interoperability of services, to verify the correctness of the composition, and to evaluate QoS metrics of the composition. Meanwhile, only few works exist for the use of Petri Nets in the specification, selection, and execution phases. We conclude this article with some challenges that still remain open problems.

At the specification phase, existing approaches either use classical Petri Nets to model WS composition considering only functional requirements [20], or use Colored Petri Net to model type of information [57], or uses Timed Petri Nets to model time constrained [55]. However, all these approaches do not consider non-functional properties such as QoS and transactional properties at the same time.

At the automatic selection phase, the biggest effort has been done to take into account functional and non-functional requirements either with classical Petri Nets [6,24] or with Colored Petri Nets [41]. Theses approaches compose WSs effectively but do not consider the potential restriction of executing order among services. Only [9] have proposed a Colored Petri Net model to automatic selection considering transactional and QoS properties as well as parallel execution order between services.

At the execution phase, there is a lack in execution engines or frameworks based on Petri Nets. Existing approaches focus on handling failures of WSs in a Composite WS either based on compensation paired Petri Net [35] or based on WSs transactional properties modeled in Colored Petri Nets [11,12].

It is clear that Petri Nets are well suited for supporting the whole WS composition process and enriching the modeling, verification, selection, and execution of Composite WSs

References

1. Angarita, R., Cardinale, Y., Rukoz, M.: FaCETa: Backward and Forward Recovery for Execution of Transactional Composite WS. In: Proc. of Fifth International Workshop on REsource Discovery (RED 2012), Heraklion, Grece, pp. 1–15 (2012)
2. Azevedo, V., Mattoso, M., Pires, P.: Handling Dissimilarities of Autonomous and Equivalent Web Services. In: Proc. of Caise-WES (2003)
3. Badr, Y., Benslimane, D., Maamar, Z., Liu, L.: Guest Editorial: Special Section on Transactional Web Services. IEEE T. Services Computing 3(1), 30–31 (2010)
4. Behl, J., Distler, T., Heisig, F., Kapitza, R., Schunter, M.: Providing Fault-tolerant Execution of Web-service based Workflows within Clouds. In: Proc. of the 2nd Internat. Workshop on Cloud Computing Platforms, CloudCP (2012)
5. Ben Lakhal, N., Kobayashi, T., Yokota, H.: FENECIA: failure endurable nested-transaction based execution of composite Web services with incorporated state analysis. VLDB Journal 18(1), 1–56 (2009)
6. Blanco, E., Cardinale, Y., Vidal, M.-E.: Aggregating Functional and Non-Functional Properties to Identify Service Compositions, pp. 1–36. IGI BOOK (2011)
7. Bonchi, F., Brogi, A., Corfini, S., Gadducci, F.: Compositional Specification of Web Services Via Behavioural Equivalence of Nets: A Case Study. In: van Hee, K.M., Valk, R. (eds.) PETRI NETS 2008. LNCS, vol. 5062, pp. 52–71. Springer, Heidelberg (2008)
8. Bushehrian, O., Zare, S., Keihani Rad, N.: A Workflow-Based Failure Recovery in Web Services Composition. Journal of Software Engineering and Applications 5, 89–95 (2012)
9. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: CPN-TWS: a coloured petri-net approach for transactional-QoS driven Web Service composition. IJWGS 7(1), 91–115 (2011)
10. Cardinale, Y., El Haddad, J., Manouvrier, M., Rukoz, M.: Transactional-aware Web Service Composition: A Survey. IGI Global - Advances in Knowledge Management (AKM) Book Series, ch. 6, pp. 116–141 (2011)
11. Cardinale, Y., Rukoz, M.: Fault Tolerant Execution of Transactional Composite Web Services: An Approach. In: Proceedings UBICOMM, Lisbon, Portugal, pp. 1–6 (2011)
12. Cardinale, Y., Rukoz, M.: A framework for reliable execution of transactional composite web services. In: MEDES, pp. 129–136 (2011)
13. Chi, Y.-L., Lee, H.-M.: A formal modeling platform for composing web services. Expert Syst. Appl. 34(2), 1500–1507 (2008)

14. Ding, Z., Wang, J., Jiang, C.: An Approach for Synthesis Petri Nets for Modeling and Verifying Composite Web Service. J. Inf. Sci. Eng. 24(5), 1309–1328 (2008)
15. Dong, Y., Xia, Y., Sun, T., Zhu, Q.: Modeling and performance evaluation of service choreography based on stochastic petri net. JCP 5(4), 516–523 (2010)
16. Du, Y., Li, X., Xiong, P.: A Petri Net Approach to Mediation-aided Composition of Web Services. IEEE Transactions on Automation Science and Engineering (2012) (to appear)
17. El Haddad, J., Manouvrier, M., Rukoz, M.: TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition. IEEE Trans. on Services Computing 3(1), 73–85 (2010)
18. Fang, X., Jiang, C., Fan, X.: Independent global constraints for web service composition based on GA and APN. In: Proc. of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC 2009, pp. 119–126 (2009)
19. Gabrel, V., Manouvrier, M., Megdiche, I., Murat, C.: A new 0-1 linear program for qos and transactional-aware web service composition. In: 4th IEEE Int. Workshop on Performance Evaluation of Communications in Distributed Systems and Web based Service Architectures (PEDISWESA), Cappadocia, Turkey (2012) (to appear)
20. Hamadi, R., Benatallah, B.: A Petri net-based Model for Web Service Composition. In: Proc. of the 14th Australasian Database Conf., ADC 2003, vol. 17, pp. 191–200 (2003)
21. Hendler, J.: Web 3.0 emerging. Computer 42, 111–113 (2009)
22. Hinz, S., Schmidt, K., Stahl, C.: Transforming BPEL to Petri nets. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 220–235. Springer, Heidelberg (2005)
23. Jaeger, M.C., Rojec-Goldmann, G., Mühl, G.: QoS Aggregation for Web Service Composition using Workflow Patterns. In: 8th Int. Enterprise Distributed Object Computing Conf. (EDOC), Monterey, California, USA, pp. 149–159 (2004)
24. Li, B., Xu, Y., Wu, J., Zhu, J.: A petri-net and qos based model for automatic web service composition. Journal of Software 7(1), 149–155 (2012)
25. Li, Q., Liu, A., Liu, H., Lin, B., Huang, L., Gu, N.: Web services provision: solutions, challenges and opportunities (invited paper). In: Proc. of the 3rd Int. Conf. on Ubiquitous Information Management and Communication, ICUIMC 2009, pp. 80–87 (2009)
26. Li, X., Fan, Y., Sheng, Q.Z., Maamar, Z., Zhu, H.: A Petri Net Approach to Analyzing Behavioral Compatibility and Similarity of Web Services. IEEE Trans. on Systems, Man, and Cybernetics, Part A, 510–521 (2011)
27. Liu, A., Li, Q., Huang, L., Xiao, M.: FACTS: A Framework for Fault Tolerant Composition of Transactional Web Services. IEEE Trans. on Services Computing 3(1), 46–59 (2010)
28. Liu, X., Xu, Z.: Independent Global Constraints Web Service Composition Optimization Based on Color Petri Net. In: Proc. of Int. Conf. on Computational Intelligence and Natural Computing, CINC 2009, vol. 02, pp. 217–220 (2009)
29. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing interacting WS-BPEL processes using flexible model generation. Data Knowl. Eng. 64(1), 38–54 (2008)
30. Lohmann, N., Verbeek, E., Dijkman, R.: Petri net transformations for business processes — a survey. In: Jensen, K., van der Aalst, W.M.P. (eds.) ToPNoC II. LNCS, vol. 5460, pp. 46–63. Springer, Heidelberg (2009)
31. Mao, C.: Control Flow Complexity Metrics for Petri Net-based Web Service Composition. Journal of Software 5(11), 1292–1299 (2010)

32. Martens, A.: On compatibility of web services. Petri Net Newsletter 65, 12–20 (2003)
33. Martens, A.: Analyzing Web Service Based Business Processes. In: Cerioli, M. (ed.) FASE 2005. LNCS, vol. 3442, pp. 19–33. Springer, Heidelberg (2005)
34. Massuthe, P., Reisig, W., Schmidt, K.: An Operating Guideline Approach to the SOA. Annals of Mathematics, Computing and Teleinformatics 1, 35–43 (2005)
35. Mei, X., Jiang, A., Li, S., Huang, C., Zheng, X., Fan, Y.: A Compensation Paired Net-based Refinement Method for Web Services Composition. Advances in Information Sciences and Service Sciences 3(4), 169–181 (2011)
36. Mei, X., Jiang, A., Zheng, F., Li, S.: Reliable Transactional Web Service Composition Using Refinement Method. In: Proc. of the 2009 WASE Int. Conf. on Information Engineering, ICIE 2009, vol. 01, pp. 422–426 (2009)
37. Montagut, F., Molva, R., Tecumseh Golega, S.: Automating the Composition of Transactional Web Services. Int. J. Web Service Res. 5(1), 24–41 (2008)
38. Morimoto, S.: A Survey of Formal Verification for Business Process Modeling. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part II. LNCS, vol. 5102, pp. 514–522. Springer, Heidelberg (2008)
39. Ochmańska, E.: Web Services Composition Framework with Petri Net Based Schemas. In: Nguyen, N.T., Katarzyniak, R.P., Janiak, A. (eds.) New Challenges in Computational Collective Intelligence. SCI, vol. 244, pp. 3–14. Springer, Heidelberg (2009)
40. Ouyang, C., Verbeek, E., van der Aalst, W.M.P., Breutel, S., Dumas, M., ter Hofstede, A.H.M.: Formal semantics and analysis of control flow in WS-BPEL. Sci. Comput. Program. 67(2-3), 162–198 (2007)
41. Qian, Z., Lu, S., Xie, L.: Colored Petri Net Based Automatic Service Composition. In: Proc. of the 2nd IEEE Asia-Pacific Service Computing Conf., pp. 431–438 (2007)
42. Rabbi, F., Wang, H., MacCaull, W.: Compensable workflow nets. In: Dong, J.S., Zhu, H. (eds.) ICFEM 2010. LNCS, vol. 6447, pp. 122–137. Springer, Heidelberg (2010)
43. Schafer, M., Dolog, P., Nejdl, W.: An environment for flexible advanced compensations of web service transactions. ACM Transactions on the Web 2 (2008)
44. Shih, D.-H., Chiang, H.-S., Lin, B.: A Generalized Associative Petri Net for Reasoning. IEEE Trans. on Knowl. and Data Eng. 19(9), 1241–1251 (2007)
45. Tan, W., Fan, Y., Zhou, M.: A Petri Net-Based Method for Compatibility Analysis and Composition of Web Services in Business Process Execution Language. IEEE T. Automation Science and Engineering 6(1), 94–106 (2009)
46. Thomas, J.P., Thomas, M., Ghinea, G.: Modeling of web services flow. In: IEEE Int. Conf. on E-Commerce (CEC), Stillwater, OK, USA, pp. 391–398 (2003)
47. Valero, V., Macià, H., Pardo, J.J., Cambronero, M.E., Díaz, G.: Transforming Web Services Choreographies with priorities and time constraints into prioritized-time colored Petri nets. Sci. Comput. Program. 77(3), 290–313 (2012)
48. van der Aalst, W.M.P.: The application of Petri nets to workflow management. The Journal of Circuits, Systems and Computers 8(1), 21–66 (1998)
49. W3C, Web Services Choreography Description Language (WS-CDL) (2004), http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/ (extracted on April 2012)
50. Wang, Y., Fan, Y., Jiang, A.: A paired-net based compensation mechanism for verifying Web composition transactions. In: 4th International Conference on New Trends in Information Science and Service Science (NISS), pp. 1–6 (2010)

51. Xia, Y., Liu, Y., Liu, J., Zhu, Q.: Modeling and performance evaluation of bpel processes: A stochastic-petri-net-based approach. IEEE Trans. on Systems, Man, and Cybernetics, Part A 42(2), 503–510 (2012)
52. Xiong, P., Fan, Y., Zhou, M.: A Petri Net Approach to Analysis and Composition of Web Services. IEEE Transact. on Systems, Man, and Cybernetics, Part A 40(2), 376–387 (2010)
53. Yang, Y., Tan, Q., Xiao, Y.: Verifying web services composition based on hierarchical colored petri nets. In: Proc. of the 1st Int. Workshop on Interoperability of Heterogeneous Information Systems, IHIS 2005, pp. 47–54 (2005)
54. Yang, Y., Tan, Q., Xiao, Y., Liu, F., Yu, J.: Transform BPEL workflow into hierarchical CP-nets to make tool support for verification. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 275–284. Springer, Heidelberg (2006)
55. Yu, H., Fan, G., Chen, L., Liu, D.: Analyzing time constrained service composition based on Petri net. In: 3rd Int. Symposium on Electronic Commerce and Security Workshops, pp. 68–71 (2010)
56. Yu, Q., Liu, X., Bouguettaya, A., Medjahed, B.: Deploying and managing web services: issues, solutions, and directions. The VLDB Journal 17(3), 537–572 (2008)
57. Zhang, Z., Hong, F., Xiao, H.: A colored petri net-based model for web service composition. Journal of Shanghai University (English Edition) 12, 323–329 (2008)
58. Zhao, Z., Wei, J., Lin, L., Ding, X.: A Concurrency Control Mechanism for Composite Service Supporting User-Defined Relaxed Atomicity. In: The 32nd Annual IEEE Int. Computer Software and Applications Conf., pp. 275–278 (2008)

Medical Image Rendering and Description Driven by Semantic Annotations

Alexandra La Cruz[1], Alexander Baranya[1,2], and Maria-Esther Vidal[2]

[1] Biophysics and Bioengineering Applied Group
[2] Semantic Web Group
Simón Bolívar University, Caracas, Venezuela
{abaranya,alacruz,mvidal}@ldc.usb.ve

Abstract. Image-driven medical applications can aid medical experts to visualize tissues and organs, and thus facilitate the task of identifying anomalies and tumors. However, to ensure reliable results, regions of the image that enclose the organs or tissues of interest have to be precisely visualized. Volume rendering is a technique for visualizing volumetric data by computing a 2D projection of the image. Traditionally, volume rendering generates a semi-transparent image, enhancing the description of the area of interest to be visualized. Particularly during the visualization of medical images, identification of areas of interest depends on existing characterizations of the tissues, their corresponding intensities, and the medical image acquisition modality, e.g., Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). However, a precise classification of a tissue requires specialized segmentation processes to distinguish neighboring tissues that share overlapped intensities. Semantic annotations of ontologies such as, RadLex and the Foundational Model of Anatomy (FMA), conceptually allow the annotation of areas that enclose particular tissues. This may impact on the segmentation process or the volume rendering quality. We survey state-of-the-art approaches that support medical image discovery and visualization based on semantic annotations, and show the benefits of semantically encoding medical images for volume rendering. As a proof of concept, we present ANISE (an ANatomIc SEmantic annotator) a framework for the semantic annotation of medical images. Finally, we describe the improvements achieved by ANISE during the rendering of a benchmark of medical images, enhancing segmented part of the organs and tissues that comprise the studied images.

Keywords: Semantic Visualization, Transfer Function, Volume Rendering, Tissue Classification.

1 Introduction

Image segmentation is the process of partitioning the image into portions that correspond to meaningful concepts, while a volume rendering technique generates a semi-transparent image based on a transfer function (TF). A transfer

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 123–149, 2013.

function (TF) maps intensity values of volumetric data or voxels into the optical properties (e.g., opacity and color) used by rendering algorithms to produce a final image. Basically, in computer graphics TFs are used to assign colors and opacity to every rendered pixel as RGB and Alpha values. TFs allow presegment a volumetric data by classifying voxels according to their intensity value; they may be defined based on existing characteristics of a tissue that relate a medical image acquisition modality and an intensity range [21]. Nevertheless, some tissues belonging to different organs may have the same range of intensities. Thus, using a defined TF would not be enough and the tissue classification will normally require a robust and specialized segmentation process to produce a precise tissue classification able to distinguish tissues with overlapped intensities. The definition of an optimal TF for a particular volumetric data is a manual and tedious task for experts.

During recent years, a great number of ontologies and controlled vocabularies have become available under the umbrella of the Semantic Web. Ontologies provide the basis for the definition of concepts and relationships that make possible a global interoperability among the Web of Data. In the Life and Health Sciences domains different ontologies and control vocabularies have been defined, e.g., SNOMED[1], MesH[17], RadLex[2], and Foundational Model of Anatomy (FMA) [23]. These ontologies and controlled vocabularies are commonly applied to encode scientific knowledge through annotations of resources, e.g., MeSH terms have been used by curators to describe PubMed[3] publications and clinical trials published at the Clinical Trials website[4]. The knowledge encoded in these annotations as well as the properties derived from reasoning tasks can be used to retrieve annotated resources or to discover potential new properties. In this paper, we illustrate for a particular type of resources, the medical images, the impact of semantically annotating these resources on the quality of the processes of image segmentation and visualization. As a proof of concept, we present a two-fold strategy able to use and augment these annotations with terms from the RadLex and FMA ontologies and exploit knowledge encoded in these annotations to improve both the image segmentation and visualization processes. Additionally, inferred facts are used to generate RDF documents that describe the organs and tissues as resources that comprise the processed images.

Recently a new research area which integrates image processing, visualization, segmentation and data mining called *Bioimage Informatics*, is emerging [20]. This area relies on labeling or the annotation of medical images to enhance the quality of traditional image-driven processing applications. Some authors proposed the usage of semantic annotations with known ontologies [18], while others are just semantically labeling without using ontologies to infer implicit knowledge [20]. Semantically annotating medical images may help in diagnosis, researching and for handling large volume of data more effectively. Further, we

[1] http://www.nlm.nih.gov/research/umls/Snomed/snomed_mail.html

[2] http://www.rsna.org/radlex/

[3] http://www.ncbi.nlm.nih.gov/pubmed

[4] http://clinicaltrials.gov/

hypothesize that semantic annotations on medical images may also help during visualization, and allow obtaining a more precise description of the resources that comprise a given medical image. To illustrate the impact on the quality of image-driven processing techniques using semantic annotations, we present ANISE (an ANatomIc SEmantIc annotator), a framework for specifying TFs based on semantic annotations and improving the visualization of our resources: the medical images. TFs are defined based on pre-elaborated semantic annotations of volumetric data which are validated against existing medical ontologies. ANISE relies on a customized reasoner to infer the bounding boxes which contain organs or tissues of a given sub-volume area, as well as optimal optical properties, e.g., color and opacity. The knowledge encoded in the ontologies contribute to characterize and locate tissues by applying specific organ selection algorithms. Thus, voxels that are not part of the organ of interest are not considered during the classification process. ANISE is used to illustrate the impact of using ontologies for annotating semantically medical images during the visualization and resource description. Initial results are published at [1].

To summarize the contributions of this paper are the following:

- A survey of the state-of-the-art approaches that support medical image discovery and visualization based on semantic annotations.
- A strategy to exploit semantic annotations of medical ontologies to improve the quality of medical image rendering, and resource description.
- An empirical evaluation of the quality of the proposed strategies on a benchmark of medical images.

This paper is organized as follows: Section 2 gives the preliminary knowledge of this work. Section 3 summarizes the related work. Section 4 presents ANISE and the workflow followed by ANISE to enhance image visualization by exploiting knowledge encoded on the semantic annotations of the image; benefits of the proposed workflow are illustrated in a benchmark of medical images. Finally, we conclude in Section 5 with an outlook to future work.

2 Background

We present basic concepts that define the problems of resource discovery and management, medical volumetric data visualization, and semantic annotation.

2.1 Resource Discovery and Management

A resource may be a data repository, a linked service, a medical image or any application or piece of information that can be accessed from large and complex networks. Resources are characterized by basic properties as name, URLs, as well as by a set of semantic annotations that describe their meaning. Resource discovery is the process of identifying, locating and selecting existing resources that satisfy specific user requirements. Particularly, in the context of Public Health, medical doctors or radiologists need to explore and analyze an interesting and complex resource, medical images, to identify tissues and organs as well as to discover tumors, lesions and anomalies.

Medical images are composed of a set of pixels (voxels) that describes a space of the patient anatomy, and they usually correspond to confidential datasets that can be explored and analyzed by the authorized specialists. Thus, medical images are resources that cannot be necessarily publicly available on the Web, but they need to be precisely selected, explored and analyzed, and robust algorithms able to ensure reliable results are required.

The majority of patients have the same general anatomical description of their tissues or organs. However, conditions of each person as race, age, gender, and size particularize his/her anatomical characteristics. For instance, according to the age, tissues may lose mass, become nodular or rigid, and change their size; therefore, algorithms are highly data depending during segmentation or visualization processes. On the one hand, domain knowledge encoded in formal data models as ontologies in conjunction with rule-based systems can be used to distinguish particular properties of the concepts that appear in an image, i.e., to improve the quality of the segmentation process. On the other hand, ontologies can be used to semantically annotate the areas that enclose the identified organs or tissues with their properties. These annotations provide the basis for resource discovery and description, and further, to precisely visualize the organs or tissues of interest, i.e., semantics encoded in the annotations can be exploited to improve the quality of the rendering process, and the resources that are part of a medical image. In this paper we illustrate the impact of semantic annotations on the quality of image-driven processes, and we describe the functionality and results obtained by the semantic annotator framework named ANISE.

