Content Negotiation in a Decentralised Semantic Context Utilising Equivalence Links

Yousouf Taghzouti[1,*], Antoine Zimmermann[1,*] and Maxime Lefrançois[1,*]

[1]*Mines Saint-Etienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, F - 42023 Saint-Etienne France*

Abstract

The Web is a decentralised system where each Web server can serve a set of URIs that identify resources. Each resource can be associated with one or more representations. This promotes content negotiation, which is the mechanism by which a Web client can request a resource representation that satisfies a set of constraints. This is also true in the Semantic Web, since when resources are described, different vocabularies are used (e.g. *Schema.org* and *FOAF*), and when they are served in the Web, they are serialised in different formats (e.g. *text/turtle* and *application/rdf+xml*). In this context, HTTP provides the means to negotiate representations using media types, and semantic validation languages (e.g. SHACL) could be used to define the constraints that knowledge graphs must conform to. If a resource was only identified by a single URI (as it would be the case with *unique names assumption*), it would mean that all representations would be present on one and only one server. This implies that one would be able to negotiate all representations of a resource with that server. However, this is not possible in the actual Web, because Web standards do not assume unique names and representations are scattered and distributed in different places. Still, a URI can only be served by one server, so in general, several servers should be consulted to get all representations of a resource. Consequently, when we negotiate with a Web server, we only consider a subset of all existing representations. In this article, we propose an approach to perform content negotiation even when representations are dispersed and present in multiple locations. We focus on this specific data management solution by leveraging *equivalence links*, which consists of querying the Web of Data with Content Negotiation, involving on-the-fly SHACL shape validation. To this end, we provide two algorithms, the first in a basic context (i.e. considering only media type constraints) and the second in a semantic context (i.e. also considering SHACL shapes). An implementation of the algorithms as well as separate experiments were conducted to measure the benefits and assess the time requirements of such methods. The conclusion is that utilising *equivalence links* (such as *owl:sameAs*) present in knowledge graphs enables more effective content negotiation of Web resources by allowing the discovery, validation and serving of representations stored in a distributed manner.

Keywords

content negotiation, semantic validation, SHACL, HTTP

ESWC'22: Workshop - Poster, May 28 – June 01, 2022, Hersonissos, Greece

*Corresponding author.
yousouf.taghzouti@emse.fr (Y. Taghzouti); antoine.zimmermann@emse.fr (A. Zimmermann); maxime.lefrancois@emse.fr (M. Lefrançois)
https://youctagh.github.io/ (Y. Taghzouti); https://www.emse.fr/~zimmermann/ (A. Zimmermann); https://www.maxime-lefrancois.info/ (M. Lefrançois)
0000-0003-4509-9537 (Y. Taghzouti); 0000-0003-1502-6986 (A. Zimmermann); 0000-0001-9814-8991 (M. Lefrançois)

1. Introduction

The core design components of the Web are 1) identification of resources through the use of URIs, 2) representation of resources states using media-types , 3) as well as the protocols enabling the interactions in that space, such as the ability to engage in content negotiation (CN), HTTP is an example of a protocol that enables it [1].

CN is the mechanism that enables the selection of a representation from among multiple ones available under the same URI [1]. CN is not a monolithic process, but rather a layered one having several stages (Discovery, Request Formulation, Selection/Adaptation, Response Indication, Response Interpretation) [2]. CN interactions have multiple characteristics [3]: 1) CN dimension (aka. constraints or preferences) which indicates the constraints taken into consideration when selecting the representation of the resource to be delivered, 2) CN style: a way or technique by which the CN process is conducted, that includes how the negotiation is performed and which CN party chooses the representation to be selected, 3) CN constraint conveyance mean: the mean by which the client transmits to the server the CN dimension as well as its value to be considered in the selection stage. HTTP provides the means for negotiating in the media type and language dimensions through the use of *accept* and *accept-language* HTTP headers. However, one can achieve CN in other dimensions such as time using the Memento framework [4], which is convenient in archiving context [5].

The Semantic Web, in contrast to the traditional Web, aims to provide structure to Web content and to enable machines to process the described content in a uniform manner using ontologies, and to this end the Resource Description Framework (RDF) is used [6]. One of the first applications of the Semantic Web to improve CN is the use of the Composite Capability/Preference Profile (CC/PP) standard [7, 8], which allows mobile phones, for instance, to describe their features so that they can negotiate more tailored content.

Over the years, RDF content on the Web used various vocabularies[1], and is accessible in multiple ways, two of them being through SPARQL endpoints and dump data. Users may want to request RDF content that uses certain vocabularies or conforms to some patterns. One technique for achieving this is to use semantic validation languages, which check that an RDF content conforms to the required patterns: SHACL [9] and ShEx [10] could be used for this purpose.