2.2 DICOM Images

A picture archiving and communication system (PACS) consists of medical images and data acquisition, storage, and display subsystems integrated by digital networks and application software [14]. In simple words, a PACS corresponds to an image management system that allows the inter-operation of several display workstations with an image database and storage device. The digital images handle by PACS are defined following the standard DICOM (Digital Imaging and Communication in Medicine)[5]. DICOM is a standard to describe digital medical images of different modalities (e.g., MRI, CT, US, IVUS, xR) and manufacturers (e.g., PHILLIPS, Siemens, General Electric). The DICOM standard consists of a header and a body with the image's data. An image may be formed by several frames (e.g., volumetric data or video). Nowadays, image medical workstations handle DICOM images, and headers from DICOM images contain metadata which can be used for semantically annotate studies of a given patient. Listing 1.1 illustrates a fragment of the DICOM header of the head visible human volumetric data[6]; attributes in lines 9, 13,

[5] http://medical.nema.org/

[6] http://www.nlm.nih.gov/research/visible/applications.html

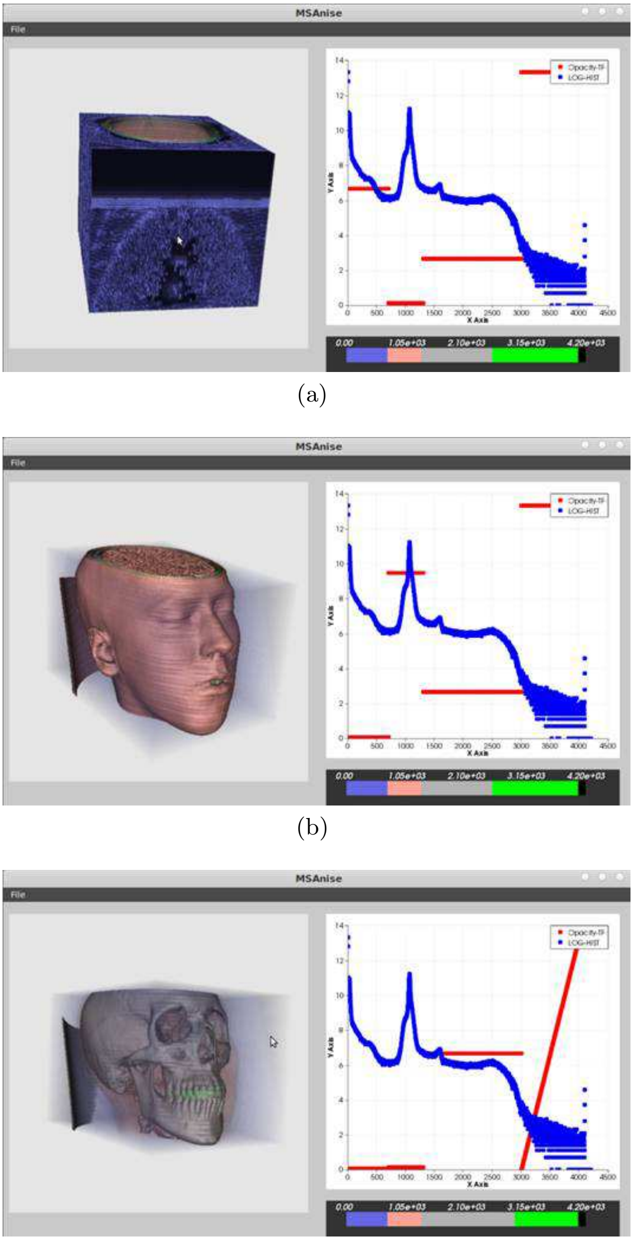
23 and 25 may be of particular interests during the annotation, description and discovery process. Note that both ontologies and DICOM metadata can be combined to complement the information contained in a DICOM file. DICOM metadata includes image modality, voxel/pixel sizes, if the data is a video (US), a volumetric data (MRI, CT) or a 2D image (Radiography). DICOM metadata referring to anatomy, clinical findings, procedures, pharmaceutical/biologic products and clinical terms correspond to SNOMED coded terms, e.g., line 13. The combination of DICOM metadata with knowledge encoded in ontologies can be exploited in rule-based systems to enrich medical image segmentation and resource description algorithms and improve both their quality and performance.

Listing 1.1. A fragment of a DICOM header of the Head Visible Human volumetric data

```
 1  # Dicom-File-Format
 2  # Dicom-Meta-Information-Header
 3  # Used TransferSyntax: LittleEndianExplicit
 4  (0002,0000) MetaElementGroupLength 194
 5  (0002,0001) FileMetaInformationVersion 00\01
 6  (0002,0002) MediaStorageSOPClassUID =CTImageStorage
 7  # Dicom-Data-Set
 8  # Used TransferSyntax: LittleEndianExplicit
 9   (0008,0008) ImageType [ORIGINAL\PRIMARY\AXIAL]
10  (0008,0020) StudyDate [20050101]
11  (0008,0030) StudyTime [010100.000000]
12  (0008,0050) AccessionNumber [1]
13  (0008,0060) Modality [CT]
14  (0008,0070) Manufacturer [GDCM]
15  (0008,0080) InstitutionName [National Library of Medicine]
16  (0008,0081) InstitutionAddress [http://www-creatis.insa-lyon.fr/Public/Gdcm]
17  (0008,1030) StudyDescription [Visible Human Male]
18  (0008,103e) SeriesDescription [Resampled to 1mm voxels]
19  (0010,0010) PatientsName [Adam]
20  (0010,0020) PatientID [000-000-002]
21  (0010,0030) PatientsBirthDate [20050101]
22  (0010,0032) PatientsBirthTime [010100.000000]
23  (0010,0040) PatientsSex [M]
24  (0018,0050) SliceThickness [1.0]
25  (0018,5100) PatientPosition [HFS]
```

2.3 Transfer Functions

A transfer function (TF) maps intensity values of volumetric data into the optical properties (e.g., opacity and color) used by rendering algorithms to visualize areas of interest and put in background the rest (see Fig. 1). Particularly, with medical image and depending on the modality, every tissue is represented by an intensity value. TFs allow for pre-segment (classify) and image, and rely on existing characterizations of the organs that relate a tissue, an intensity range and medical image modality [21], e.g., Computed Tomography (CT), Magnetic Resonance (MR) or Ultra Sound (US) [21]. 2D images generated using a configuration of a TF are tailored for a particular volume rendering technique used to render the image. Radiologists usually need to manually tune a TF to identify the thresholds of intensity values that best represent tissues or organs of interest. Nevertheless, tissues of different organs may have overlapped intensities; thus, considering only intensity values is not enough to produce a precise tissue classification, and specialized segmentation processes are required.



(a)

(b)

(c)

Fig. 1. Examples of applying different configurations of Transfer Function to the same volume. The volume corresponds to a CT image of a head. Left-hand side represents the rendered images while the right-hand side presents the corresponding Transfer Functions used during rendering.

A simpler way to define a TF is using the histogram of intensity value distribution of the volume (right plots in Fig.1 (a), (b) and (c)). Different colors and levels of opacity are assigned to different areas in the histogram. Thus, each particular manual configuration of a TF generates a different visualization (left images in Fig.1 (a), (b) and (c)).

2.4 Medical Image Segmentation

In medical imaging, segmentation is the process of classifying and separating different regions. It is a prerequisite for quantification of morphological disease manifestation, for volume visualization and modeling, for surgical planning and simulation (e.g., using virtual endoscopy) [31]. A general segmentation technique classifies the range of intensity values. This technique relies on the assumption that every tissue from each anatomical organ has a different intensity value [21]. This is due to the absorption and/or emission property of the object being imaged by a medical image modality, which is different for blood, muscle, bone, air, fat, and so on. Based on this fact, it is possible to classify different objects according to the thresholds of intensity values defined for every tissue. Nevertheless, due to factors such as noise, partial volume effect, and artifacts, this approach is not enough for an accurate segmentation. To overcome this limitation, the technique has been extended with the capabilities to adapt the local threshold or to analyze the neighborhood of a pixel (voxel). Furthermore, another technique named region growing, extends the threshold technique and relies on the classification of pixels (voxels) to fulfill certain constrains given as input. From an initial pixel (voxel), the neighborhood is analyzed and added to the region if it satisfies a decision rule. Normally the decision rule is defined using threshold values, the gradient operator, and/or spatial proximity. This method assumes that discontinuities are not possible between objects. The growing criteria should be sufficient to face local image variations; however, it can produce holes and over-segmentation.

Several segmentation techniques have been applied to process medical images [4,15,19] (e.g., Model Based Segmentation [12], Level Set [32], Multi-scale [26] between others). However, we focus on simple segmentation techniques e.g., threshold and histogram based methods, and show the benefits of combining them with annotations of medical images from medical ontologies. Particularly, we propose heuristics that exploit knowledge about anatomic concepts and their relationships from medical ontologies, and illustrate how both visualization and description of the resources that comprise a medical image can be improved.

2.5 Ontologies

Controlled vocabularies have been used to avoid ambiguous interpretations of referential terms during resource description and analysis. Most of vocabularies are limited to particular and very specific subject domains. The Health Science domain has widely adopted such practices from long time to create universal descriptions of this knowledge domain. Several ontologies have been defined to describe relations among controlled terms of different domains, providing them the basis for the Web of Data. Annotating volumetric images using controlled terms allow information search and retrieval, classification or comparisons among different visual representations of medical data. Different (semi-)automatic techniques have been defined to exploit knowledge encoded on ontologies during the retrieval, gathering and management of information, or just for inferring new knowledge. Biomedical ontologies such as, SNOMED, MeSH, NCI Thesaurus [28], RadLex and the *Foundational Model of Anatomy*(FMA) have been used for authoring and annotating resources, e.g., scientific publications and images. These ontologies are described as follows:

- ***SNOMED*:** is an ontology of medical terms that represents synonyms and definitions of concepts as diseases, findings, procedures, microorganisms, and drugs.
- ***MeSH*:** is a taxonomy of a controlled vocabulary of medical terms defined by the United States National Library of Medicine (NLM), to describe and indexing publications in the Life Sciences domain. MeSH encloses concepts of 16 categories of medical concepts which include diseases, drugs, anatomy, and organisms.
- ***NCI Thesaurus*:** is also an ontology of medical terms defined by National Cancer Institute as reference terminology for coding diseases, drugs, and procedures. It provides definitions, synonyms, and different relationship concepts defined in the ontology.
- ***Foundational Model of Anatomy*:** is an ontology comprised of classes and relationships that describe the human anatomy. FMA provides the basis for defining membership and spatial relationships among voxels in a volumetric dataset which will correspond to the facts in a rule-based system that will be used to infer new facts. Furthermore, there are terms in this ontology that are used for annotating non-anatomical elements, e.g., bounding boxes around particular anatomical organs or some particular points of interest.
- ***RadLex*:** RadLex is an ontology defined for radiologist; it is composed of terms required to annotate medical images. ANISE relies on RadLex terms to describe characteristics of the image itself such as modality, and other acquisition related characteristics that may alter the interpretation and visualization of an image, e.g., orientation.

We show as a proof of concept the benefits of combining of medical ontologies, semantic annotations, resource description, and visualization techniques during rendering. ANISE precisely distinguishes the organs and tissues that need to be visualized and described, even though neighboring tissues share the same intensities, when semantic annotations are used in conjunction with knowledge encoded in the FMA and RadLex ontologies.

2.6 Probabilistic Soft Logic

A Probabilistic Soft Logic [3] program is a set of weighted rules of the form *Body(X)→Head(Y)* where, each rule is associated with a score greater than zero that represents the relative importance of the rule in the program. Additionally, *Body(X)* corresponds to a conjunction of predicates whose truth values are in the interval [0,1]. Soft predicates correspond to similarity functions between explicit or implicit facts, while *Head(Y)* is a single predicate. Consider the following rule that illustrates the PSL semantics:

$$modality(M) \wedge baseVoxel(X, Y, Z, I) \wedge tissueMapModality(T, M, D, I) \rightarrow tissue(T, X, Y, Z, I). \quad (1)$$

Rule 1 states that a tissue T has an intensity I and appears in the voxel with coordinates *X,Y,Z*, depending on how similar is the intensity I to the intensity D of the tissue T when the image is captured using a modality *M.* The truth degree of the *tissue(T,X,Y,Z,I)* corresponds to the Lukasiewicz t-norm of the truth values of the predicates *modality(M)*, *baseVoxel(X,Y,Z,I)* and *tissueMapModality(T,M,D,I)*, considering that the conjunction of two predicates is computed as the maximal value between 0 and the sum of the truth values of *modality(M)*, *baseVoxel(X,Y,Z,I)* and *tissueMap(T,D,I)* minus 1. In general the interpretation In of the logical operators conjunction $\wedge$, disjunction $\vee$, negation $\neg$, and $\rightarrow$ is computed as follows:

$$In(S_i \wedge S_j) = max\{0, In(S_i) + In(S_j) - 1\} \quad (2)$$

$$In(S_i \vee S_j) = min\{In(S_i) + In(S_j), 1\} \quad (3)$$

$$In(\neg S_i) = 1 - In(S_i) \quad (4)$$

$$In(S_i \rightarrow S_j) = In(\neg S_i \vee S_j) \quad (5)$$

Once all the variables in a rule are instantiated, i.e., the rule is grounded, the interpretation of the rule is computed according to rule 5. A rule is satisfied if and only if $In(S_i) \leq In(S_j)$, i.e., the truth value of the head $(In(S_j))$ is at least the same truth value than the body$(In(S_i))$. Note that the interpretation of a PSL rule does not coincide with the traditional interpretation of Horn clauses implemented by the refutation inference rule performed in Programming Logic languages as Prolog or Datalog [6]. Given the truth values of a rule r under an interpretation In, the distance for satisfaction of $In(r)$, $d(In(r))$, is defined as how far is the truth values of $In(r)$ to 1, i.e.,

$$d_r(In) = max\{0, In(S_i) - In(S_j)\} \quad (6)$$

Finally, an interpretation In is a model of a PSL program P, if In is the interpretation that satisfies with the highest probability, the majority of the ground rules of P. Given the interpretation In, the probability of satisfaction of a rule r weighted with the score $s(r)$ is computed as $s(r) \times In(r)$.

3 Related Works

According to Peng [20], a new research area named *Bioimage Informatics*, is emerging. This area relies on approaches developed to: *i*) manage and retrieval bioinformatics and biomedical images, *ii*) (semi-)automatically annotate 2D and 3D images, and *iii*) exploit the semantic of annotations and tags during segmentation, visualization and registration. In this section we summarize the contributions of existing approaches and discuss limitations of these approaches, mainly whenever semantics encoded in the tags or annotations of the images is required to enhance the processes of management, retrieval and visualization of medical images.

3.1 Image Retrieval From Databases of Tagged Images

Semantic annotations have been well-studied in areas such as multimedia analysis of video [2], images classification [10,34], retrieval [16,18,30], retrieval and discovery of images from a database [10,34], as well as in medical image-driven information systems [10]. If resources are images, queries can be image- or text-based. Querying image-based databases requires describing an image by its optical characteristics (e.g., Content-based image retrieval (CBIR) systems [16], sketch or features [33]) while text-based queries exploit image annotations or tags to retrieve those that contain tags according to the query. Normally querying image-based systems use artificial intelligence and pattern recognition techniques to classify the images that satisfy a given query.

3.2 Pattern Recognition Based Approaches for Automatic Annotation of Images

Automatic annotation of histopathological images normally require recognition of morphological and architectural features that describe pathological lesions. Cruz et al. [10] propose a method for automatically annotating histopathological images in two stages: training and prediction. A training set consists of mono-label annotated histopathological images, while automatic annotated images are generated using a probabilistic support for prediction and spatial location of morphological and architectural features in healthy and pathological tissues. They used a Bag of Features Representation (BOF) which is a model that allows representing an object by integrating its parts. A BOF comprises a feature detection and description, codebook of visual vocabulary or optical properties and BOF image representation. Cruz et al. [10] define its own ontology to manage the medical terms or pathological terms used for classifying and annotating the images. Basically, the proposed BOF corresponds to an ontology used as a general model for representing the features and relations between different parts of a particular object.

3.3 Machine Learning Based Approaches for the (Semi-)Automatic Annotation of 2D Images

Some approaches [5,7,34] have focused on techniques to (semi-)automatically annotate image regions which have been previously located by using a rule-based system. Basically, these approaches rely on annotations to apply image segmentation. Yu et al. [34] use the Hidden Markov Model to automatically annotate images; annotations are derived from sequences of visual features and keywords associated with the most appropriate concepts in equivalence classes. These classes define visual features such as, the sky is blue, and the mountains are green or brown, and so on. In these cases, images from the Corel system were considered. Even when these techniques are used for medical images; this is a demonstration of the feasibility of using semantic annotations on image databases in general.

Chan et al. [7] define a Supervised Multi-class Labeling (SML) model for semantic annotation, which is compared with existing modeling and is able to identify an optimal semantic labeling and retrieval. Carneiro et al. [5] propose a probabilistic formulation for semantic image annotation and retrieval. This formulation provides a solution to the problem of classifying images into different semantic classes and labeling them according to this classification. This supervised learning process involves the definition of a training set of labeled images and their corresponding semantic classification. This training set is then used for a machine learning based algorithm to automatically classify, annotate and further retrieval images from an image database. These approaches provide a solution to the problem of semantically annotating images, but they have been mainly applied to 2D images from the Web and Social Networks. Although some 2D algorithms can be easily extended to 3D [21], in most of the cases, there is not a natural extension and new strategies must be applied, because working with 3D data requires to manage at least with two more parameters e.g., deep and rotation. This may increase time and space complexity of both the semantic annotation algorithms and the strategies required for retrieval these volumetric data based on the semantic annotations.

3.4 Semantically Annotation of 3D Medical Images

Möller et al. [18] present a technique for annotating and searching medical images using ontological semantic concepts for retrieving images from a Picture Archiving and Communication System (PACS); ontologies as FMA and RadLex are used to annotate and retrieve the images. Rubin et al. [24,25] have created a Semantic Annotation system for radiological images. This system was designed to access and analyze the semantic content of images in a digital medical image database, and add semantic content to the images. These annotations are labels that indicate anomalies or pathologies presented in organs or tissues visualized from an image. Rubin et al. [25] try to make available on the Internet, medical information about annotated images; they also created an ontology for image annotation able to represent the semantics of medical images by pixel-content. They integrate different formats, e.g., medical records systems and image archives from hospitals. The main idea behind this approach is to offer physicians and researchers a platform for better understanding of biological and physiological significance of a medical image content. The main limitation of this approach is the lack of semantics, e.g., they do not mention the use of DICOM images even though they provide the basic metadata for supporting semantic annotations.

3.5 Enhancing Visualization of Volumetric Data by Using Semantic Annotations

Recently, the problem of tissue classification by using semantically annotating volumetric data has gained attention in the literature [8,9,13,22]. Rautek et al. [22] present a fuzzy rule-based system that maps volumetric attributes to visual styles; rules are defined by users without special knowledge about the rendering technique. Gerl et al. [13] overcome this limitation and propose a rule-based system for semantic shader augmentation; rules enhance static visualization mappings in a shader program. Although both systems rely on rule-based systems to characterize TFs, they do not exploit knowledge encoded in ontologies during visualization or tissue classification to improve the quality of the visualization process. Although applications of semantic annotations have been illustrated, nothing is said about the benefits of using these annotations and the encoded semantics during the definition of TFs.

3.6 Summary of Semantic Annotations of Images Approaches

Table 1 summarize and give a view of different semantic annotations of image-based approaches existing in the literature. In general all of them are used for annotating 2D images, some others are used for annotating 3D images. Bloehdorn et al. [2] use videos, where every frame (2D image) is annotated. Recently we found interest in applying semantic annotation techniques for improving visualization in medical images, using ontologies [8,9] and without using ontologies [13,22].

4 Exploiting Semantic Annotations to Enhance Image Visualization and Resource Description

As a proof of concepts we present ANISE, a tool able to exploit annotations of medical images to improve the quality of the image visualization. ANISE relies on knowledge encoded in medical ontologies FMA and RadLex, reasoning tasks and a PSL rule-based system to partition the original volumetric data into a bounding box that only encloses a given tissue or organ, and annotates this bounding box with the corresponding terms in the FMA and RadLex ontologies. The strategy implemented by ANISE comprises three main phases: *i*) semantic segmentation, *ii*) resource description, and *iii*) semantic volume rendering. First, the semantic segmentation phase partitions the volumetric data into regions

Table 1. Summary table of Semantic Annotation of Image Approaches

Approach	**2D**	**3D**	**Medical Images**	**Ontology**	**Visualization**
Bloehdorn et al. [2]	per Frames	NOT	NOT	YES	NOT
Yoon et al. [33]	YES	NOT	NOT	NOT	NOT
Ko et al. [16]	YES	NOT	NOT	NOT	NOT
Cruz et al. [10]	YES	NOT	YES	YES	NOT
Yu et al. [34]	YES	NOT	NOT	NOT	NOT
Wei et al. [30]	YES	YES	YES	YES	NOT
Möller et al. [18]	YES	YES	YES	YES	NOT
Chan et al. [7]	YES	NOT	NOT	NOT	NOT
Carneiro et al. [5]	YES	NOT	NOT	NOT	NOT
Criminisi et al. [8,9]	YES	YES	YES	YES	YES
Rautek et al. [22] and Gerl et al. [13]	YES	YES	YES	NOT	YES

that enclose tissues or organs; these regions are annotated with ontology terms that describe these organs or tissues. This phase relies on a rule-based system that implements inference tasks to derive facts that will be used to annotate the image. Annotations regarding to visualization methods and anatomic parts are inferred using ontology relations (e.g., `SubClass`) for specific classes (e.g., the Anatomical Set); we will illustrate this process in Section 4 with a use case. Simultaneously, inferred facts are translated to RDF, and a document that describes the resources that comprise the image is generated and stored in the ANISE catalog; this document is named an *.ANI* file. Further, during the semantic volume rendering phase different TFs defined from annotations are used. TFs have been enhanced with the capability to exploit knowledge encoded in the annotations of the volumetric data, in order to use the inferred optical properties in the rendered image.

Semantic Segmentation: annotates an image with information about: *i*) resource authoring, type and identification; *ii*) acquisition modality; *iii*) acquisition characteristics like patient orientation in the image; *iv*) structural and anatomic elements presented and identified in the image; *v*) regions and points of particular interest; and vi) rendering information (definition of a TF according to the particular tissue).

Resource Description: information encoded in DICOM metadata in conjunction with facts inferred by the rule-based systems are used to generate an RDF document that comprises the description of the tissues and organs that are part of the image. Predicates in FMA and RadLex are used to describe these tissues and organs as RDF resources.

4.1 Architecture

Achieving high quality rendered images require interpreting each intensity value according to a given tissue. In consequence, a correct representation of information through semantic annotations should ensure: *i*) minimal error tissue classification due to reasoning and inference, and *ii*) an accurate visual representation.

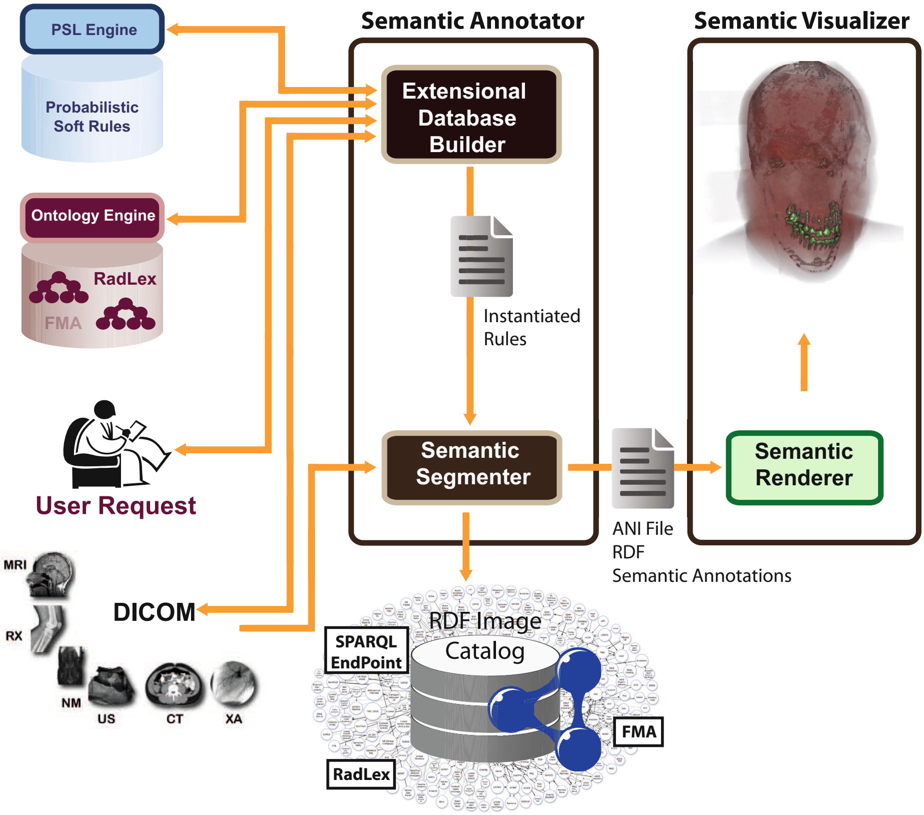


Fig. 2. The ANISE Architecture

Figure 2 shows the main components of ANISE: (1) a Semantic Annotator, and (2) a Semantic Visualizer. The Annotator extends an image original annotation with terms that encode the properties of the classified tissues. Thereafter, the Semantic Visualizer executes visualization algorithms based on the annotated volumetric data to render it. ANISE receives as input a volumetric DICOM file, a user request, and a set of medical ontologies. The DICOM file corresponds to the medical image and is comprised of metadata fields that describe main characteristics of the image, as well as a bounding box that encloses the area of interest and a seed point. The user request indicates the tissue or organ to be visualized and the optical properties to be used during the visualization. We assume the bounding box and the seed point can be either manually identified by a radiologist or by a machine learning based approach as the one proposed by Criminisi et al. [9].

1. **Semantic Annotator:** annotates an image with information about: i) resource authoring, type and identification; ii) acquisition modality; iii) acquisition characteristics like patient orientation in the image; iv) structural and anatomic elements presented and identified in the image; v) regions and points of particular interest; and vi) rendering information. The Semantic Annotator is comprised of two sub-components: the **Extensional Database Builder** and the **Semantic Segmenter**. This component assumes the anatomical area that encloses the tissue/organ of interest is rounded by a bounding box; also, a seed point is highlighted. All this information is encoded in the DICOM header of the image volumetric data file. A bounding box method similar to the one proposed by Criminisi et al. [9] is used to model this anatomic information.

   The **Extensional Database Builder** analyzes information described by DICOM header fields to ground predicates that will comprise the extensional database of the **probabilistic Soft Rule** system. An **Ontology Engine** is used to perform inference tasks such as, classification of the `SubClass` relationship between anatomical concepts. The FMA and RadLex ontologies and facts encoded in relevant DICOM header fields correspond to the input to these inference processes. Derived facts describe the semantic properties of the organ represented in the DICOM header fields, e.g., the tissues or organs that comprise its neighborhood or anatomic region (Head, Torso, Legs, From abdominal until foot, etc.), or the tissues that compose them (Head, Dentition, Chest, Mouth, Eyes, etc.). The knowledge encoded in the ontologies is exploited to identify the relevant properties of the organ of interest; all the inferred facts are also represented as ground predicates in the extensional database. Ontology classification reasoning tasks are performed with Jena[7].

   Once the extensional database is built, the **Semantic Segmenter** contacts the Probabilistic Soft Logic (PSL) engine to segment the tissue or organ of interest. To perform this task, the **Semantic Segmenter** first analyses the image acquisition characteristics and correlates body structures of particular interest in order to normalize information for further processing. The bounding box and the seed point are used in conjunction with tissue pre-classification facts to feed the inference process expressed in Probabilistic Soft Logic (PSL) [3]. This process determines the likelihood for a given tissue to be included in a particular region. Closely located tissues with similar intensity values are usually treated as the same values; thus, spatial and anatomic information are used to discriminate by annotating specific

[7] `http://jena.apache.org/`

points. Segmentation based on voxels neighborhood represent these tissues considering the associated semantic annotations. A use case illustrates the process for a particular tissue are presented on following subsection.

Extensional ground facts as well as the ones inferred by the PSL engine are used to describe the tissues or organs that are part of the image, as RDF resources. These documents are stored in a RDF catalog that can be accessed by a SPARQL endpoint. Additionally, all these facts are expressed in a set of *.ANI* files describing the soft probabilities for each point of the volumetric data to represent annotated tissues.

2. **Semantic Visualizer:** derived annotations and *.ANI* files describing point tissue similarity probabilities are used for specifying TFs based on semantic annotations. Generated TFs are used by rendering algorithms to visualize the classified tissues. Partial piece-wise TFs are used to select appropriate color and opacity values according to tissues likeness, and for rendering them. A default TF is applied on non-annotated voxels and regions.

4.2 Applying an ANISE Workflow- A Use Case

We illustrate the ANISE workflow in multiple dataset (Table 2), to visualize the FMA term *dentition* and *jaw* from a CT-Head volume data.

Table 2. Datasets used for illustrating the utility of using semantic annotations on Medical Images. This dataset is available in [11], [27], and [29], respectively.

Volume Data	Dimensions (voxels)	Voxel size (mm)	File size (MB)
ct_head.dat	256x256x113	1x1x2	14.8
skewed_head.dat	184x256x170	1x1x1	16.0
visible_head.dat	512x512x245	1x1x1	128.0

Fig. 3(a), (b) and (c) illustrate the rendering of the images applying a simple TF which maps intensity values to visualize the tissues that have the same intensity that dentition; these tissues are colored in *green*. Although data were properly pre-classified, it is not possible to discriminate only dentition by just considering the corresponding intensities, i.e., some other tissues were painted, and it was not possible further tuning the TF. In this case the intensity value range used for identifying the dentition overlaps with the intensity value range of other tissues like cranial bone, for example. Nevertheless, if semantic annotations are used in conjunction with knowledge encoded in the FMA and RadLex ontologies, ANISE can determine that only the teeth should be colored different than the rest (*green* in our example); this is done by selecting an appropriate set of points, applying **Normalization** rules, and considering the **Image Modality** taxonomy. Thus, a better classification of different tissues can be done in an automatic way.

- **Image Modality:** supports a generic tissue classification process which is independent of the image modality. The RadLex term used for *Tomography*

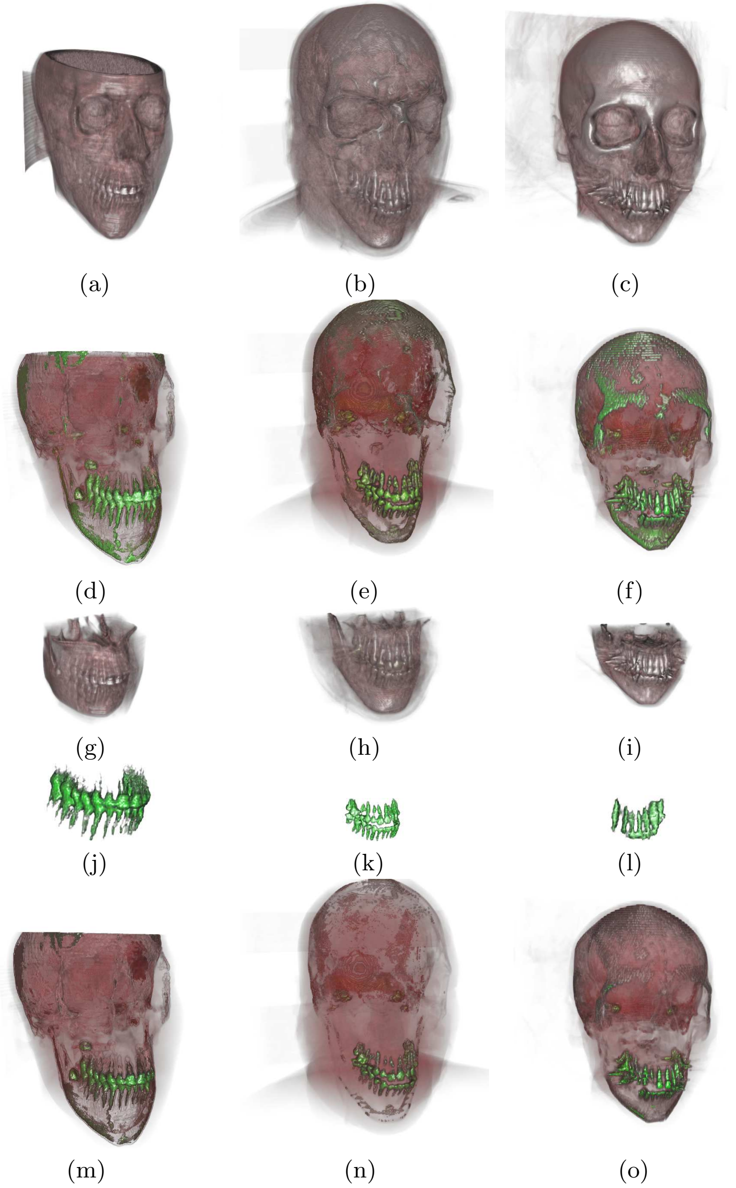


Fig. 3. Results of running the proposed approach on 3 different images based on volumes: (a) skewed_head (b) visible_human and (c) ct_head. Results of rendering tissue mapping rules applied to (a) (b) (c) volumes are shown on (d) (e) (f), respectively. Rendering annotated bounding box rules on same original volume are shown on (g) (h) (i). The tissue region growing rule based on similarity by annotating seed point are shown on (j) (k) (l). The results of applying all segmentation rules shown on (m) (n) (o).

is `RID28840`[8] and the term `RID10311` (imaging modality) can be reached by using the `SubClass` relationship. Further, whenever the image is an MRI the term `RID10312` from the same taxonomy is used to annotate the image, i.e., terms `RID28840` and `RID10312` share an ancestor `RID10311`. Tissues' intensity ranges are represented as facts and used during the inference process in conjunction with these annotations to pre-classify the image voxels.

- **Volume format:** ANISE current version receives images in raw format, i.e., data correspond to a sequence of intensity values. This information is extracted from the attribute `format` from DCMI[9] metadata.
- **Normalization** rules: are used to transform volumes into a uniform scale considering orientation, voxel size, and modality. Default values are assumed if they are not given. In our use case, we used the term *voxel geometry* `RID29031` from RadLex and its ancestors in the `SubClass` branch, i.e., *non-isotropic voxels*, *near-isotropic voxels*, *isotropic voxels*.
- **Dimension:** we used the term *location* (`RID39038`) from RadLex to represent the header size, and dimensions in x, y and z of the volume.
- **Tissue:** *dentition* from FMA is the most relevant term in this use case.

We chose *dentition* because it is characterized as the tissue with the highest intensity value, and the challenge consists of separating the dentition tissue from tissues around it. PSL rules are used to compute the degrees of membership of a voxel to the tissue of interest (*dentition*); it is mainly based on the intensity value range. The rules that comprise the rule-based system are as follows; they specify TFs that better visualize the tissue of interest:

$$\begin{aligned} &TransferFunctionRule: \\ &targetTissue(T) \wedge tissue(T,X,Y,Z,I) \wedge insideRegion(R,T,X,Y,Z) \wedge \\ &sameOrgan(S,T,X,Y,Z,I) \rightarrow opacity(T,X,Y,Z). \end{aligned} \tag{7}$$

where, the truth values of $opacity(T,X,Y,Z)$ representing the opacity value at point (X,Y,Z) when rendering tissue T, are determined by the sum of truth values of the following predicates:

- $tissue(T,X,Y,Z,I)$ describes truth values of the voxel X,Y,Z with intensity I that belong to the tissue T, mapping to objective or target tissue by instantiating T in ground predicate $targetTissue$. This value is defined by:

$$\begin{aligned} &TissueMappingRule: \\ &modality(M) \wedge baseVoxel(X,Y,Z,I) \wedge \\ &tissueMapModality(T,M,D,I) \rightarrow tissue(T,X,Y,Z,I). \end{aligned} \tag{8}$$

where, *baseVoxel(X,Y,Z,I)* is a fact representing the voxel on the image; *tissueMapModality(T,M,D,I)* is a PSL predicate that represents for a

[8] `http://purl.bioontology.org/ontology/RID/RID28840`

[9] `http://dublincore.org/documents/dcmi-terms/`

particular tissue T (e.g., *dentition*) and a particular acquisition method M (e.g. Computed Tomography) the probability of the voxel *X,Y,Z* belongs to the intensity value range D considering its intensity value I. Initially a intensity value range is specified, and as far as the inference over the annotations are generated, a new intensity value range is produced and then, a more precise TF is defined.

- *insideRegion(R,T,X,Y,Z)* describes truth values of the voxel X, Y, Z belonging to a region R being representing tissue T. Applying the inference process, a bounding box that best fits the area of the tissue of interest is derived by instantiating variables R, T, AX, AY, AZ from an initial location specified in ground predicate *anatomicRegion*.

$$\begin{aligned}&BoundingBoxRule:\\ baseVoxel(X,Y,Z,I) \wedge anatomicRegion(R,T,AX,AY,AZ)&\\ \wedge inside(X,AX) \wedge inside(Y,AY) \wedge inside(X,AY)&\\ \rightarrow insideRegion(R,T,X,Y,Z).&\end{aligned} \tag{9}$$

- *sameOrgan(S,X,Y,Z,I)* describes truth values of the voxel X, Y, Z belonging to the same organ that seed point S having an intensity of I. This value is defined by the rule:

$$\begin{aligned}&RegionGrowingRule:\\ baseVoxel(X,Y,Z,I) \wedge close(X,Y,Z,I,X1,Y1,Z1)\wedge&\\ seed(S,T,X1,Y1,Z1) \rightarrow sameOrgan(S,T,X,Y,Z,I).&\end{aligned} \tag{10}$$

  Given a seed point (*seed(S,T,X,Y,Z)*), known to be part of the tissue of interest T and analyzing its neighborhood, the area around this seed point is augmented. A point will be part of the tissue if its intensity value belongs to the intensity value range of the tissue T, and close to the tissue area.

Finally, applying the same approach and datasets but to identify a different tissue and visually representing it with different optical properties. We have used the same set of PSL rules, but the extensional database was populated with facts that represent the properties of the *jaw*. Fig. 4 shows the results by performing rule execution to the segment and visualize*jaw* with blue color.

To summarize, as shown in both use cases, the following facts are required for each dataset and user request to accurately visualize a medical image:

- **Base Voxels:** intensity value map for each voxel in the dataset.
- **Intensity value Map:** an intensity value map should be specified initially according to acquisition mechanism; however, it can be adapted according to results inferred from the rules.
- **Seed point:** this is a fixed value, received from the user describing a voxel known to be part of the tissue of interest.
- **Anatomic Region:** the rule-based system identifies the enclosed anatomical entity from an input bounding box; this entity is the one that better fits the tissue of interest.

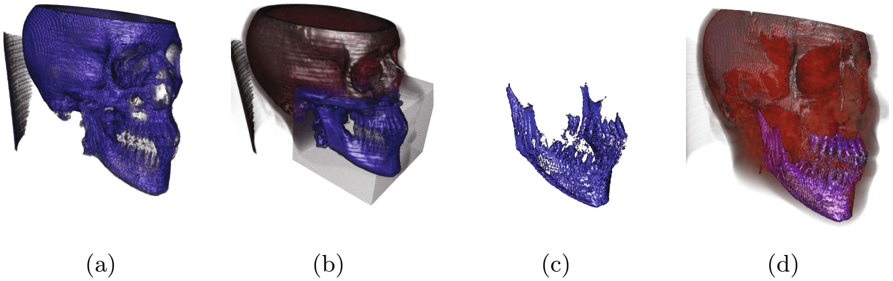


Fig. 4. Results of running the proposed approach on the visible human dataset to select the anatomic region of the jaw and render it in blue. (a) Application of tissue mapping rules. (b) Results of rendering with the application of the bounding box. Image (c) results of rendering with the application of region growing rule, showing tissues sharing the same intensity range than a seed point. And Image (d) showing the final results from rendering with all the rules.

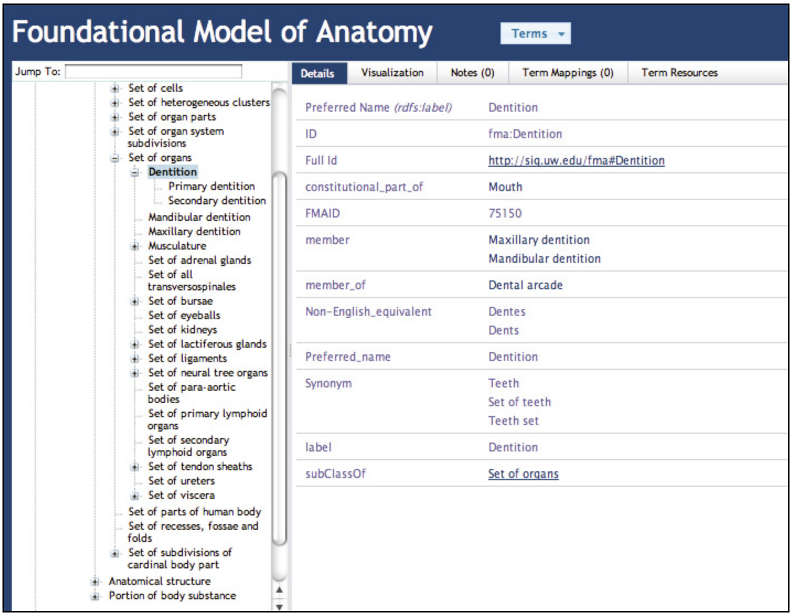
- **Modality:** indicates the method used to acquire the volumetric data; it expresses the mode intensity values should be processed.

4.3 Discussion

In this section we summarize our approach and discuss its advantages and limitations.

We have proposed a methodology that can scale up to any type of medical images, anatomical regions and tissues as far as the following remarks hold: i) Input Medical images comply with the DICOM standard, ii) areas of interest are enclosed in a bounding box and annotated with a general term (e.g., FMA: face); iii) a user request specifies target tissues and the optical properties to be used in the visualization; and, iv) ontology concepts are classified using an ontology engine. As proof of concept, we evaluate our approach on the use cases shown in the paper. These cases require to identify high intensity tissues (e.g., bones, teeth), and Computed Tomography scans are better suitable, because they provide finer granularity of bone physical details. ANISE just considers the most likely localization of a given tissue. First, an initial and basic TF is defined using a normalized model. Then, this model is used for further inferences; rules are applied independently to the acquisition method by selecting when an intensity value for a given point in the space falls inside an appropriate interval. As previously stated, simple intensity classification is not enough to properly determinate matching between voxels of a same tissue or anatomical organ. Additional inference processes need to be conducted; they depend on the annotations. In this example, the region of interest that describes the tissue to be analyzed is presented.

A first approach consists of selecting the most likely location of a region of interest, i.e., a bounding box covering the organ of interest. We assume ground


Foundational Model of Anatomy
Terms
Jump To:
Set of cells
Set of heterogeneous clusters
Set of organ parts
Set of organ system subdivisions
Set of organs
Dentition
Primary dentition
Secondary dentition
Mandibular dentition
Maxillary dentition
Musculature
Set of adrenal glands
Set of all transversospinales
Set of bursae
Set of eyeballs
Set of kidneys
Set of lactiferous glands
Set of ligaments
Set of neural tree organs
Set of para-aortic bodies
Set of primary lymphoid organs
Set of secondary lymphoid organs
Set of tendon sheaths
Set of ureters
Set of viscera
Set of parts of human body
Set of recesses, fossae and folds
Set of subdivisions of cardinal body part
Anatomical structure
Portion of body substance
Details
Visualization
Notes (0)
Term Mappings (0)
Term Resources

Preferred Name *(rdfs:label)*	Dentition
ID	fma:Dentition
Full Id	http://sig.uw.edu/fma#Dentition
constitutional_part_of	Mouth
FMAID	75150
member	Maxillary dentition Mandibular dentition
member_of	Dental arcade
Non-English_equivalent	Dentes Dents
Preferred_name	Dentition
Synonym	Teeth Set of teeth Teeth set
label	Dentition
subClassOf	Set of organs

Fig. 5. Properties of the term Dentition in the Taxonomy of Anatomical Entity Template of FMA

information about the image and the anatomical area of interest. The latter is inferred by following a machine learning approach as the proposed by Criminisi et al [8,9], or by manual annotations provided by an expert. Also, PSL predicates are considered as a possible better approximation of this region with non-zero probability. This is done by considering the neighborhood around the region of interest and knowing that *dentition*, for example, should not be located around the eyes or upper areas of the head; voxels belonging to *dentition* should be closer around an area, and distance between dentition voxels should not be longer than a certain threshold.