The unique names assumption states that: different names in the world refer to different things. On the Web, this would imply that different URIs should be used to identify different resources. Therefore, a resource identified by a URI would have all its possible representations on a single server. As a result, a client would be able to negotiate a desirable representation of a resource with that server. However, on the actual Web, representations are distributed across different servers, and a URI can only be served by a single one. Consequently, when negotiating with a server, only a subset of all existing representations is considered. Similarly, the Web of Linked Data (the largest publicly available Knowledge Graph (KG)) is inherently distributed, and the unique names assumption does not hold. Fortunately, equivalent links such as *owl:sameAs* can be used to indicate that two URIs actually refer to the same entity, but they have not yet been used in the CN flow.

[1]As shown in the stats from LODStats http://lodstats.aksw.org/stats last accessed: 23rd of February 2023.

The main contribution of our work is to show how equivalent links can be successfully used to promote distributed CN. This is done by discovering alternative URIs of the requested resource present in KGs. Two algorithms are proposed to support this solution, each for a specific context: (1) A basic one, where only media-type constraints are considered. (2) A second one where semantic validation is involved to negotiate representations conforming to a SHACL shape graph. To illustrate the applicability of such an approach, a portal is proposed in which both algorithms are implemented. We show, through experiments, the benefits of using equivalence relations in CN flows by measuring the discovery of new appropriate representations and the time required for such an extended flow.

The remainder of the paper is structured as follows, in Sect. 2, we present a use case that motivates the use of our approach, then in Sect. 3 we present the approach and its use in two different scenarios while discussing the algorithms for its implementation. The technologies and tools used for the implementation is presented in Sect. 4. Next, our experimentation is discussed in Sect. 5, including the experiments and data collection methodology. Then, a review of related work is provided in Sect. 6. Finally, we conclude and point to future directions in Sect. 7.

2. Motivating Use case

Bob is a senior data scientist at *building-twin*[2] based in Rome, Italy, which provides the service of creating a virtual replica of a physical building. It provides its clients with the plans of the city's buildings serialised using the media type *application/vnd.geoplan*. It also provides a description of the buildings in RDF serialised as *text/turtle*, the description uses *Schema.org*, *dbpedia* and *DCAT* vocabularies. Recently, *building-twin* acquired a competitor company, *your-twin-house*, which provides similar services. Its plans of the city's buildings are also provided in the *application/vnd.geospace* media type, but its RDF graphs are serialised in *application/rdf+xml* and mainly use internally created vocabularies as well as some *Schema.org* terms. In addition to this heterogeneity between services, to identify the same *real-world* building, for example the one at the address: *Viale della Moschea, 85, Rome, Italy*, the first company uses: <http://www.building-twin.com/building/B45A124>, while the second identifies it with: <http://www.your-twin-house.com/construction/b/74asd1>.

Our aim is to solve this interoperability problem and allow the client to negotiate an appropriate building representation (based on media type or vocabulary constraints) with both companies data without merging all content into a new silo.

3. Employ CN in a Decentralised Semantic Context

The idea that all representations of a resource must reside on one and only one server follows the unique names assumption: different names refer to different things in the world. But as representations are scattered in different places, this implies that all Web servers would use the same URIs to refer to the same resource. This is not possible in the practical Web, because a URI

[2]This use case is fictional, all the given information is imaginary.

can only be served by one server, so we need different URIs for different servers. Fortunately, the Web Ontology Language (OWL) comes with the *owl:sameAs* property[3], which allows us to indicate that two URIs actually identify the same thing (in our case, the same building).

In this section we present our approach utilising *owl:sameAs*. In the next two subsections we present two algorithms on how CN could be improved. Algorithm 1 presents the idea in a basic context with a dimension such as media type. And Algorithm 2 presents how the improvement could be achieved in a semantic context by negotiating RDF graphs that validate a SHACL shape graph. Finally, we discuss how the approach could be generalised to other dimensions.

3.1. Basic context CN

In the following, the server considers that a client has a resource URI i and a set of constraints $\mathbf{C}$ (in this case, a set of acceptable media types). The client expects to have a representation d (for web document) that validates $\mathbf{C}$, and possibly a set of plausible alternative URIs $\mathbf{I}$ to continue the negotiation if so desired.

The negotiation starts with the Web server serving i trying to check whether any of the representations it has validates $\mathbf{C}$. If not, a common practice is to respond with a *406 (NOT ACCEPTABLE)* or *404 (NOT FOUND)* error or with a preconfigured default (even invalid) representation [11, Section 12]. In our approach, for the opposite case, the negotiation continues by first looking for a set of sameAs URIs $\mathbf{I}_s$.