Another inference process to adjust the probability for points is performed by considering knowledge derived from ontology relationships, i.e., the classification of the term *dentition* in the Anatomical Set branch shown in Figures 5, and 6. Considering the transitive property of `constutionalPart`, `regionalPart` and `SubClass` (see Fig. 6), a seed point is annotated to identify different properties of the term *dentition*: *i*) it is a set of organs, *ii*) it is a member of the maxillary and mandibular dentition, and *iii*) it is part of the mouth. Current set of rules is used to segment any tissue as long as it can be enclosed in a bounding box. Therefore, segmentation of other tissues (e.g., nerves, vessels) or body nonstructural components (e.g., bodily fluids, hormones) may be inaccurate, or they will not be segmented at all. The voxel neighborhood detection algorithm is performed using PSL predicates for that particular seed point considering

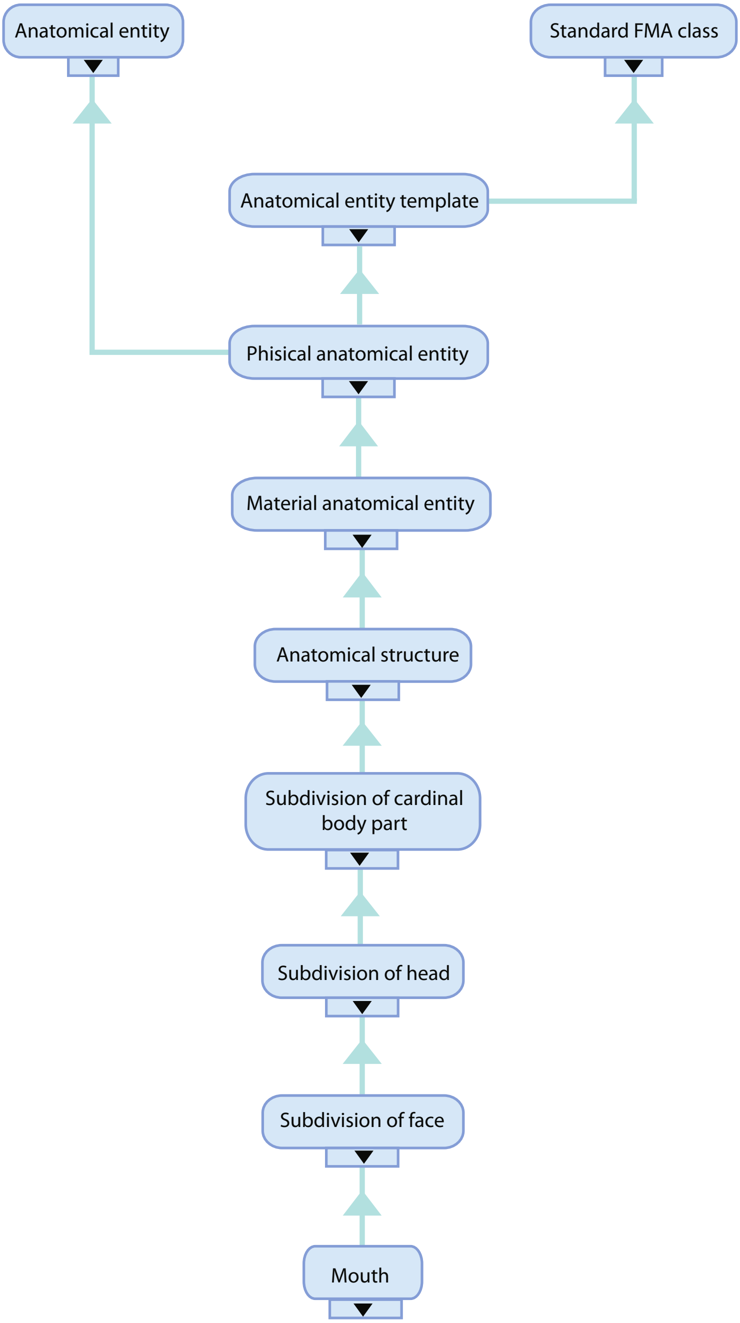

Anatomical entity
Standard FMA class
Anatomical entity template
Phisical anatomical entity
Material anatomical entity
Anatomical structure
Subdivision of cardinal body part
Subdivision of head
Subdivision of face
Mouth

Fig. 6. Portion of FMA that includes the Mouth as Part of the Face in the Taxonomy of Anatomical Entity Template

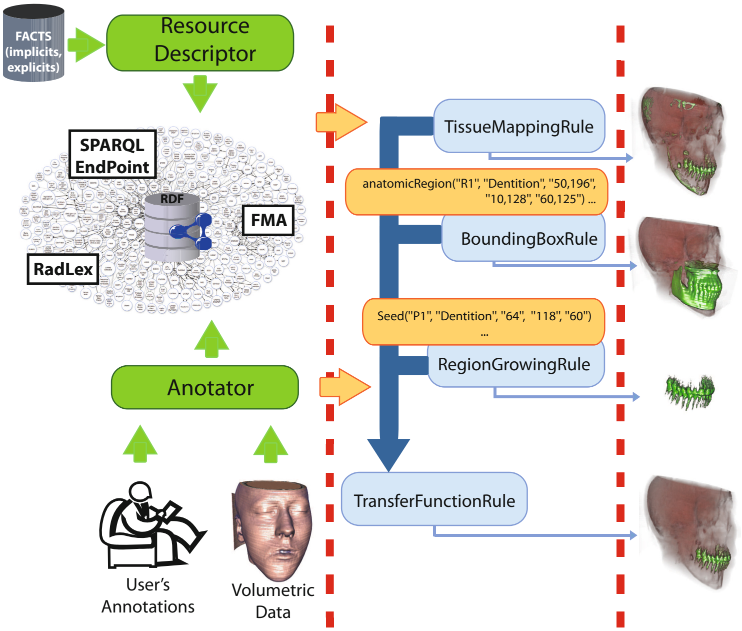


Fig. 7. The ANISE Workflow

`regionalPart` and `constutionalPart`. Finally, combining all inferred facts and probabilities of the given points, likelihood of points that represent a particular tissue are estimated; Fig. 7 illustrates the whole process. Further, appropriate TFs for each region are defined and performed. This is done just using the same TF (Fig. 3(b)) but performing a reasoning task that allows to detect the voxels that semantically do not correspond to the tooth tissue and that should not be included in the final volume rendering (see Fig. 3(c)). Additionally, an RDF document is generated to describe the image and the tissues that are part of this image. A fragment of the generated RDF document is presented in Listing 1.2.

Listing 1.2. Fragment of the annotation results as n-triples for Skewed Head volumetric data

```
 @prefix dc: <http://purl.org/dc/elements/1.1/> .
 @prefix rad: <http://purl.bioontology.org/ontology/RID/> .
 @prefix fma: <http://sig.uw.edu/fma#> .
 @prefix swg: <http://www.ldc.usb.ve/SWG/> .
 #...
<swg:data/skewed_head.dat> <dc:title> "Skewed Head" .
<swg:data/skewed_head.dat> <dc:description> "Computed Tomography Scan of Head" .
<swg:data/skewed_head.dat> <dc:date> "2012-03-26" .
<swg:data/skewed_head.dat> <dc:format> "image/dat" .
 #...
<swg:data/skewed_head.dat> <rad:RID10461> <rad:RID10462> .
<swg:data/skewed_head.dat> <rad:RID10311> <rad:RID10321> .
<swg:data/skewed_head.dat> <rad:RID13066> <fma:Dentition> .
<swg:data/skewed_head.dat> <rad:RID13063> <fma:Head> .
<swg:data/skewed_head.dat> <fma:Anatomical_structure> <fma:Head> .
```

```
16  #...
17  <swg:data/skewed_head.dat> <fma:cube> _:R1 .
18  _:R1 <fma:Anatomical_structure> <fma:Mouth> .
19  _:R1 <fma:Anatomical_coordinate_point> "50 10 60" .
20  _:R1 <fma:Anatomical_coordinate_point> "196 128 125" .
21  #...
22  <swg:data/skewed_head.dat> <fma:cube> _:R2 .
23  _:R2 <fma:Anatomical_structure> <fma:Dentition> .
24  _:R2 <fma:Anatomical_coordinate_point> "64 20 65" .
25  _:R2 <fma:Anatomical_coordinate_point> "192 118 102" .
26  #...
27  <swg:data/skewed_head.dat> <fma:Point> _:P1 .
28  _:P1 <fma:Anatomical_structure> <fma:Dentition> .
29  _:P1 <fma:Anatomical_coordinate_point> "64 118 60" .
30  #...
```

These annotations are received as input or are automatically generated from ANISE. For example line 11 in Listing 1.2, the term $RID10461$ corresponds to the patient orientation (*patient_orientation*) in RadLex ontology which match to the term $(0018, 5100)$ (*PatientPosition*) on DICOM header (see Listing 1.1). Hence, *HFS* value in field $(0018, 5100)$ is mapped to term $RID10462$ (*head_first*). Thus, in line 12 $RID10311$ (*imaging_modality*) match with $(0008, 0060)$ field on DICOM header; and, CT value is mapped to RadLex term $RID10321$ (*computed_tomography*). These annotations are generated automatically. Some others annotations are given by the user, e.g., anatomical region and the region of interest for visualizing. Line 14 in Listing 1.2 presents the term $RID13063$ (*body_region_covered*) which is used to describe the general content of the data in anatomic terms by using FMA term $Head$. Furthermore, the FMA Ontology term *Anatomical_structure* is mapped to $Head$ to express that main anatomical structure correspond to Head; this allows to correlate both ontologies (RadLex and FMA) at this point (see line 15 in Listing 1.2). For visualizing, the rendering focus is set using the term $RID13066$ (*area_of_interest*) valued with the FMA term *Dentition*.

These annotations represent knowledge about the resource; e.g., bounding box regions containing particularized tissues are defined. Such is the example on lines 17-20 in Listing 1.2, where a cubic region is defined by two points (*anatomical_coordinate_point*) to contain any particular tissue (*anatomical_structure*). Seed point is annotated using a similar method defining a single Point and its coordinates (*anatomical_coordinate_point*) and the tissue that it represents (*anatomical_structure*), shown on lines 27-29 in Listing 1.2.

Using these generated annotations, rules previously described are executed by specifying the terms for ground predicates. For example, term $RID10321$ specifying a computed tomography is related to the *Modality("CT")* predicate and is used in rule (8). Other annotations are used to apply the segmentation rules; for example, rule (9) by mapping annotations expressed in lines 17-20 into predicate *anatomicRegion("R1", "Dentition", "50,196", "10,128", "60,125")*. The annotated seed point is the ground predicate *Seed("P1", "Dentition", "64", "118", "60")* used on rule 10 (*RegionGrowingRule*). Finally, note that these rules are used to infer the facts required to visualize both the anatomic region and the tissue of interest.

5 Conclusions and Future Work

We treat the problem of processing and describing medical images as semantically annotated resources whose semantics is exploited to improve the quality of the image visualization and resource description. As proof-of-concept, we present ANISE, a framework that exploits knowledge encoded by ontology annotations of 3D medical images, and enhances the rendering process of the images. Quality of ANISE renderings has been studied in different images, and we have observed that they can accurately locate tissues that comprise a medical image. Annotations allow identifying or validating patterns on images, accurate image retrieval, applying the visualization process on regions of interest, and generate RDF documents that describe these regions and the tissues enclosed by them. Methods to filter relevant information have been developed at a high abstraction level, allowing extension of the inference process to perform particular algorithms, i.e., voxel neighborhood predicates could be improved to allow different methods. In the future, we plan to enhance the rule-based system to normalize a wider range of conditions, and include different image modalities (e.g., MR, and PET) as well as the tissues (e.g., blood vessels). Furthermore, we will extend tissue identification algorithms and rules to: *i*) detect and annotate anomalies, and *ii*) identify special conditions on tissues inside the region of interest. Development of visualization algorithms to consider not only TF definitions, but also different interpretations of semantic annotations of particular tissues of interest and its corresponding representation of rendered images, are also part of our future work. Finally, we will conduct a user study with more datasets and different user requests to analyze the quality of the proposed approach.

References

1. Baranya, A., Landaeta, L., LaCruz, A., Vidal, M.-E.: A workflow for improving medical visualization of semantically annotated ct-images. In: SATBI+SWIM 2012 in Conjunction with ISWC 2012 (2012)
2. Bloehdorn, S., et al.: Semantic annotation of images and videos for multimedia analysis. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 592–607. Springer, Heidelberg (2005)
3. Broecheler, M., Mihalkova, L., Getoor, L.: Probabilistic similarity logic. In: Conference on Uncertainty in Artificial Intelligence (2010)
4. Bühler, K., Felkel, P., La Cruz, A.: Geometric methods for vessel visualization and quantification – a survey. In: Geometric Modelling for Scientific Visualization, pp. 399–420. Springer (2002)
5. Carneiro, G., Vasconcelos, N.: A database centric view of semantic image annotation and retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2005, pp. 559–566. ACM, New York (2005)
6. Ceri, S., Gottlob, G., Tanga, L.: What you always wanted to know about datalog (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering 1(1) (1989)

7. Chan, A.B., Moreno, P.J., Vasconcelos, N.: Using statistics to search and annotate pictures: an evaluation of semantic image annotation and retrieval on large databases. In: Proceedings of Joint Statistical Meetings, JSM (2006)
8. Criminisi, A., Shotton, J., Bucciarelli, S.: Decision forests with long-range spatial context for organ localization in ct volumes. In: MICCAI Workshop on Probabilistic Models for Medical Image Analysis (MICCAI-PMMIA). Springer (2009)
9. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision 7(2-3) (2012)
10. Cruz-Roa, A., Díaz, G., Romero, E., González, F.A.: Automatic annotation of histopathological images using a latent topic model based on non-negative matrix factorization. J. Pathol. Inform. 2(4) (2011)
11. `http://www-graphics.stanford.edu/data/voldata/CThead.tar.gz`
12. Freedman, D., Radke, R., Zhang, T., Jeong, Y., Lovelock, D., Chen, G.: Model-based segmentation of medical imagery by matching distributions. IEEE Transactions on Medical Imaging 24(3), 281–292 (2005)
13. Gerl, M., Rautek, P., Isenberg, T., Gröller, E.: Semantics by analogy for illustrative volume visualization. Computers & Graphics 36(3), 201–213 (2012)
14. Huang, H.K.: PACS and Imaging Informatics: Basic Principles and Applications. John Wiley and Son Inc., Hoboken (2010)
15. Kirbas, C., Quek, F.: A review of vessel extraction techniques and algorithms. ACM Computing Surveys 36(2), 81–121 (2005)
16. Ko, B.C., Byun, H.: Query-by-gesture: An alternative content-based image retrieval query scheme. Journal of Visual Languages & Computing 13(4), 375–390 (2002)
17. Lipscomb, C.E.: Medical subject headings (mesh). Bulletin of the Medical Library Association 88(3), 265 (2000)
18. Möller, M., Mukherjee, S.: Context-driven ontological annotations in dicom images: Towards semantic pacs. In: Proceedings of International Joint Conference on Biomedical Engineering Systems and Technologies (2008)
19. Olabarriaga, S., Smeulders, A.: Interaction in the segmentation of medical images: A survey. Medical Image Analysis 5(2), 127–142 (2001)
20. Peng, H.: Bioimage informatics: a new area of engineering biology. Bioinformatics 24(17), 1827–1836 (2008)
21. Preim, B., Bartz, D.: Visualization in Medicine: Theory, Algorithms, and Applications. The Morgan Kaufmann Series in Computer Graphics (2007)
22. Rautek, P., Bruckner, S., Gröller, E.: Semantic layers for illustrative volume rendering. IEEE Trans. Vis. Comput. Graph. 13(6), 1336–1343 (2007)
23. Rosse, C., Mejino, J.: The foundational model of anatomy ontology. In: Anatomy Ontologies for Bioinformatics: Principles and Practice. The Morgan Kaufmann Series in Computer Graphics (2007)
24. Rubin, D.L., Mongkolwat, P., Kleper, V., Supekar, K., Channin, D.S.: Medical imaging on the semantic web: Annotation and image markup. In: AAAI Spring Symposium: Semantic Scientific Knowledge Integration, pp. 93–98. AAAI (2008)
25. Rubin, D.L., Rodriguez, C., Shah, P., Beaulieu, C.: iPad: Semantic Annotation and Markup of Radiological Images. In: AMIA Annual Symposium Proceedings, pp. 626–630. PMC US National Library of Medicine National Institute of Health (2008)
26. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikinis, R.: Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. Medical Image Analysis 2(2), 143–168 (1998)

27. http://www.cg.tuwien.ac.at/courses/Visualisierung/1999-2000/skewed_head.zip
28. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W.-L., Wright, L.W.: Nci thesaurus: A semantic model integrating cancer-related clinical and molecular information. J. of Biomedical Informatics 40(1), 30–43 (2007)
29. http://mri.radiology.uiowa.edu/VHDicom/VHMCT1mm/VHMCT1mm_Head.tar.gz
30. Wei, W., Barnaghi, P.M.: Semantic support for medical image search and retrieval. In: Proceedings of the Fifth IASTED International Conference: Biomedical Engineering, BIEN 2007, Anaheim, CA, USA, pp. 315–319. ACTA Press (2007)
31. Williams, D., Grimm, S., Coto, E., Roudsari, A., Hatzakis, H.: Volumetric curved planar reformation for virtual endoscopy. IEEE Transactions on Visualization and Computer Graphics 14(1), 109–119 (2008)
32. Yang, Y., Huang, S., Lin, P., Rao, N.: Medical image segmentation based on level set combining with region information. In: Fourth International Conference on Natural Computation, ICNC 2008, vol. 5, pp. 70–74 (October 2008)
33. Yoon, S.M., Kuijper, A.: Query-by-sketch based image retrieval using diffusion tensor fields. In: 2010 2nd International Conference on Image Processing Theory Tools and Applications (IPTA), pp. 343–348 (July 2010)
34. Yu, F., Ip, H.H.-S.: Automatic semantic annotation of images using spatial hidden markov model. In: Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, ICME 2006, Toronto, Ontario, Canada, July 9-12, pp. 305–308. IEEE (2006)

Towards an Efficient Datalog Based Evaluation of the FSAQL Query Language

Afef Bahri, Rafik Bouaziz, and Faïez Gargouri

University of Sfax, MIRACL Laboratory,
Route de M'harza Km 1.5; B.P.: 1030 Sfax: 3018, Tunisia
afef.bahri@gmail.com, {raf.bouaziz,faiez.gargouri}@fsegs.rnu.tn

Abstract. The resources in the Semantic Web are described using particular metadata called "Semantic annotations". A semantic annotation is a particular case of annotation which refers to ontology. The Web content is, for the most part, subject to uncertainty or imperfection. FSAQL is a fuzzy query language proposed to query semantic annotations defined with fuzzy RDFS. The efficiency of Datalog systems to query large amount of data has been proven in the literature. We propose in this paper an efficient Datalog based approach for the evaluation of the FSAQL query language. The particularity of our approach consists on the fact that we use crisp Datalog programs instead of fuzzy ones. In fact, there is no known implementation of fuzzy Datalog systems and the use of crisp Datalog allows the interporability of our query language. Two approaches have been proposed for a correct mapping of fuzzy RDFS to crisp Datalog programs. The defuzzification approach defines crisp "$\alpha-$cut" classes and properties and maps them to crisp Datalog predicates. The skolemisation approach represents fuzzy classes and properties with crisp Datalog predicates having the same names. The membership degrees are then defined as terms of theses predicates. The two approaches are implemented and evaluated using the $\mathcal{F}$lora-2 Datalog system.

1 Introduction

The Semantic Web is an infrastructure that enables the interchange, the integration and the reasoning about information on the Web. One of the most important layer of the Semantic Web is hence the ontology layer. In fact, it provides the conceptual structure that can be used to describe web resources and opens up opportunities for automated information processing. In the Semantic Web, the resources are described using particular metadata called "Semantic annotations". Their purpose is to assign to objects of a resource a meaning using ontology terms. Many formal standard representations like RDF, RDFS and OWL may be used to allow the representation of web resources in a common and unified way. All these formalisms are based on crisp logic and suppose an exact definition of the resources. In the real world, however, information is subject to imperfection. Also, a web resource is itself source of imperfection and by this way,

Z. Lacroix, E. Ruckhaus, and M.-E. Vidal (Eds.): RED 2012, LNCS 8194, pp. 150–180, 2013.

its annotation is often inexact or uncertain. For these reasons, many extensions of ontology languages have been proposed in the literature to deal with uncertainty and vagueness using probabilistic [9], possibilistic [16], bayesian [10] or fuzzy [29] logics. Fuzzy extensions still the most studied ones [12,21,29,4]. Fuzzy extension of OWL have been proposed in [12,29] and fuzzy extensions of RDF and RDFS have been proposed in [21]. If many approaches have been proposed to extend ontology languages on the Semantic Web, the problem of fuzzy ontology querying is not well treated. The only works realized on this subject propose to use persistent storage system to take advantage from database capabilities for efficient query processing over large fuzzy knowledge bases [23,24,27,28,30]. As we will see in section 3, this may increases the inference cost needed to maintain the completeness and the consistency of the knowledge base stored in a database and makes restriction on the expressivity of the used ontology language as not all ontology descriptors may be defined using database schemas. Motivated by this concern, we propose to use Datalog systems which combine storage, querying and reasoning capabilities over large amount of Data for efficient evaluation of fuzzy ontologies queries. The evaluation of the proposed approach is made on the FSAQL (*Fuzzy Semantic Annotation Query Language*) query language that we proposed [2].

The particularity of our approach consists on the fact that we use crisp Datalog programs instead of fuzzy ones. In fact, there is no known implementation of fuzzy Datalog systems and the use of crisp Datalog allows the interoperability of our query language. Two approaches have been proposed for a correct mapping of fuzzy RDFS to crisp Datalog programs.The defuzzification approach defines crisp "$\alpha-$cut" classes and properties and map them to crisp Datalog predicates. The skolemisation approach represents fuzzy classes and properties with crisp Datalog predicates having the same names. The membership degrees are then defined as terms of theses predicates. The two approaches are implemented in the $\mathcal{F}$lora-2 system which is an object-oriented knowledge base language and application development environment based on the XSB datalog system. Many criteria are used to evaluate the efficiency of the two approaches: the number of membership degrees, the size of the knowledge base, the complexity of the query and the expressivity of the ontology language. In fact, if the last version of FSAQL is used to query fuzzy RDFS, in this paper, we enrich the expressivity of RDFS with some OWL2EL descriptors.

The rest of the paper is organized as follows. Firstly, in section 2, we briefly recall some basics on Fuzzy Logic and fuzzy ontologies. A discussion of some related works is given in section 3. A short theoretical description of the fuzzy RDFS data model is made in section 4. Section 5 defines the syntax and the semantics of the FSAQL query language. In section 6, two approaches are proposed for a correct mapping of fuzzy RDFS into crisp Datalog programs. Section 7 proposes a Datalog based evaluation of FSAQL queries. Section 8 discusses implementations issues, presents and compares some experimental results. Finally, conclusions and future works are exposed in section 9.

2 Preliminaries

2.1 Fuzzy Logic

Fuzzy set introduced by [34] is a natural extension to the classical crisp set where either an object is a member of a set or it is not a member of a set. Classical two-valued logic applies when the set has crisp boundaries but in real-world this is rarely the case.

Each fuzzy set is fully defined through its *membership function* that maps the elements of the interest domain—often called *universe of discourse*—to [0,1]. Mathematically, let U be the universe of discourse and F a fuzzy set defined on U. Then, the membership function associated with the fuzzy set F is defined as follows:

$$\begin{aligned} \mu_F\colon U &\to [0,1] \\ u &\mapsto \mu_F(u) \end{aligned}$$

The function μ_F associates to each element u of U a degree of membership (d.o.m) $\mu_F(u)$ in the range [0,1]; where 0 implies no-membership and 1 implies full membership. A value between 0 and 1 indicates the extent to which u can be considered as an element of fuzzy set F.

There are many membership functions for fuzzy sets membership specification. For example, triangular, trapezoidal, open ended right and open ended left functions. These functions are defined on non-negative reals. They are characterized by parameters used to calculate membership degrees. The trapezoidal function, for example, is characterized by four parameters α, β, γ and δ. The parameters β and γ represent the support of the fuzzy set associated with the attribute value and α and λ represent the limits of the transition zones. Figure 1 shows a graphical representation of a trapezoidal membership function. This function is defined as follows:

$$\mu(x;\alpha,\beta,\gamma,\delta) = \begin{cases} 1, & \text{if } \beta \leq x \leq \gamma; \\ \frac{\lambda-x}{\lambda-\gamma}, & \text{if } \gamma < x < \lambda; \\ \frac{x-\alpha}{\beta-\alpha}, & \text{if } \alpha < x < \beta; \\ 0, & \text{otherwise.} \end{cases}$$

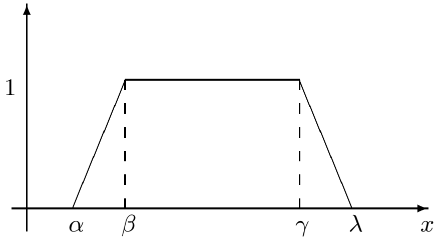


Fig. 1. Trapezoidal membership function

2.2 Fuzzy Ontologies

An ontology is a shared model of some domain which is often conceived as a typically hierarchical data structure containing all relevant concepts and their relationships. A Fuzzy Ontology can be defined as consisting of fuzzy concepts, of fuzzy relations among concepts and a set of instances. More precisely, a fuzzy concept C is considered as a fuzzy set over a universe of discourse E, thus an instance does not fully belong or not to C but possesses a membership degree to this concept. This is defined by a membership function $\mu_C : E \to [0, 1]$, which given an instance $a \in E$ returns the degree of membership of a being an instance of C. The specification of a concept defines the conditions that its instances must verify. In fuzzy case, these conditions may be partially satisfied as fuzzy concepts are not precisely defined. The degree of membership of one instance to a fuzzy concept may be seen as the degree of satisfaction of the corresponding conditions. The concept *ExpensiveBook* may be defined with fuzzy description logic as following:

$$ExpensiveBook = Book \sqcap \exists price.Expensive$$

where *Expensive* is a linguistic label and we may define it with a membership function like trapezoidal or triangular distribution.

A fuzzy relation R is equally defined by a membership function over $E * E$, defined by the function $\mu_R : E * E \to [0, 1]$. Fuzzy relations denote a type of interaction between concepts of the ontology's domain. Ontology relations may be classified into two types: taxonomic relations and descriptive relations. We may found many kinds of taxonomic relations to express for example synonymy, composition or subsumption. Subsumption known equally as Is-a relation is the most adopted taxonomic relation. Is-a relation permits property inheritance from ancestors to descendants and allows concepts to have more specific sub-concepts. The notion of membership degree may be extended to the Is-a relation which means that two fuzzy concepts may have a degree to be related by an Is-a relation. This degree may be attributed by the user or computed by specific membership functions.

Descriptive relations between concepts depend generally on the domain or the data we want to model and are used to define relation between instances. For example, we may define a descriptive relation named *isNear* between the concept *Hotel* and the concept *Airport*. We can equally associate a degree in [0,1] to the instantiated relations. For example the *Hotel Ibis* is related to the *Airport CDG* with the relation *isNear* with a degree of membership equal to 0.7. The difference between membership degrees associated to fuzzy taxonomic relations and descriptives ones is that the first quantifies relations between concepts while the second quantifies relations between instances.

3 Related Works

In the literature, the problem of fuzzy ontology querying is generally treated as a kind of reasoning [31,23,24,27,28]. A principle equally adopted in crisp description logics [1,13]. Given a fuzzy ontology O, a conjunctive threshold query is of the form [28]:

$$q(X) \leftarrow C(x) \geq \alpha, R(y,z) \geq \beta$$

C and R correspond respectively to a concept and a role and α, β are real values $\in$ [0,1]. Given an evaluation $[X \mapsto S]$, S is a solution for $q(X)$ iff for each model I of O we have $C^I(x)_{[X \mapsto S]} \geq \alpha$ (resp. $R^I(y,z)_{[X \mapsto S]} \geq \beta$). For example, we define in the following a conjunctive threshold query:

$$q(x) \leftarrow Model(x) \geq 1.0, Tall(x) \geq 0.7$$

Many variations of threshold queries have been proposed such as fuzzy threshold, fuzzy aggregation and fuzzy weighted queries [27,28]. For example, in a fuzzy threshold query, an evaluation $[X \mapsto S]$ has a degree $\in [0,1]$ to satisfy a query.

The first works realized on this topic are those on fuzzy DL-Lite ontologies [30,23,24] where fuzzy ontologies are supposed to be stored in a relational database. That is the fuzzy knowledge base is closed based on some inference rules and stored in a database in order to effectively retrieve instances. In [30], query answering procedure closely follows the crisp DL-Lite querying approach proposed in [8]. The fuzzy query is reformulated into a set of conjunctive queries which are evaluated as SQL queries over the fuzzy knowledge base stored previously in a relational database. The authors of [23,24] propose more expressive fuzzy like queries over fuzzy knowledge bases defined equally with a fuzzy extension DL-Lite. The queries are reformulated using a fuzzy extension of SPARQL, named fSPARQL, and evaluated over the ONTOSEARCH2 system [22]. For example, the conjunctive threshold query $q(x)$ is written with fSPARQL as follows [24]:

```
SELECT ?x
WHERE { ?x rdf:type Model #TH# 1.0
        ?x rdf:type Tall  #TH# 0.7 }
```

#TH# is used to specify a threshold and is replaced with #DG# in the case of fuzzy threshold queries [24]:

```
SELECT ?x
WHERE { ?x rdf:type Model #DG# 1.0
        ?x rdf:type Tall  #DG# 0.7 }
```

Despite the fact that the use of fuzzy DL-Lite enables query answering procedures by making use of database, DL-Lite is not expressive enough to support all the descriptors used in OWL-DL or even in OWL-Lite or RDFS. This makes its use restricted to some applications domains in which the use of some ontology descriptors such as role inclusion is not mandatory. In this sense, the authors of [28] proposed an approach for querying fuzzy ontologies defined with a fuzzy extension of the description logic $\mathcal{SHIN}$ [32]. Close to the works realized on fuzzy DL-Lite [30,23,24], fuzzy ontologies are supposed to be stored in a persistent storage system. The authors propose an approach to serialize f$-\mathcal{SHIN}$ knowledge base into RDF triple based on a fuzzy OWL to RDF mapping using blank nodes. The same typology of queries proposed in fuzzy DL-Lite approaches are equally used in this work. Fuzzy queries defined on the Fire fuzzy reasoning engine [26] are translated into SPARQL and executed over the Sesame RDF store. The conjunctive threshold query $q(x)$ is written with SPARQL as follows [28]:

```
SELECT ?x
WHERE { ?x rdf:type Model
        ?x O:degreeModel ?dom1
        ?x rdf:type Tall
        ?x O:degreeTall ?dom2
        Filter (?dom1 >= "1.0∧∧xsd:float")
        Filter (?dom2 >= "0.7∧∧xsd:float") }
```

Unfortunately and despite the use of f$-\mathcal{SHIN}$, the proposed approach does not really allow the querying of expressive description logics. In fact, like the fuzzy DL-Lite approaches, the evaluation of a query directly invoke the RDF triple store and no f$-\mathcal{SHIN}$ reasoning system is involved. The Fire inference engine is invoked before storing fuzzy knowledge into the RDF triple store in order to materialize as much implicit knowledge as possible. This make the approach complete only for ground conjunctive queries. The problem of efficiently querying expressive description logics is only partially resolved and still an open problem. Equally, what we may deduce is that the fuzzy DL-Lite approaches as well as f$-\mathcal{SHIN}$ maximize the querying efficiency by storing all the inferred data in a persistent storage instead of calculating them on the fly. This may cause a limitation in the expressivity of the used ontology language as it is shown in [28]. Equally, this manipulation needs to be realized each time we want to maintain the consistency and the completeness of the knowledge base which may increase the inference cost [17,25]. A common approach which provides persistent storage and reasoning capabilities is the use of Datalog systems. In this paper, we propose an approach which takes advantage from Datalog systems for evaluating FSAQL queries over large amount of fuzzy ontology instances.

The integration of ontology languages with rules and logic programming has attracted the interest of many researches [14,19,15,7,20,11]. The rules languages which are subject of theses integration are the ones based on Horn clausal logics. The integration of the two paradigms should play an important role in the Semantic Web. In fact, despite their inference capabilities over complex TBoxes, DL reasoners have a high ABox reasoning complexity which may constitute a serious limitation in the Semantic Web where we rely mainly on query answering (i.e. instance checking). The subsomption algorithm, for example, reduces instance checking into concept satisfiability. That is, to retrieve the instances of a given concept, we need to run the subsumption algorithm for each individual in the ABox. In Datalog systems, query answers are computed in one pass (i.e. bottom-up, top-down). Two principal integration approaches are used in the literature: the hybrid and the homogenous approaches. In the hybrid approach, the two paradigms are used and a knowledge base $\mathcal{KB}$ is defined as $\mathcal{KB} = \langle \mathcal{LD}, \mathcal{PL} \rangle$ where $\mathcal{LD}$ is defined with description logic and $\mathcal{PL}$ is defined with Logic Programming. In hybrid approach, the reasoning over DL ontologies is performed only by the DL reasoner. The rules are used to define constraints on the defined ontology and rule engine are just used for rule execution. While in hybrid approaches, the rules and ontologies are treated separately, in homogenous approaches, rules and ontology are combined in a new single logic language. In practice, the description logic is mapped into a rule based formalisms known as description logic programs. Such an integration approach may lead to undecidability of reasoning problems due the opposite assumption (Closed world and Open world assumption) of the two paradigms. Decidability may be obtained by restricting rules to DL-safe ones [19]. Homogenous approaches perform reasoning only by rules engine.

The problem of combination of rules and ontologies has equally been treated for fuzzy ontologies. The majority of the works realized on this subject propose homogenous integration approaches in which fuzzy DL programs are defined as a result of the integration of fuzzy DLs and fuzzy rule languages [33]. The reasoning should be performed by fuzzy inference engines. The fact that there is no common implementation of fuzzy rule engines, theses works have only focused on the theoretical aspects of the integration. In our approach, we equally use an homogenous approach but we choose a different way for the combination of rules and fuzzy ontologies as we use crisp rule language instead of fuzzy ones. The issue that we have faced consists on the representation of membership degrees, which constitute the principle characteristic of fuzzy logic. Two common ways may be adopted: the first way inspired from [4], which we call the defuzzification approach, defines "$\alpha - cut$" classes and properties and maps them into Datalog predicates; the second, which we call the skolemisation approach consists on representing membership degrees using terms of Datalog predicates representing fuzzy classes and properties. The proposed approaches are equally implemented and evaluated using $\mathcal{F}$lora-2 and XSB Datalog systems.

4 Fuzzy RDFS

We propose in this section a fuzzy extension of RDFS to allow the representation of semantic annotations with imperfection. This fuzzy extension is based on the fact that RDFS statements may have imprecise definitions. RDFS properties used to define relationships between resources are associated to a degree $\in [0, 1]$ to denote their degree of truth. On the other hand, RDFS properties used to define attributes of resources (attribute-value pairs) can take fuzzy values: linguistic label, fuzzy range or approximate value. Fuzzy attribute values are associated to membership functions (ex. trapezoidal, triangular). In order to model fuzzy attribute values with RDFS we define a new class named `membershipFunction` instance of the resource `Class`. All membership functions are defined as subclasses of this class.

A fuzzy RDFS statement is an RDFS statement associated to a real value in [0,1]. `n:`(s,p,o) is a fuzzy RDFS statement where (s,p,o) is an RDFS triple and `n` is a real value in [0,1]. A crisp RDFS statement (s,p,o) is equivalent to the fuzzy RDFS statement 1:(s,p,o). The RDFS `rdfs:subClassOf` and `rdfs:subPropertyOf` properties are extended to fuzzy logic and have different semantics in fuzzy RDFS.

A fuzzy RDFS interpretation I is defined by:

- IR: a non empty set of resources.
- IC: the set of all classes in the interpretation.
- IP: the set of properties of I.
- $IEXT$: a fuzzy mapping from IP to $IR \times IR$. Given a property p and two resources x and y, $\mu_{IEXT(p)}(x, y) \in [0, 1]$.
- $ICEXT$: a fuzzy mapping from IC to IR:
  - $IC = ICEXT(I(\texttt{rdfs:Class}))$.
  - $IR = ICEXT(I(\texttt{rdfs:Resource}))$.
  - $LV = ICEXT(I(\texttt{rdfs:Literal}))$.

Given a vocabulary V, a fuzzy RDFS interpretation I of `range`, `domain`, `type`, `subProperty` and `subClassOf` properties satisfies the following conditions:

- If E is a ground statement `n:(s,p,o)` then $I(E)$ = *true* if `s`,`p` and `o` are defined in V, $I(\mathrm{p}) \in IP$ and $\mu_{IEXT(I(p))}(I(s), I(o)) \geq n$, otherwise $I(E)$ = *false*.
- If E is a fuzzy ground RDFS graph, then $I(E)$ = *false* if $I(E')$ = *false* for some $E' \in E$, otherwise $I(E)$ = *true*.
- $x \in ICEXT(y)$ iff $\langle x,y\rangle \in IEXT(I(\texttt{rdf:type}))$.
- If $\langle x,y\rangle \in IEXT(I(\texttt{rdfs:domain}))$ and $\langle u,v\rangle \in IEXT(x)$ then $u \in ICEXT(y)$.
- If $\langle x,y\rangle \in IEXT(I(\texttt{rdfs:range}))$ and $\langle u,v\rangle \in IEXT(x)$ then $v \in ICEXT(y)$.
- If $\mu_{IEXT(I(rdfs:subClassOf))}(\langle c, d\rangle) = \alpha$ then c and $d \in IC$ and

$$\min(\mu_{ICEXT(I(c))}(x), \alpha) \leq \mu_{ICEXT(I(d))}(x) \forall x \in IR$$

- If $\mu_{IEXT(I(rdfs:subPropertyOf))}(\langle p, q\rangle) = \alpha$ then p and $q \in IP$ and

$$\min(\mu_{IEXT(I(p))}(x, y), \alpha) \leq \mu_{IEXT(I(q))}(x, y) \forall x, y \in IR$$

We note that the choice of a fuzzy subsumption measure influences the reasoning process. The entailment rules defined for RDFS `subClassOf` and `subPropertyOf` properties are based on their transitivity. If we want to extend these entailment rules to fuzzy logic, we need to use a transitive fuzzy subsumption measure. That is if `i`:(`u`, `rdfs:subClassOf`, `v`) and `j`:(`v`,`rdfs:subClassOf`,`x`) are explicitly defined in a fuzzy RDFS graph, we can infer that `k:`(`u`,`rdfs:subClassOf`,`x`) where `k` is defined as a function of `i` and `j`. To allow transitivity we suppose that `k`=min(`i`,`j`).

5 The FSAQL Query Language

FSAQL is a declarative query language that we proposed in [2] to query fuzzy semantic annotations in the Semantic Web . We present in this section the

Table 1. Fuzzy RDFS entailment rules

rule	If G contains	then add
1	i:xxx,aaa,yyy	1:aaa,rdf:type,rdf:Property
2	i:aaa,rdfs:domain,zzz j:xxx,aaa,yyy	1:aaa,rdf:type,zzz
3	iii:xxx,rdfs:range,zzz j:xxx,aaa,uuu	1:xxx,rdf:type,zzz
4a	i:uuu,aaa,xxx	1:uuu,rdf:type,rdfs:Resource
4b	i:uuu,aaa,vvv	1:vvv,rdf:type,rdfs:Resource
5	i:uuu,rdfs:subPropertyOf,vvv j:vvv,rdfs:subPropertyOf,xxx	k:uuu,rdfs:subPropertyOf,xxx k= min(i,j)
6	i:xxx,rdf:type,rdf:Property	1:xxx,rdfs:subPropertyOf,xxx
7	i:aaa,rdfs:subPropertyOf,bbb j:uuu,aaa,yyy	k:uuu,bbb,yyy k = min(i,j)
8	i:uuu,rdf:type,rdfs:Class	i:uuu,rdfs:subClassOf,Resource
9	i:uuu,rdfs:subClassOf,xxx i:vvv,rdf:type,uuu	k:vvv,rdf:type,xxx k = min(i,j)
10	i:uuu,rdf:type,rdf:Class	i:uuu,rdfs:subClassOf,uuu
11	i:uuu,rdfs:subClassOf,vvv j:vvv,rdfs:subClassOf,xxx	k:uuu,rdfs:subClassOf,xxx k = min(i,j)
12	i:uuu,rdf:type,rdfs:Datatype	1:uuu,rdfs:subClassOf,rdfs:Literal

syntax and the semantics of FSAQL. The syntax of an FSAQL query is defined as follows:

```
SELECT           <variables>
FROM             <RDF repository>
WHERE            <RDF patterns>
HAVE VALUES      <value-constraints>
[WITH DOM <value>]
```

The query asks about a set of subgraphs that match the fuzzy RDFS query patterns in the **WHERE** clause. The argument of the **FROM** clause is a fuzzy RDFS repository. The argument of the **SELECT** clause is a list of variables (e.g. $?X$, $?Y$). The argument of the **HAVE VALUES** clause is a list of value constraints. Each fuzzy semantic annotation has a degree $\in [0, 1]$ defined in the **WITH DOM** clause to respond to the query.

An FSAQL graph pattern is defined recursively as follows:

1. A triple pattern $t \in (I \cup B) \times I \times I \cup B \cup L$ associated to a degree $n \in [0, 1]$ is an FSAQL graph pattern.
2. If P_1 and P_2 are graph patterns then (P_1 AND P_2) and (P_1 UNION P_2) are graph patterns.
3. If P is a graph pattern and R a value constraint then (P HAVE VALUES R WITH DOM n) is a graph pattern.

Given a fuzzy graph pattern P and a fuzzy RDFS graph G. $[[P]]_G$ denotes the RDF evaluation of a query pattern P in G. $[[P]]_G^{\texttt{rdfs}}$ denotes the RDFS evaluation of a query pattern P in G. The difference between $[[P]]_G$ and $[[P]]_G^{\texttt{rdfs}}$ consists on the fact that the RDFS evaluation $[[P]]_G^{\texttt{rdfs}}$ takes into consideration RDFS entailment rules and transitive closure of `rdfs:subClassOf` and `rdfs:subPropertyOf` properties which is not the case of RDF evaluation $[[P]]_G$ as it constitutes a simple subgraph matching of the graph pattern P against the graph G based on a set of mappings σ from W to the terms of G:

$$[[P]]_G = \{\sigma : W \rightarrow T \mid dom(\sigma) = var(P) \text{ and } \sigma(P) \subseteq G\}$$

$dom(\sigma)$ is the subset of W where σ is defined and $var(P)$ is the set of variables appearing in the graph pattern P. If $\sigma \in [[P]]_G$, we say that σ is a solution for P in G or is the answer of the query (W,P). As we can see when we realize the mapping we do not consider the degrees of truth associated to fuzzy RDFS statements. In fact, we define conditions on these degrees of truth with value constraints in the `HAVE VALUES` statement.

The RDFS evaluation of an FSAQL *Select* query (W,P) over G is defined recursively as follows:

1. $[[P]]_G^{\mathtt{rdfs}} = [[P]]_G \cup [[P]]_G^{\mathtt{rule}_1} ... \cup [[P]]_G^{\mathtt{rule}_k}$. The degree of truth of the union is equal to the max of the degrees of truth of the arguments of the union.
2. $[[P]]_G^{\mathtt{rule}_i} = \Pi_{\mathtt{W},\mu(P)}$ $([[P_1]]_G^{\mathtt{rdfs}} ... \bowtie [[P_n]]_G^{\mathtt{rdfs}})$. The rule $\mathtt{rule}_i$ has P as a head and the P_i $(i$:1...$n)$ in the body (n is a finite number). μ(P) is the degree of truth obtained when the rule $\mathtt{rule}_i$ is invoked. $\mathcal{F}_{\mathtt{rule}_i}$ is the membership function associated to the entailment rules defined in table 1 (see section 4). μ(P) is obtained with $\mathcal{F}_{\mathtt{rule}_i}$. Π has the same definition of projection in relational algebra.
3. $[[P_i \text{ AND } P_j]]_G^{\mathtt{rdfs}} = [[P_i]]_G^{\mathtt{rdfs}} \bowtie [[P_j]]_G^{\mathtt{rdfs}}$
4. $[[P_i \text{ UNION } P_j]]_G^{\mathtt{rdfs}} = [[P_i]]_G^{\mathtt{rdfs}} \cup [[P_j]]_G^{\mathtt{rdfs}}$.
5. $[[P \text{ HAVE VALUES } R \text{ WITH DOM } n]]_G^{\mathtt{rdfs}} = \{ \sigma \in [[P]]_G^{\mathtt{rdfs}} \mid \sigma \models_n R \}$ which denotes the set of mappings in $[[P]]_G^{\mathtt{rdfs}}$ that satisfy R with a degree $\geq n$.

Given $?X$, $?Y \in V$ and $u \in I \cup L$, an FSAQL value constraint is defined as follow:

- $?X$ operator u, $?X$ operator $?Y$ where operator $\in \{=,<,>,\leq,\geq\}$ are atomic value constraints. When atomic value constraints are used to compare fuzzy values (ex. $?X$ = *Very Expensive*), we talk about fuzzy atomic value constraint and the used operator needs to be extended to fuzzy logic. A fuzzy value constraint returns a degree $\in [0, 1]$.
- If R_1 and R_2 are atomic value constraints, then $\neg R_1$, $R_1 \wedge R_2$, and $R_1 \vee R_2$ are value constraints. When R_1 and/or R_2 are fuzzy atomic value constraints, the $\neg$, $\wedge$ and $\vee$ need equally to be extended to fuzzy logic. The `WITH DOM` statement gives the degree of satisfaction of the corresponding fuzzy value constraint.

Given fuzzy value constraints R_1 and R_2, μ_{R1} (resp. μ_{R2}) $\in [0, 1]$ gives the degree of satisfaction of R_1 (resp. R_2). $\mu_{\neg R_1}$, $\mu_{R_1 \wedge R_2}$ and $\mu_{R_1 \vee R_2}$ are defined as follow:

- $\mu_{\neg R_1} = 1$ - μ_{R_1}
- $\mu_{R_1 \wedge R_2} = \min(\mu_{R_1}, \mu_{R_2})$
- $\mu_{R_1 \vee R_2} = \max(\mu_{R_1}, \mu_{R_2})$

6 Rewriting Fuzzy RDFS with Crisp Datalog Programs

We propose in this section two approaches for a correct mapping of fuzzy RDFS to crisp Datalog programs. The first approach is based on a defuzzification of the fuzzy RDFS descriptors. That is, based on the work of [4], we transform fuzzy RDFS into crisp RDFS by using α-cut classes and properties which are then defined with predicates and rules in the corresponding Datalog programs. The second approach is based on a skolemisation of fuzzy RDFS which consists on representing fuzzy classes and properties with Datalog predicates having the same names. The membership degrees are defined as terms on these predicates.

6.1 The Defuzzication Approach

To transform a fuzzy RDFS knowledge base fKB$_{rdfs}$ into a Datalog program P$_{rdfs}$, we adopt the same principle used in [4] for a crisp representation of fuzzy description logics. Fuzzy classes and properties are transformed into crisp $\alpha-$cut classes and properties. The membership degrees used to define $\alpha-$cut classes and properties are defined based on the Zadeh semantics as follows:

$$\mathcal{N} = X^{FK} \cup \{1-\alpha | \alpha \in X^{FK}\} \text{ where } X^{FK} = \{0, 0.5, 1\} \cup \{\gamma | \gamma : (s,p,o) \in fKB_{rdfs}\}$$

We propose in the following a list of mapping rules used to transform a fuzzy RDFS knowledge base fKB$_{rdfs}$ into a crisp Datalog program P$_{rdfs}$:

1. For each fuzzy RDFS statement α:<`a`,`rdf:type`,`c`> :

   (a) we define a fact of the form `type`(a, c$_{\geq\alpha}$) where c$_{\geq\alpha}$ is a new crisp class defined as follows:
      - $ICEXT(c_{\geq\alpha}) \subseteq ICEXT(c)$
      - $ICEXT(c_{\geq\alpha}) = \{\ a \in ICEXT(c) |\ \mu_{ICEXT(c)}(a) \geq \alpha\ \}$;

   (b) we equally add a Datalog fact of the form `c`$_{\geq\alpha}$(`a`).

2. For each fuzzy RDFS statement α:<`c`,`rdfs:subClassOf`,`d`>, we define a datalog fact of the form:

   `subClassOf`(c$_{>1-\alpha}$,d$_{\geq\alpha}$)

   where c$_{>1-\alpha}$ and d$_{>\alpha}$ are new exact classes defined as follows:
   - $ICEXT(c_{>1-\alpha}) = \{\ x \in ICEXT(c) |\ \mu_{ICEXT(c)}(x) > 1-\alpha\ \}$.
   - $ICEXT(d_{\geq\alpha}) = \{\ x \in ICEXT(d) |\ \mu_{ICEXT(d)}(x) \geq \alpha\ \}$.

3. For each fuzzy RDFS statement α:<`p`,`rdfs:subPropertyOf`,`q`>, we define a datalog fact of the form:

   `subPropertyOf`(p$_{>1-\alpha}$,q$_{\geq\alpha}$)

where $p_{>1-\alpha}$ and $q_{\geq\alpha}$ are new crisp properties defined as follows:
- $IEXT(p_{>1-\alpha}) = \{\ (x,y) \in IEXT(p)|\ \mu_{IEXT(p)}(x,y) > 1-\alpha\ \}$.
- $IEXT(p_{\geq\alpha}) = \{(x,y) \in IEXT(p)|\ \mu_{IEXT(p)}(x,y) \geq \alpha\ \}$.

4. For each fuzzy RDFS statement α:<a,onto:p,c>, where onto:p is a user defined property, two cases may occur:

   (a) The RDFS property is used to define relation between resources. In this case, the fuzzy RDFS statement α:<a,onto:p,c> is transformed into a Datalog fact as follows:

   $$\mathtt{p}_{\geq\alpha}(\mathrm{a,c})$$

   where $\mathtt{p}_{\geq\alpha}$ is a new crisp property defined as follows:

   $$IEXT(\mathtt{p}_{\geq\alpha}) = \{\ (x,y) \in IEXT(\mathrm{p})|\ \mu_{IEXT(p)}(x,y) \geq \alpha\ \}$$

   For example, the fuzzy RDFS statement 0.6:<Chile,isNear,Brazil> is defined with Datalog as follows:

   $$\mathtt{isNear}_{\geq 0.6}(\mathrm{Chile,Brazil})$$

   (b) The fuzzy RDFS property is used to define the value of a resource. A fuzzy RDFS statement α:<a,onto:p,c>, where c is a fuzzy value is transformed into a Datalog fact as follows:

   $$\mathtt{p}(\mathtt{a},\mathtt{c}_{\geq\alpha})$$

   where $\mathtt{c}_{\geq\alpha}$ is a new crisp class defined as follows:

   $$ICEXT(\mathtt{c}_{\geq\alpha}) = \{\ x \in ICEXT(\mathrm{c})|\ \mu_{ICEXT(c)}(x) \geq \alpha\ \}$$

   For example, the fuzzy RDFS statement 0.6:<Brazil,hasPopulation, Large> is defined with Datalog as follows:

   $$\mathtt{hasPopulation}(\mathtt{Brazil},\mathtt{Large}_{\geq 0.6})$$

   The class $\mathtt{Large}_{\geq 0.6}$ is defined as follows:

   $$\mathtt{Large}_{\geq 0.6}(\mathrm{x}) \leftarrow \mathtt{Large}(\mathrm{x}),\ \mathtt{hasDomLarge}(\mathrm{x,v}),\ \mathrm{v}{>=}0.6$$

   where, hasDomLarge(x,v) is a membership function defined for the linguistic label *Large*. If we consider that *Large* is defined with a trapezoidal membership function with four parameters (50 000 000,150 000 000,300 000 000,400 000 000), we may define the membership function hasDomLarge(x,v) with *built-in* predicates as follows:

$$\begin{array}{l}
\mathrm{hasDomLarge(x,0)} \leftarrow x <= 50000000 \\
\mathrm{hasDomLarge(x,0)} \leftarrow x >= 400000000 \\
\mathrm{hasDomLarge(x,1)} \leftarrow x > 150000000, x < 300000000 \\
\mathrm{hasDomLarge(x,v)} \leftarrow x > 50000000, x < 150000000, v = (x - 50000000)/100000000 \\
\mathrm{hasDomLarge(x,v)} \leftarrow x > 300000000, x < 400000000, v = (400000000 - x)/100000000
\end{array}$$

5. For each class c, we define a Datalog fact of the form `class`(c).
6. For each property p, we define a Datalog fact of the form `property`(p).
7. Add a Datalog fact $O(x)$ for each resource x in IR.
8. For each fuzzy class c, fuzzy property p and for each $\alpha \in]0, 1]$ and $\beta \in [0, 1]$ defined in $\mathcal{N}$ where $\alpha > \beta$, the following facts are defined in P_{rdfs}:

$$\begin{array}{ll} \texttt{subClassOf}(c_{\geq\alpha}, c_{>\beta}) & \texttt{subClassOf}(c_{>\gamma}, c_{\geq\gamma}) \\ \texttt{subPropertyOf}(p_{\geq\alpha}, p_{>\beta}) & \texttt{subPropertyOf}(p_{>\gamma}, p_{\geq\gamma}) \end{array}$$

RDFS inference in Datalog is equally based on rules. As we have transformed fuzzy classes and properties into crisp ones, crisp RDFS rules may be used for reasoning. We present in table 2, the rewriting of crisp RDFS inference rules with Datalog.

Table 2. Rewriting RDFS inference rules with Datalog based on the defuzzification approach

Rule	Body	Head
1	aaa(xxx,yyy)	rdf:type(aaa,rdf:Property)
2	rdfs:domain(aaa,zzz), aaa(xxx,zzz)	rdf:type(aaa,zzz)
3	rdfs:range(xxx,zzz), aaa(xxx,uuu)	rdf:type(xxx,zzz)
4a	aaa(uuu,xxx)	rdf:type(uuu,rdfs:Resource)
4b	aaa(uuu,xxx)	rdf:type(uuu,rdfs:Resource)
5	rdfs:subPropertyOf(uuu,vvv), rdfs:subPropertyOf(vvv,zzz)	rdfs:subPropertyOf(uuu,zzz)
6	rdf:type(xxx,rdf:Property)	rdfs:subPropertyOf(xxx,xxx)
7	rdfs:subPropertyOf(aaa,bbb), aaa(uuu,yyy)	bbb(uuu,yyy)
8	rdf:type(uuu,rdfs:Class)	rdfs:subClassOf(uuu,Resrouce)
9	rdfs:subClassOf(uuu,xxx), rdf:type(vvv,uuu),	rdf:type(vvv,xxx)
10	rdf:type(uuu,rdf:Class)	rdfs:subClassOf(uuu,uuu)
11	rdfs:subClassOf(uuu,vvv), rdfs:subClassOf(vvv,xxx)	rdfs:subClassOf(uuu,xxx)
12	rdf:type(uuu,rdfs:Datatype)	rdfs:subClassOf(uuu,rdfs:Literal)

Given a fuzzy RDFS interpretation I of fKB_{rdfs}, a Herbrand interpretation $I_P = (D, \phi, \sigma)$ of P_{rdfs} is defined as follows:

1. D is equal to IR,
2. ϕ is a mapping which assigns each constant to itself,
3. σ is a mapping which assigns each predicate p of arity n to a membership function $\sigma(p) : D^n \rightarrow \{true, false\}$,
4. $c^{I_P}_{\geq\alpha} = \{x \in IR \mid \mu_{ICEXT(\mathtt{c}^I)}(x) \geq \alpha\}$,
5. $p^{I_P}_{\geq\alpha} = \{(x, y) \in IR \times IR \mid \mu_{IEXT(\mathtt{p}^I)}(x, y) \geq \alpha\}$,
6. $c^{I_P}_{>\alpha} = \{x \in IR \mid \mu_{ICEXT(\mathtt{c}^I)}(x) > \alpha\}$,
7. $p^{I_P}_{>\alpha} = \{(x, y) \in IR \times IR \mid \mu_{IEXT(\mathtt{p}^I)}(x, y) > \alpha\}$.

The interpretation I_P of P_{rdfs} has to satisfy the following conditions:

- $\sigma(p_{>\alpha})(s^{I_P}, o^{I_P})$ is true iff $\mu_{IEXT(p^I)}(s^I, o^I) > \alpha$,
- $\sigma(p_{\geq\alpha})(s^{I_P}, o^{I_P})$ is true iff $\mu_{IEXT(p^I)}(s^I, o^I) \geq \alpha$,
- $\sigma(\mathtt{class})(c^{I_P}_{>\alpha})$ is true iff $c \in IC$,
- $\sigma(\mathtt{class})(c^{I_P}_{\geq\alpha})$ is true iff $c \in IC$,
- $\sigma(\mathtt{property})(p^{I_P}_{>\alpha})$ is true iff $p \in IP$,
- $\sigma(\mathtt{property})(p^{I_P}_{\geq\alpha})$ is true iff $p \in IP$,
- $\sigma(\mathtt{range})(p^{I_P}, c^{I_P}_{>\alpha})$ is true iff $(p, c) \in IEXT(\mathtt{range}^I)$,
- $\sigma(\mathtt{range})(p^{I_P}, c^{I_P}_{\geq\alpha})$ is true iff $(p, c) \in IEXT(\mathtt{range}^I)$,
- $\sigma(\mathtt{domain})(p^{I_P}, c^{I_P}_{>\alpha})$ is true iff $(p, c) \in IEXT(\mathtt{domain}^I)$,
- $\sigma(\mathtt{domain})(p^{I_P}, c^{I_P}_{\geq\alpha})$ is true iff $(p, c) \in IEXT(\mathtt{domain}^I)$,
- $\sigma(\mathtt{type})(x, c^{I_P}_{>\alpha})$ is true iff $x \in IR$, $c \in IC$ and $\mu_{ICEXT(\mathtt{c}^I)}(x) > \alpha$,
- $\sigma(\mathtt{type})(x, c^{I_P}_{\geq\alpha})$ is true iff $x \in IR$, $c \in IC$ and $\mu_{ICEXT(\mathtt{c}^I)}(x) \geq \alpha$,
- $\sigma(\mathtt{subClassOf})(c^{I_p}_{>1-\alpha}, d^{I_p}_{\geq\alpha})$ is true iff c, $d \in IC$ and
$$\min(\mu_{ICEXT(c^I)}(x), \alpha) \leq \mu_{ICEXT(d^I)}(x) \ \forall x \in IR,$$
- $\sigma(\mathtt{subPropertyOf})(p^{I_p}_{>1-\alpha}, q^{I_p}_{\geq\alpha})$ is true iff p, $q \in IP$ and
$$\min(\mu_{IEXT(p^I)}(x, y), \alpha) \leq \mu_{IEXT(q^I)}(x, y) \ \forall x, y \in IR.$$

An interpretation I_P is a model of P_{rdfs} if it satisfies the following conditions:

1. If $\sigma(\mathtt{range})(p^{I_P}, c^{I_P}_{>\alpha})$ is true and $\sigma(p_{>\alpha})(x^{I_P}, y^{I_P})$ is true then $\sigma(c^{I_P}_{>\alpha})(y)$ is true,
2. If $\sigma(\mathtt{range})(p^{I_P}, c^{I_P}_{\geq\alpha})$ is true and $\sigma(p_{\geq\alpha})(x^{I_P}, y^{I_P})$ is true then $\sigma(c^{I_P}_{\geq\alpha})(y)$ is true,
3. If $\sigma(\mathtt{domain})(p^{I_P}, c^{I_P}_{>\alpha})$ is true and $\sigma(p_{>\alpha})(x^{I_P}, y^{I_P})$ is true then $\sigma(c^{I_P}_{>\alpha})(y)$ is true,
4. If $\sigma(\mathtt{domain})(p^{I_P}, c^{I_P}_{\geq\alpha})$ is true and $\sigma(p_{\geq\alpha})(x^{I_P}, y^{I_P})$ is true then $\sigma(c^{I_P}_{\geq\alpha})(y)$ is true,
5. If $\sigma(c^{I_P}_{>\alpha})(x^{I_P})$ is true then $\sigma(\mathtt{type})(x^{I_P}, c^{I_P}_{>\alpha})$ is true,
6. If $\sigma(c^{I_P}_{\geq\alpha})(x^{I_P})$ is true then $\sigma(\mathtt{type})(x^{I_P}, c^{I_P}_{\geq\alpha})$ is true,
7. If $\sigma(\mathtt{subClassOf})(c^{I_P}_{>\alpha}, d^{I_P}_{\geq\beta})$ is true then $\sigma(d^{I_P}_{\geq\beta})(x) \leftarrow \sigma(c^{I_P}_{>\alpha})(x)$ is true,
8. If $\sigma(\mathtt{subPropertyOf})(p^{I_P}_{>\alpha}, q^{I_P}_{\geq\beta})$ is true then $\sigma(q^{I_P}_{\geq\beta})(x, y) \leftarrow \sigma(p^{I_P}_{>\alpha})(x, y)$ is true.

Theorem 1. *The transformation of a fuzzy knowledge base fKB$_{rdfs}$ into a crisp Datalog program P_{rdfs} based on the defuzzifcation approach is correct.*

Proof. We need to show here that the transformation maintains the transitivity of RDFS properties subClassOf and subPropertyOf on which the entailment of RDFS is based. We show here that the transitivity of the RDFS subClassOf is maintained, the case of the subPropertyOf property follows the same principle. We consider a fuzzy RDFS graph G with the following fuzzy RDFS statements:

α:<c,rdfs:subClassOf,d>
β:<d,rdfs:subClassOf,e>

We have $G \models \gamma{:}{<}\texttt{c,rdfs:subClassOf,e}{>}$ where $\gamma = \min(\alpha, \beta)$ iff $\alpha + \beta - 1 \geq 0$. These fuzzy RDFS statement are transformed into Datalog facts as follows:

$$\alpha{:}{<}\texttt{c,rdfs:subClassOf,d}{>} \Rightarrow \texttt{subClassOf}(\mathrm{c}_{>1-\alpha},\mathrm{d}_{\geq\alpha})$$
$$\beta{:}{<}\texttt{d,rdfs:subClassOf,e}{>} \Rightarrow \texttt{subClassOf}(\mathrm{d}_{>1-\beta},\mathrm{e}_{\geq\beta})$$
$$\gamma{:}{<}\texttt{c,rdfs:subClassOf,e}{>} \Rightarrow \texttt{subClassOf}(\mathrm{c}_{>1-\gamma},\mathrm{e}_{\geq\gamma})$$

We consider a Datalog program P with the following facts:

$$\texttt{subClassOf}(\mathrm{c}_{>1-\alpha},\mathrm{d}_{\geq\alpha}) \text{ and } \texttt{subClassOf}(\mathrm{d}_{>1-\beta},\mathrm{e}_{\geq\beta})$$

we need to show that if $\alpha + \beta - 1 \geq 0$ then $P \models \texttt{subClassOf}(\mathrm{c}_{>1-\gamma},\mathrm{e}_{\geq\gamma})$ where $\gamma = \min(\alpha, \beta)$. The following Datalog rule results from the transformation of inference rules of the transitive closure of `subClassOf`.

$$\text{subClassOf(x,y)} \leftarrow \text{subClassOf(x,z), subClassOf(z,y)}$$

We have:

- $\alpha \geq min(\alpha, \beta)$: $\texttt{subClassOf}(\mathrm{c}_{>1-\min(\alpha,\beta)},\mathrm{c}_{>1-\alpha})$ is defined in the Datalog program P, we deduce $\texttt{subClassOf}(\mathrm{c}_{>1-\min(\alpha,\beta)},\mathrm{d}_{\geq\alpha})$,
- $\beta \geq min(\alpha, \beta)$: $\texttt{subClassOf}(\mathrm{e}_{\geq\beta},\mathrm{e}\geq \min(\alpha, \beta))$ is defined in the Datalog program P, we deduce $\texttt{subClassOf}(\mathrm{d}_{>1-\beta},\mathrm{e}_{\geq\min(\alpha,\beta)})$,
- $\alpha + \beta - 1 \geq 0$ ($\alpha \geq 1 - \beta$): $\texttt{subClassOf}(\mathrm{d}_{\geq\alpha},\mathrm{d}> 1 - \beta)$ is defined in the Datalog program P.

We obtain:

- $\texttt{subClassOf}(\mathrm{c}_{>1-\min(\alpha,\beta)},\mathrm{d}_{\geq\alpha})$,
- $\texttt{subClassOf}(\mathrm{d}_{\geq\alpha},\mathrm{d}_{>1-\beta})$,
- $\texttt{subClassOf}(\mathrm{d}_{>1-\beta},\mathrm{e}_{\geq\min(\alpha,\beta)})$.

We induce $\texttt{subClassOf}(\mathrm{c}_{>1-\min(\alpha,\beta)},\mathrm{e}_{\geq\min(\alpha,\beta)})$ which corresponds to the result of transformation of the fuzzy RDFS statement $\min(\alpha, \beta){:}{<}\texttt{c,rdfs:subClassOf,e}{>}$.

Complexity of the Defuzzification Approach: We recall that based on the principle of the defuzzification approach, the transformation of a given fuzzy RDFS knowledge base fKB$_{rdfs}$ with n classes (resp. properties) and a set $\mathcal{N}$ of membership degrees into a Datalog program P$_{rdfs}$, we need to define $2 * n * |\mathcal{N}|$ classes (resp. properties). Equally, for each fuzzy class a (resp. property p) and for each $\alpha \in]0, 1]$ and $\beta \in [0, 1]$ defined in $\mathcal{N}$, we need to add new terminological RDFS statements of the form:

$$\texttt{subClassOf}(a_{\geq\alpha}, a_{>\beta}),\ \texttt{subPropertyOf}(p_{\geq\alpha}, p_{>\beta})$$

and for each $\gamma \in \mathcal{N}$, we need to add new terminological RDFS statements of the form:

$$\texttt{subClassOf}(a_{>\gamma}, a_{\geq\gamma}),\ \texttt{subPropertyOf}(p_{>\gamma}, p_{>\gamma})$$

We obtain that $|\mathrm{P}_{rdfs}| = 2^{*}\ |\mathcal{N}|\ *\ |\mathrm{fKB}_{rdfs}|$. We deduce that the defuzzification approach is quadratic $\mathrm{O}(|\mathrm{fKB}_{rdfs}|^2)$.

6.2 The Skolemisation Approach

We present in this section the principle of the second approach of rewriting of fuzzy RDFS with crisp Datalog programs. The principle of skolemisation approach consists on representing fuzzy RDFS statements with Datalog predicates having the same names. The membership degrees are defined as terms of theses predicates. We present in the following a list of mapping rules used to transform fuzzy RDFS statements into crisp Datalog predicates with the skolemisation approach:

1. For each fuzzy RDFS statement α:<a,rdf:type,c> :

   (a) we define a Datalog fact of the form type(a,c,α).
   (b) we add a Datalog fact of the form c(a,α).

2. For each fuzzy RDFS statement α:<c,rdfs:subClassOf,d>, we define a Datalog fact of the form:

   subClassOf(c,d,α)

3. For each fuzzy RDFS statement α:<p,rdfs:subPropertyOf,q>, we define a Datalog fact of the form:

   subPropertyOf(p,q,α)

4. For each fuzzy RDFS statement α:<a,onto:p,c>, where onto:p is a user defined property, we define a Datalog fact of the form:

   p(a,c,α)

   For example, the fuzzy RDFS statement 0.6:<Chile,isNear,Brazil> used to define the relation between two resources is written with Datalog as follows:

   isNear(Chile,Brazil,0.6)

   The fuzzy RDFS statement 0.6:<Brazil,hasPopulation,Large> used to define the value of a resource is written with Datalog as follows:

   hasPopulation(Brazil,Large,0.6)

   where:

   hasPopulation(x,Large,0.6) ← hasDomLarge(x,v), v>=0.6

   hasDomLarge(x,v) is a membership function defined for the linguistic label *Large*. We note that the definition of membership functions follows the same principle of the defuzzification approach.
5. For each class c, we define a Datalog fact of the form class(c).
6. For each property p, we define a Datalog fact of the form property(p).
7. Add a fact $O(x)$ for each resource x in IR.

Table 3. Rewriting fuzzy RDFS inference rules with Datalog based on the skolemisation approach

Rule	Body	Head
1	aaa(xxx,yyy,α)	rdf:type(aaa,rdf:Property,1)
2	rdfs:domain(aaa,zzz,α), aaa(xxx,zzz,β)	rdf:type(aaa,zzz,1)
3	rdfs:range(xxx,zzz,α), aaa(xxx,uuu,β)	rdf:type(xxx,zzz,1)
4a	aaa(uuu,xxx,α)	rdf:type(uuu,rdfs:Resource,1)
4b	aaa(uuu,vvv,α)	rdf:type(uuu,rdfs:Resource,1)
5a	rdfs:subPropertyOf(uuu,vvv,α), rdfs:subPropertyOf(vvv,zzz,β), $\alpha >= \beta$	rdfs:subPropertyOf(uuu,zzz,β)
5b	rdfs:subPropertyOf(uuu,vvv,α), rdfs:subPropertyOf(vvv,zzz,β), $\alpha < \beta$	rdfs:subPropertyOf(uuu,zzz,α)
6	rdf:type(xxx,rdf:Property,α)	rdfs:subPropertyOf(xxx,xxx,1)
7a	rdfs:subPropertyOf(aaa,bbb,α), aaa(uuu,yyy,β), $\alpha >= \beta$	bbb(uuu,yyy,β)
7b	rdfs:subPropertyOf(aaa,bbb,α), aaa(uuu,yyy,β), $\alpha < \beta$	bbb(uuu,yyy,α)
8	rdf:type(uuu,rdfs:Class,α)	rdfs:subClassOf(uuu,Resource,1)
9a	rdfs:subClassOf(uuu,xxx,α), rdf:type(vvv,uuu,β), $\alpha >= \beta$	rdf:type(vvv,xxx,β)
9b	rdfs:subClassOf(uuu,xxx,α), rdf:type(vvv,uuu,β), $\alpha < \beta$	rdf:type(vvv,xxx,α)
10	rdf:type(uuu,rdf:Class,α)	rdfs:subClassOf(uuu,uuu,1)
11a	rdfs:subClassOf(uuu,vvv,α), rdfs:subClassOf(vvv,xxx), $\alpha >= \beta$	rdfs:subClassOf(uuu,xxx,β)
11b	rdfs:subClassOf(uuu,vvv,α), rdfs:subClassOf(vvv,xxx), $\alpha < \beta$	rdfs:subClassOf(uuu,xxx,α)
12	rdf:type(uuu,rdfs:Datatype,α)	rdfs:subClassOf(uuu,rdfs:Literal,1)

The table 3 presents the principle of rewriting fuzzy inference rules with Datalog based on the skolemisation approach.

Given a fuzzy RDFS interpretation I of fKB$_{rdfs}$, a Herbrand interpretation $I_P = (D, \phi, \sigma)$ of P$_{rdfs}$ is defined as follows:

1. D is equal to IR,
2. ϕ is a mapping which assigns each constant to itself,
3. σ is a mapping which assigns each predicate p of arity n to a membership function $\sigma(p) : D^n \to \{true, false\}$,
4. $c^{I_P} = ICEXT(\mathtt{c}^I)$,
5. $p^{I_P} = IEXT(\mathtt{p}^I)$.

The interpretation I_P of P$_{rdfs}$ has to satisfy the following conditions:

- $\sigma(p)(s^{I_P}, o^{I_P}, \alpha)$ is true iff $\mu_{IEXT(p^I)}(s^I, o^I) \geq \alpha$,
- $\sigma(\mathtt{class})(c^{I_P})$ is true iff $c \in IC$,
- $\sigma(\mathtt{property})(p^{I_P})$ is true iff $p \in IP$,
- $\sigma(\mathtt{range})(p^{I_P}, c^{I_P}, \alpha)$ is true iff $\mu_{IEXT(\mathtt{range}^I)}(p^I, c^I) \geq \alpha$,
- $\sigma(\mathtt{domain})(p^{I_P}, c^{I_P}, \alpha)$ is true iff $\mu_{IEXT(\mathtt{domain}^I)}(p^I, c^I) \geq \alpha$,

- $\sigma(\mathtt{type})(x, c^{I_P}, \alpha)$ is true iff $x \in IR$, $c \in IC$ and $\mu_{ICEXT(\mathtt{c}^I)}(x) \geq \alpha$,
- $\sigma(\mathtt{subClassOf})(c^{I_p}, d^{I_p}, \alpha)$ is true iff c, $d \in IC$ and

$$\min(\mu_{ICEXT(c^I)}(x), \alpha) \leq \mu_{ICEXT(d^I)}(x) \ \forall x \in IR,$$

- $\sigma(\mathtt{subPropertyOf})(p^{I_p}, q^{I_p}, \alpha)$ is true iff p, $q \in IP$ and

$$\min(\mu_{IEXT(p^I)}(x, y), \alpha) \leq \mu_{IEXT(q^I)}(x, y) \ \forall x, y \in IR,$$

An interpretation I_P is a model of P_{rdfs} if it satisfies the following conditions:

1. If $\sigma(\mathtt{range})(p^{I_P}$,$c^{I_P}, \alpha)$ is true and $\sigma(p)(x^{I_P}$,y^{I_P},$\alpha)$ is true then $\sigma(c^{I_P})(y, \alpha)$ is true,
2. If $\sigma(\mathtt{domain})(p^{I_P}$,$c^{I_P}, \alpha)$ is true and $\sigma(p)(x^{I_P}$,y^{I_P},$\alpha)$ is true then $\sigma(c^{I_P})(y, \alpha)$ is true,
3. If $\sigma(c^{I_P})(x^{I_P}$,$\alpha)$ is true then $\sigma(\mathtt{type}^{I_P})(x^{I_P}$,$c^{I_P}$,$\alpha)$ is true,
4. If $\sigma(\mathtt{subClassOf})(c^{I_P}$,$d^{I_P}$,$\alpha)$ is true and $\min(\gamma, \alpha) \leq \beta$ is true then

$$\sigma(d^{I_P})(x, \beta) \leftarrow \sigma(c^{I_P})(x, \gamma) \text{ is true}$$

5. If $\sigma(\mathtt{subPropertyOf})(p^{I_P}$,$q^{I_P}$,$\alpha)$ is true and $\min(\gamma, \alpha) \leq \beta$ then

$$\sigma(q^{I_P}(x, y, \beta) \leftarrow \sigma(p^{I_P})(x, y, \gamma) \text{ is true}$$

Theorem 2. *The transformation of a fuzzy knowledge base fKB_{rdfs} into a crisp Datalog program P_{rdfs} based on the skolemisation approach is correct.*

Proof. As in the case of the defuzzification approach, we need to show here that the transformation maintains the transitivity of RDFS properties subClassOf and subPropertyOf on which the entailment of RDFS is based. We show here that the transitivity of the RDFS subClassOf is maintained, the case of the subPropertyOf property follows the same principle. We consider a fuzzy RDFS graph G with the following fuzzy RDFS statements:

α:<c,rdfs:subClassOf,d>
β:<d,rdfs:subClassOf,e>

We have $G \models \gamma$:<c,rdfs:subClassOf,e> where $\gamma = \min(\alpha, \beta)$. These fuzzy RDFS statement are transformed into Datalog facts as follows:

α:<c,rdfs:subClassOf,d> ⇒ subClassOf(c,d,α)
β:<d,rdfs:subClassOf,e> ⇒ subClassOf(d,e,β)

We consider a Datalog program P with the following facts:

subClassOf(c,d,α) and subClassOf(d,e,β)

we need to show that $P \models$ subClassOf(c,e,γ) where $\gamma = \min(\alpha, \beta)$. The following Datalog rules result from the transformation of inference rules of the transitive closure of subClassOf:

$$\text{subClassOf(x,y,m)} \leftarrow \text{subClassOf(x,z,n), subClassOf(z,y,m)}, n \geq m$$
$$\text{subClassOf(x,y,n)} \leftarrow \text{subClassOf(x,z,n), subClassOf(z,y,m)}, n < m$$

Based on theses rules, we deduce that $P \models$ `subClassOf`(c,e,γ) where $\gamma = \min(\alpha, \beta)$.

Complexity of the Skolemisation Approach: Based on the principle of the skolemisation approach, fuzzy knowledge base fKB$_{rdfs}$ and the crisp Datalog program P$_{rdfs}$ obtained as a result of the transformation have the same size as we do not need to add new classes or statements as in the defuzzification approach. That is, we may deduce that the skolemisation approach has a linear complexity O(|fKB$_{rdfs}$|). A particular attention should be allowed on the Datalog rules in table 3 which use an inequality between two variables. The definition of such a kind of rules may not be allowed on Datalog systems like $\mathcal{F}$lora-2 and XSB which we use on the experimentations. What we need to do is to define two different rules for each membership degree used in the fuzzy knowledge base. The concerned rules are the ones which contain an inequality between two variables α and β (see section 8.

7 Datalog Based FSAQL Query Processing

We propose in this section the principle of rewriting of FSAQL queries with crisp Datalog. We denote with τ a mapping function of an FSAQL query to a Datalog one. Given an FSAQL query *Select*(W,P), the semantic of τ may be defined as follows:

1. $\tau(W, P) = Answer(W) \leftarrow \Pi_D^{rdfs}(P)$. With Π_D^{rdfs} a function that allow the transformation of a fuzzy RDFS statement into a Datalog fact,
2. $\tau(W, P_i$ AND $P_j) = Answer(W) \leftarrow Answer(W_i), Answer(W_j)$. With W_i and W_j the variables defined respectively in P_i and P_j,
3. $\tau(W, P_i$ UNION $P_j) = Answer(W) \leftarrow Answer(W_i)$,
   $Answer(W) \leftarrow Answer(W_j)$
4. $\tau(W, P$ Have Values $R) = Answer(W) \leftarrow \Pi_D^{rdfs}(P), R$.

We propose in the following an algorithm to transform an FSAQL query into a Datalog query:

1. Replace the variables defined in the **WHERE** clause based on the conditions defined in the **HAVE VALUES** clause:
   (a) Equality condition ($?X = v$): replace the variable $?X$ with the value v in fuzzy RDFS statements defined in the **WHERE** clause.
   (b) Equality or inequality condition about the degree of truth associated to a fuzzy RDFS statement ($n = v$ or $n \geq v$): replace the degree n with the value v.
2. Transform the fuzzy RDFS statement defined in the **WHERE** clause into crisp Datalog facts based on the defuzzification or skolemisation approach. The conjunction of the obtained fact constitutes the body of the Datalog query.

3. Define in the Head of the query the predicate *Answer*. The terms of the *Answer* predicate are those variables defined in the **SELECT** clause.
4. Add the condition defined in the **HAVE VALUES** to the body of the Datalog query.

Example 1. Given the following FSAQL query:

```
SELECT       ?X
FROM         ⟨Fuzzy RDFS repository⟩
WHERE        n:⟨?X rdf:type GreatCountry⟩
             m:⟨?X rdf:hasPopulation ?Y⟩
HAVE VALUES  n=0.8, m=0.7, ?Y=Large
```

The transformation of this FSAQL query into a Datalog one is realized as follows:

1. Replace the variables in the **WHERE** clause with their values defined in the **HAVE VALUES** clause:

   ```
   0.8:⟨?X rdf:type GreatCountry⟩
   0.7:⟨?X rdf:hasPopulation Large⟩
   ```

2. Transform RDFS statements in the **WHERE** clause into Datalog facts:

   – Defuzzification approach:

   $$\texttt{type(?X,GreatCountry}_{\geq 0.8}\texttt{)}$$
   $$\texttt{hasPopulation(?X,Large}_{\geq 0.7}\texttt{)}$$

   – Skolemisation appraoch:

   ```
   type(?X,GreatCountry,0.8)
   hasPopulation(?X,Large,0.7)
   ```

3. We obtain the following Datalog query:

   – Defuzzification approach:

   $$\text{Answer(?X)} \leftarrow \texttt{type(?X,GreatCountry}_{\geq 0.8}\texttt{)},\ \texttt{hasPopulation(?X,Numerous}_{\geq 0.7}\texttt{)}$$

   – Skolemisation approach:

   $$\text{Answer(?X)} \leftarrow \texttt{type(?X,GreatCountry,0.8)},\ \texttt{hasPopulation(?X,Numerous,0.7)}$$

Theorem 3. *Every response Answer*(a) *of* $\tau(W, P)$ *is a solution of the FSAQL query* $[[P]]_G^{rdfs}$ *(correction) and every responses of* $[[P]]_G^{rdfs}$ *are included into Answer*(W) *(completeness).*

Proof. We note that the proof is proposed for the skolemisation approach, the proof of the defuzzification approach follows the same principle.

***Correction*:** We need to show that if $\Pi_D^{rdfs}(P)$ is true for a given set of individuals $\{a_i|$i:1...n-1,$\alpha \in$ [0,1]$\}$ where n is the number of variables W then P satisfies the a_i with a degree $\geq \alpha$. Tow cases may occurs:

1. $\Pi_D^{rdfs}(P)$ = `type`(?x,c,?n) (written equally as `c`(?x,?n)): we need to show that if $Answer(a, \alpha)$ is a solution of the Datalog query $Answer(W) \leftarrow$ `type`$(?x, c, ?n)$ then (a,α) is a solution of the FSAQL query $[[?n :<?x, rdf : type, c>]]_G^{rdfs}$ which is true if $\mu_{ICEXT(I(c))}(a) \geq \alpha$ is verified,
2. $\Pi_D^{rdfs}(P)$ = `p`(a,b,α): we need to show that if $Answer(a, b, \alpha)$ is a solution of the Datalog query $\tau(W, P)$ then (a, b, α) is a solution of the FSAQL query $[[?n :<?x, rdf : p, ?y>]]_G^{rdfs}$ which is true if $\mu_{IPEXT(I(p))}(a, b) \geq \alpha$ is verified.

Case 1: $\Pi_D^{rdfs}(P)$ = **`type`(?x,c,?n)**. We suppose that $Answer(a, \alpha)$ is a solution of the Datalog query $Answer(W) \leftarrow$ `type`$(?x, c, ?n)$. This may occur in two cases:

- The fact `type`(a,c,α) (or `c`(a,α)) is explicitly defined in the Datalog knowledge base P$_{rdfs}$. Based on the Herbrand interpretation $I_P = (D, \phi, \sigma)$ of P$_{rdfs}$ (see section 6), $\mu_{ICEXT(I(c))}(a) \geq \alpha$. We deduce then that ($a$,$\alpha$) is a solution of the FSAQL query $[[?n :<?x, rdf : type, c>]]_G^{rdfs}$.
- The fact `type`(a,c,α) (or `c`(a,α)) is deduced based on the inference rules 1, 2, 3, 9a or 9b defined in table 3. We show the case where inference rules 9a or 9b are invoked, the other rules follow the same principle. We suppose that `type`(a,c,α) is induced based on the inference rules 9a or 9b. That is, there exist a class d, a degree of membership β such that `type`(a,d,α) and `subClassOf`(d,c,β) (or `type`(a,d,β) and `subClassOf`(d,c,α)) such that $\alpha = \min(\alpha, \beta)$. `type`(a,d,$\alpha$) induces that $\mu_{ICEXT(I(d))}(a) \geq \alpha$ and `subClassOf`(d,c,β) induces that $\min(\mu_{ICEXT(I(d))}(x), \beta) \leq \mu_{ICEXT(I(c))}(x)$ $\forall x \in IR$. We obtain $\min(\mu_{ICEXT(I(d))}(a), \beta) \leq \mu_{ICEXT(I(c))}(a)$. Based on the fact `type`(a,d,α), we deduce that $\min(\mu_{ICEXT(I(d))}(a), \beta) = \alpha$ and $\alpha \leq \mu_{ICEXT(I(c))}(a)$. We conclude that ($a$,$\alpha$) is then a solution of the FSAQL query $[[?n :<?x, rdf : type, c>]]_G^{rdfs}$.

Case 2: $\Pi_D^{rdfs}(P)$ = **`p`(a,b,α)**. We suppose that $Answer(a, b, \alpha)$ is a solution of the Datalog query $Answer(W) \leftarrow$ `p`$(?x, ?y, ?n)$. This may occur in two cases:

- The fact `p`(a,b,α) is explicitly defined in the Datalog knowledge base P$_{rdfs}$. Based on the Herbrand interpretation $I_P = (D, \phi, \sigma)$ of P$_{rdfs}$ (see section 6), $\mu_{IPEXT(I(p))}(a, b) \geq \alpha$. We deduce then that ($a$,$b$,$\alpha$) is a solution of the FSAQL query $[[?n :<?x, rdf : p, c>]]_G^{rdfs}$.
- The fact `p`(a,b,α) is deduced based on the inference rules (table 6) 5a, 5b or 6 if p corresponds to the rdfs property `subPropertyOf`; 8, 10, 11a, 11b or 12 if p

corresponds to the rdfs property `subClassOf` and 7a or 7b if p corresponds to any type of property. We show the case where rules 7a or 7b are invoked, the other rules follow the same principle. We suppose that p(a,b,α) is induced based on the inference rules 7a or 7b. That is, there exist a property q, a degree of membership β such that q(a,b,α) and `subPropertyOf`(q,p,β) (or q(a,b,β) and `subPropertyOf`(q,p,α)) such that $\alpha = \min(\alpha, \beta)$. q(a,b,$\alpha$) induces that $\mu_{IPEXT(I(q))}(a, b) \geq \alpha$ and `subPropertyOf`(q,p,β) induces that $\min(\mu_{IPEXT(I(q))}(x, y), \beta) \leq \mu_{IPEXT(I(p))}(x, y)$ $\forall x, y \in IR$. We obtain that $\min(\mu_{IPEXT(I(q))}(a, b), \beta) \leq \mu_{IPEXT(I(p))}(a, b)$. Based on the fact q(a,b,α), we deduce that $\min(\mu_{IPEXT(I(q))}(a, b), \beta) = \alpha$ and $\alpha \leq \mu_{IPEXT(I(p))}(a, b)$. We conclude that ($a$,$b$,$\alpha$) is then a solution of the FSAQL query $[[?n :< ?x, rdf : p, ?y >]]_G^{rdfs}$.

Completeness: We need to show that if a set of individuals $\{a_i$|i:1...n-1$\}$ is a solution of an FSAQL query $[[P]]_G^{rdfs}$ with a degree greater or equal than a given degree $\alpha \in]0, 1]$ then $\Pi_D^{rdfs}(P)$ is true for the set of individuals $\{a_i$|i:1...n-1,$\alpha \in$ [0,1]$\}$.Tow cases may occurs:

1. $[[P]]_G^{rdfs} = [[?n :< ?x, rdf : type, c >]]_G^{rdfs}$: we need to show that if (a,α) is a solution of the FSAQL query $[[P]]_G^{rdfs}$, then $Answer(a, \alpha)$ is a solution of the Datalog query $Answer(W) \leftarrow \mathtt{type}(?x, c, ?n)$ (or $Answer(W) \leftarrow \mathtt{c}(?x, ?n)$).
2. $[[P]]_G^{rdfs} = [[?n :< ?x, rdf : p, ?y >]]_G^{rdfs}$: we need to show that if (a,α) is a solution for the FSAQL query $[[P]]_G^{rdfs}$, then $Answer(a, b, \alpha)$ is a solution of the Datalog query $Answer(W) \leftarrow \mathtt{p}(?x, ?y, ?n)$.

We omit to present in details the proof of the completeness as it follows the same principle of the proof of the correction.

8 Experimental Results

We present in this section experimentations realized to evaluate the efficiency of the defuzzification and skolemisation approaches for a Datalog based implementation of the FSAQL query language. The experimentations are realized with the $\mathcal{F}$lora-2 Datalog system. Different criteria are used to evaluate their efficiency: the number of membership degrees, the number of instances, the complexity of the query and the expressivity of the knowledge base.

8.1 Membership Degrees

The definition of α-cut classes and properties in the defuzzification approach is based on a set of membership degrees $\mathcal{N}$ which depends on the used semantics: Zadeh [3], Gödel [5] or Lukasiewicz [6]. To transform a fuzzy RDFS knowledge base with n fuzzy classes and a set $\mathcal{N}$ with m membership degrees into a Datalog program, we need to define $2 * n * m$ α-cut classes (2 corresponds to the number of used operators ($>$ and $\geq$)). For example, given a fuzzy RDFS knowledge base with 16 fuzzy classes and a set $\mathcal{N}$ with 10 membership degrees, the number of needed α-cut classes is equal to 3200. In the experimentations, the time needed to query such a knowledge base is equal to 17,285 seconds. The problem that we find consists on the fact that when we use a set $\mathcal{N}$ with more than 12 membership degrees, the approach poses a problem of implementability and the Datalog program does not respond.

Concerning the skolemisation approach, the number of inference rules equally depends on the number of membership degrees. In fact, Datalog rules defined in table 3 which use an inequality between two variables α and β are not allowed with the F-Logic syntax used in the $\mathcal{F}$lora-2 system. That is, if we consider the following Datalog RDFS inference rule:

?S[rdfs_subClassOf(β) -> ?B] :- ?S[rdfs_subClassOf(β) -> ?O],
?O[rdfs_subClassOf(α) -> ?B], $\alpha >= \beta$.

We need to define two different rules with the F-Logic syntax for each membership degree. For example, the rules defined for the membership degree 0.6 are defined as follows:

?S[rdfs_subClassOf(0.6)-> ?B] :- ?S[rdfs_subClassOf(0.6)-> ?O],
?O[rdfs_subClassOf(β)-> ?B], $\beta >= 0.6$.
?S[rdfs_subClassOf(β)-> ?B] :- ?S[rdfs_subClassOf(0.6)-> ?O],
?O[rdfs_subClassOf(β)-> ?B], $\beta < 0.6$.

The number of RDFS inference rules concerned with this rewriting is equal to 4. That is for a fuzzy RDFS knowledge base with m degrees, the number of inference rules is equal to $2 * 4 * m$. In the experimentations (figure 2), we obtain a linear variation of the execution time of a simple query relatively to the number of used membership degrees.

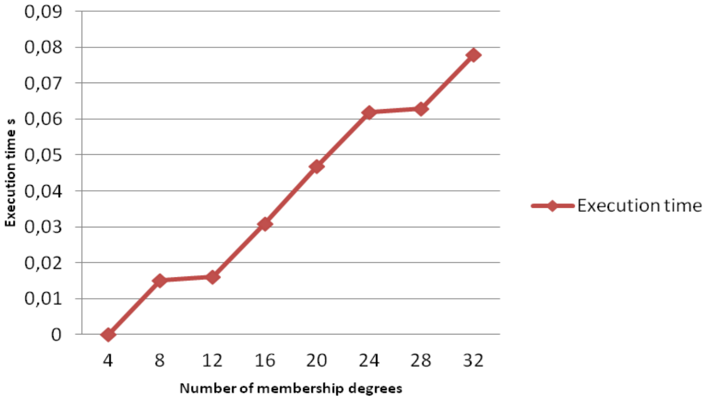


Fig. 2. Linear variation of the execution time of a query in the skolemisation approach

8.2 Size of the Knowledge Base

We present in this section the results of experimentations realized on fuzzy RDFS knowledge bases with different sizes. The FSAQL query used in the experimentations is defined as follows:

```
SELECT  ?X
FROM    <RDF repository>
WHERE   0.6:<?X rdf:type SunnyCountry>
```

The table 4 presents execution times of this query obtained with fuzzy RDFS knowledge bases with different sizes. The table shows for each fuzzy knowledge base the number of solutions, the execution time obtained with the defuzzification and the skolemisation approaches denoted respectively with A1 and A2. The size of the used knowledge is defined from 1600 to 20800 instances. We note that the number of used membership degrees is equal to 12. In fact, as it was seen in the previous section we can not implement the defuzzification approach with more important number of membership degrees. As it is shown in this table and in figure 3 the execution times obtained with the defuzzification approach (A1) are much more important than the ones obtained with the skolemisation approach (A2). In the other hand, the execution time in the two approaches does not increase considerably with the size of the knowledge base. We can talk about a linear variation of the execution time.

Table 4. Execution times obtained with fuzzy RDFS knowledge bases with different sizes

Number of instances	Number of solutions	Execution time A1	Execution time A2
1600	50	17.472	0.078
3200	99	17.909	0.094
4800	149	18.58	0.109
6400	199	18.83	0.125
8000	249	19.219	0.141
9600	299	19.297	0.156
11200	349	19.828	0.172
12800	399	20.498	0.188
14400	449	20.717	0.203
16000	499	21.341	0.218
17600	549	21.903	0.265
19200	599	22.184	0.328
20800	649	22.683	0.359

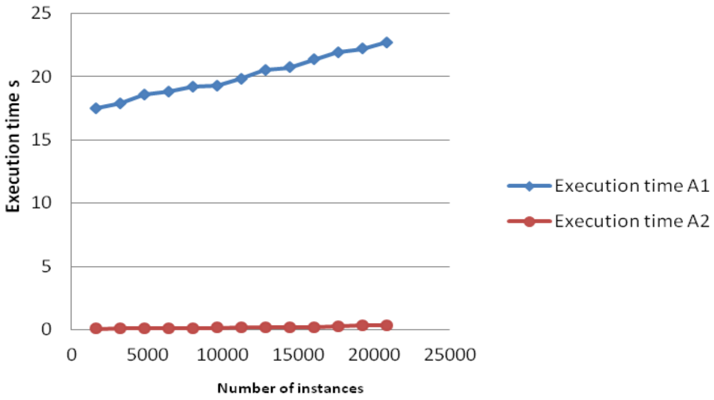


Fig. 3. Execution times of a simple query obtained with the defuzzification (A1) and the skolemisation (A2) approaches

8.3 Query Complexity

We use in this section the same fuzzy RDFS knowledge bases presented previously to evaluate a complex query having the following form:

```
SELECT  ?X, ?Y
FROM    <RDF repository>
WHERE   0.6:<?X rdf:type CheapHotel>
        0.5:<?X rdf:LocatedIn Paris>
```

The experimentations realized for the complex query equally shows the efficiency of the skolemisation approach compared to the defuzzification one. We obtain approximatively the same execution time curves. We present in figure 4, the execution times obtained with complex query compared to the ones obtained with simple query with the skolemisation approach. As we can see, the execution times obtained with this query do not considerably increase compared to the execution times obtained with the simple one.

8.4 Expressivity of the Knowledge Base

The expressivity of a knowledge base may influence the execution time of queries. Equally, we may define complex query if we use more expressive ontology languages. In this section, we enrich fuzzy RDFS with some important OWL2EL descriptors: existential quantification, concept conjunction and role inclusion. Experimentations are realized with queries defined on theses OWL2EL descriptors. The problem that we find with the defuzzification approach, consists on

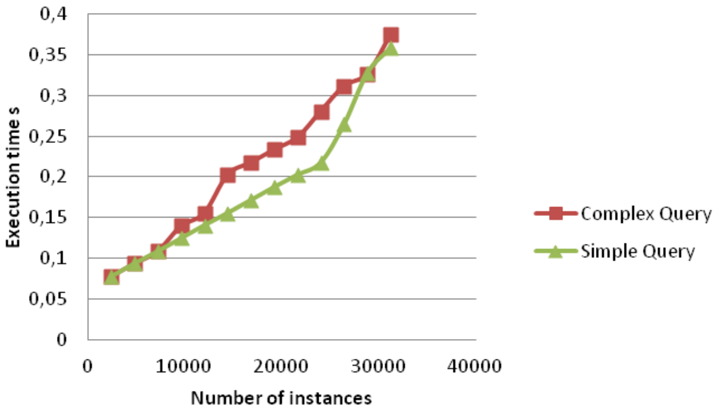


Fig. 4. Execution times of complex and simple queries with the skolemisation approach

the fact that concept conjunction and role inclusion cause a problem of implementability when they are added to a fuzzy RDFS knowledge base defined with the defuzzification approach. Experimentations are only realized for existential quantification (EQ). We define then existential quantification of the form $\exists hasSights.Many \sqsubseteq_{0.6} GoodDestination$ and we ask the following FSAQL query:

```
SELECT  ?X
FROM    <RDF repository>
WHERE   0.6:<?X rdf:type GoodDestination>
```

We present in figure 5 the execution times obtained with the defuzzification approach for the EQ query compared to the ones obtained with a simple query. As we can see the execution times of EQ query relatively increase compared to the ones obtained with a simple query.

Concerning the skolemisation approach, the OWL2EL descriptors do not cause any problem of implementability. We add then fuzzy role inclusion of the form:

$$IsNearBeach \circ hasSunIndex \sqsubseteq_{0.6} hasRisqSwim$$

and concept conjunction of the form:

$$SunnyCountry \sqcap SeaCountry \sqsubseteq_{0.6} SummerDestination$$

We note that the number of inference rules increases when we add existential quantification, role inclusion and concept conjunction. For each descriptor, we need three different rules for each membership degree. That is, if we have m membership degrees, we need to define $9 * m$ new inference rules for a complete

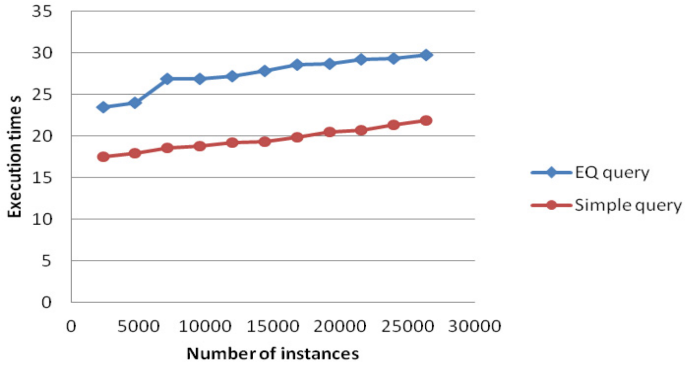


Fig. 5. Execution times obtained with a query about an existential quantification (EQ) and a simple query with the defuzzification approach

definition of these descriptors. As we can see, we still have a linear evolution of the number of inference rules.

The figure 6 shows the execution times obtained with the skolemisation approach for a query about an existential quantification, a role inclusion and a concept conjunction. The execution times obtained with a conjunctive query are much more important than the ones obtained with existential quantification and role inclusion queries. Equally, the figure 7 shows that the existential quantification gives the less important execution times.

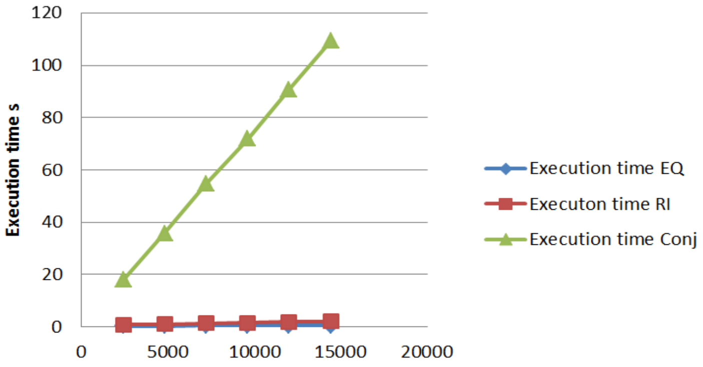


Fig. 6. Execution times obtained with the skolemisation approach for a query about an existential quantification (EQ), a role inclusion (RI) and a concept conjunction(Conj)

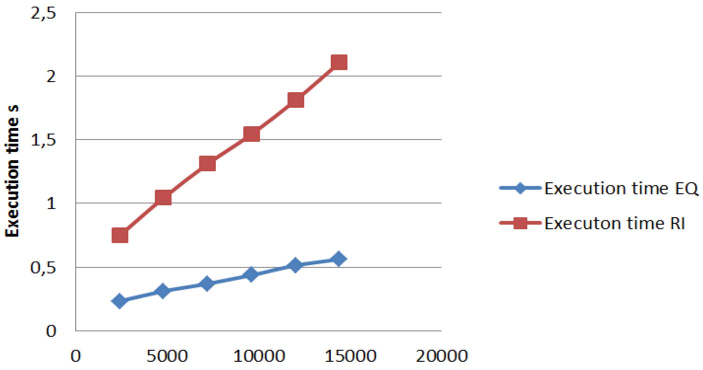


Fig. 7. Execution times obtained with the skolemisation approach for a query about an existential quantification (EQ) and a role inclusion (RI)

8.5 Discussion of the Experimental Results

We present in this section the experimentations that we have realized to evaluate and compare the efficiency of the defuzzification and the skolemisation approaches. Many criteria are used: the number of membership degrees, the size of the knowledge base, the complexity of the query and the expressivity of the ontology language. The experimental results show that the defuzzification approach poses problems of implementability when we increase the number of membership degrees or when we enrich the expressivity of the knowledge base with some OWL2EL descriptors. This is justified by its quadratic complexity which increases with the number of membership degrees and the expressivity of the used ontology language. The skolemisation approach has in the other hand a linear complexity, when we increase the number of membership degrees we equally obtain a linear variation of the execution time. The skolemisation approach leads then to lower execution times compared to the defuzzification approach and does not cause any problem of implementability when we use existential quantification, role inclusion or concept conjunction. Equally, we note that the execution times obtained with the skolemisation approach increase considerably with concept conjunction. The use of databases are generally proposed as a solution in the literature to decrease the complexity of such a kind of queries in the fuzzy DL-Lite works presented in section 3 or in description logic works which use databases for efficient evaluation of conjunctive queries [18,13].

9 Conclusions and Future Works

We propose in this paper a new approach for efficient evaluation of fuzzy RDFS queries based on Datalog. The queries are defined with the FSAQL query language. As there is no known implementation of fuzzy Datalog systems, we use crisp instead of fuzzy Datalog to implement the FSAQL query language. The use of crisp Datalog allows the interoperability of the FSAQL query language. We realize a correct mapping between fuzzy RDFS and crisp Datalog programs. Two approaches are proposed and evaluated in this paper: the defuzzification and the skolemisation approaches. The first approach defines crisp "$\alpha-$cut" classes and properties and maps them into crisp Datalog predicates. The skolemisation approach represents fuzzy classes and properties with crisp Datalog predicates having the same names. The membership degrees are then defined as terms of theses predicates. The experimentations are realized with $\mathcal{F}$lora-2 which is an object-oriented knowledge base language and application development environment based on the XSB datalog system. As future work, we intend to evaluate the skolemisation approach on the fuzzy OWL2EL profile and to integrate the computing of membership degrees on the reasoning process.

References

1. Baader, F., Nutt, W.: Basic description logics. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.) The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Bahri, A., Bouaziz, R., Gargouri, F.: Querying Fuzzy RDFS Semantic Annotations. In: Proc. IEEE International Conference on Fuzzy Systems, Barcelona, Spain, July 18-23 (2010)
3. Bobillo, F., Delgado, M., Gómez-Romero, J.: A crisp representation for fuzzy $\mathcal{SHOIN}$ with fuzzy nominals and general concept inclusions. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005-2007. LNCS (LNAI), vol. 5327, pp. 174–188. Springer, Heidelberg (2008)
4. Bobillo, F., Delgado, M., Romero, J.G.: DeLorean: A Reasoner for Fuzzy OWL 1.1. In: Proc. the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW), Karlsruhe, Germany
5. Bobillo, F., Delgado, M., Gómez-Romero, J.: Optimizing the crisp representation of the fuzzy description logic $\mathcal{SROIQ}$. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005-2007. LNCS (LNAI), vol. 5327, pp. 189–206. Springer, Heidelberg (2008)
6. Bobillo, F., Straccia, U.: Towards a Crisp Representation of Fuzzy Description Logics under Lukasiewicz semantics. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) ISMIS 2008. LNCS (LNAI), vol. 4994, pp. 309–318. Springer, Heidelberg (2008)
7. Cali, A., Gottlob, G., Lukasiewicz, T.: A General Datalog-Based Framework for Tractable Query Answering over Ontologies. In: Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, New York, USA (2009)
8. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. Journal of Automated Reasoning 39, 385–429 (2007)
9. Costa, P., Laskey, K.: PR-OWL: A Framework for Probabilistic Ontologies. In: Proc. International Conference on Formal Ontology in Information Systems, FOIS 2006, Baltimore, Maryland, USA, November 9-11 (2006)

10. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. In: Ma, Z. (ed.) Soft Computing in Ontologies and Semantic Web. STUDFUZZ, vol. 204, pp. 3–29. Springer, Heidelberg (2007)
11. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R.: Well-founded semantics for description logic programs in the semantic web. Transactions on Computational Logic (TOCL) 12(2) (January 2011)
12. Gao, M., Liu, C.: Extending OWL by fuzzy description logic. In: Proc. 17th IEEE International Conference on Tools with Artificial Intelligence, Hong Kong, China (2005)
13. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive Query Answering for the Description Logic SHIQ. CoRR (November 2011)
14. Hustadt, U.: Description Logics and Disjunctive Datalog – The Story so Far. In: Horrocks, I., Sattler, U., Wolter, F. (eds.) Proceedings of the 2005 International Workshop on Description Logics (DL 2005), Edinburgh, Scotland, UK, July 26-28 (2005)
15. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable Rules for OWL 2. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 649–664. Springer, Heidelberg (2008)
16. Loiseau, Y., Boughanem, M., Prade, H.: Evaluation of Term-based Queries using Possibilistic Ontologies. In: Herrera-Viedma, E., Pasi, G., Crestani, F. (eds.) Soft Computing in Web Information Retrieval. STUDFUZZ, vol. 197, pp. 135–160. Springer, Heidelberg (2006)
17. Lu, J., Yu, Y., Tu, K., Lin, C., Zhang, L.: An approach to RDF(S) query, manipulation and inference on databases. In: Fan, W., Wu, Z., Yang, J. (eds.) WAIM 2005. LNCS, vol. 3739, pp. 172–183. Springer, Heidelberg (2005)
18. Lutz, C., Toman, D., Wolter, F.: Conjunctive Query Answering in the Description Logic EL Using a Relational Database System. In: Proc. the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17 (2009)
19. Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with rules. Journal of Web Semantics, 41–60 (2005)
20. Motik, B., Rosati, R.: Reconciling description logics and rules. Journal of the ACM 57(5) (2010)
21. Mazzieri, M., Dragoni, A.F.: A Fuzzy Semantics for the Resource Description Framework. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005-2007. LNCS (LNAI), vol. 5327, pp. 244–261. Springer, Heidelberg (2008)
22. Pan, J.Z., Thomas, E., Sleeman, D.: ONTOSEARCH2: Searching and Querying Web Ontologies. In: Proc. of the IADIS International Conference, San Sebastian, Spain, pp. 211–218 (2006)
23. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E.: Expressive Querying over Fuzzy DL-Lite Ontologies. In: 20th International Workshop on Description Logics, Brixen-Bressanone, Italy (2007)
24. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E.: Scalable querying services over fuzzy ontologies. In: Proceedings of the International World Wide Web Conference (WWW 2008), Beijing (2008)
25. Pérez, J., Arenas, M., Gutierrez, C.: nSPARQL: A Navigational Language for RDF. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 66–81. Springer, Heidelberg (2008)

26. Simou, N., Kollias, S.: FiRE: A Fuzzy Reasoning Engine for Impecise Knowledge. In: K-Space PhD Students Workshop, September 14 (2007)
27. Simou, N., Stoilos, G., Tzouvaras, V., Stamou, G., Kollias, S.: Storing and Querying Fuzzy Knowledge in the Semantic Web. In: Proc. 7th International Workshop on Uncertainty Reasoning for the Semantic Web, Karlsruhe, Germany (October 2008)
28. Simou, N., Stoilos, G., Stamou, G.: Storing and Querying Fuzzy Knowledge in the Semantic Web using FiRE. In: Bobillo, F., et al. (eds.) URSW 2008-2010/UniDL 2010. LNCS (LNAI), vol. 7123, pp. 158–176. Springer, Heidelberg (2013)
29. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Fuzzy OWL: Uncertainty and the Semantic Web. In: Proc. the International Workshop of OWL: Experiences and Directions, Galway, Ireland (2005)
30. Straccia, U.: A Fuzzy Description Logic for the Semantic Web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web, Capturing Intelligence, ch. 4, pp. 73–90. Elsiver (2006)
31. Straccia, U.: Answering Vague Queries in Fuzzy DL-LITE. In: Proc. of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2006 (2006)
32. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with Very Expressive Fuzzy Description Logics. Journal of Artificial Intelligence Research 30(8), 273–320 (2007)
33. Venetis, T., Stoilos, G., Stamou, G., Kollias, S.: f-DLPs: Extending Description Logic Programs with Fuzzy Sets and Fuzzy Logic. In: IEEE International Conference on Fuzzy Systems, London, UK, July 23-26 (2007)
34. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)

Author Index