A straightforward implementation could be to redirect the client to these other potential representations by sending a *300 (MULTIPLE CHOICE)* response to propose other alternatives that could possibly satisfy the constraints[4]. Alternatively, the Web server can go a step further to check whether the constraints are satisfied as described in the Algorithm 1: After obtaining the set of sameAs URIs $\mathbf{I}_s$, the server iterates over it and check whether there is a representation d_s that satisfies the constraints $\mathbf{C}$. In this proposal, the server responds with the first d_s found. For the rest of the URIs, if a valid representation could be found, the server stores its URI i_s in the set of alternative IRIs $\mathbf{I}$ in order to form the final answer pair $\langle d_s, \mathbf{I} \rangle$. If no representation could be found, only then the server responds with an error.

3.2. Semantic context CN

The same idea could be carried over from the Web of documents to the Web of data, but with some modifications to adapt it to the Semantic Web context. First, the server assumes that the client is negotiating RDF documents, and therefore constraints need to be defined for this type of documents (e.g. using SHACL [9] or ShEx [10]). Second, the server can use the First Found First Served approach as explained above (i.e. when iterating over the sameAs URIs $\mathbf{I}_s$, it serves the first RDF graph that validates the requested constraints, e.g. a SHACL shape graph s).

The Algorithm 2 illustrates a more generic serving method, where the server has two variables g_b and i_b to store the best graph and its URI respectively. The function *isABetterRepresentationThan* tests whether a new valid graph is superior to the current best graph, by taking two

---

[3]OWL sameAs: https://www.w3.org/TR/owl-ref/#sameAs-def

[4]The negotiation style considered in this paper is *proactive*. Other styles exist e.g. *reactive*, *transparent*, ect. However, each option inherits the advantages and disadvantages of using such style [3].

Algorithm 1: Decentralised CN Based on Media Type Pseudocode

input : A resource URI i and acceptable constraints (media types + quality values) $\mathbf{C}$
output: A representation d that validates $\mathbf{C}$ if available, and potentially a set of plausible alternative URIs $\mathbf{I}$

```
d = getRepresentation(i,C)
if d != NULL then
    return ⟨d,∅⟩
I_s = getSameAsIRIs(i)
foreach IRI i_s of I_s do
    d_s = getRepresentation(i_s,C)
    if d_s != NULL then
        if d != NULL then
            d = d_s
        else
            add i_s to the set of alternative IRIs I
if d != NULL then
    return ⟨d, I⟩
else
    return No Acceptable representation
```

graphs and producing a Boolean scoring e.g. the scoring could be based on the number of nodes, a graph with more nodes potentially holds more knowledge.

These algorithms could be further generalised to other dimensions [3], e.g. *language* (i.e. to negotiate a representation that uses a preferred language) by following the same model and creating appropriate scoring functions.

4. Implementation

We provide a Java implementation of our algorithms. Our prototype[5] is built using the *Spring framework*[6], we provide two endpoints *</dcn/api/media-type>* and *</dcn/api/profile>*. The resource URI is provided using the *iri* query parameter, and constraints are passed in a header based manner, *accept* and *accept-profile* for the media type and profile dimensions respectively. The *Alternates* header is used to provide the set of alternative URIs[7]. The *sameAs service*[8] is used to obtain the set of equivalent URIs. *SHACL* (the W3C recommendation) is used to write the constraints that RDF sources must conform to, while *Apache Jena*[9] is used to manipulate the

[5]Github repository: https://github.com/YoucTagh/decentralised-cn
[6]Spring Framework: https://spring.io/
[7]The same approach is also used by *dbpedia* to provide alternatives
[8]SameAs service homepage: http://sameas.org/
[9]Apache Jena: https://jena.apache.org/

Algorithm 2: Decentralised CN Based on SHACL Shapes Pseudocode

input : A resource URI i and acceptable constraints (SHACL shape graph) s
output : An RDF representation that validates s if available, and potentially a set of plausible alternative URIs $\mathbf{I}$

g = `getRepresentation`(i,s)
if g != *NULL* **then**
  return $\langle g, \emptyset \rangle$
$\mathbf{I}_s$ = `getSameAsIRIs`(i)
g_b = NULL , i_b = NULL
foreach *IRI* i_s *of* $\boldsymbol{I}_s$ **do**
  g_s = `getRepresentation`(i_s,s)
  if g_s != *NULL* **then**
    if `isABetterRepresentationThan` (g_s,g_b) **then**
      $g_b = g_s$
      add i_b to the set of alternative IRIs $\mathbf{I}$
      $i_b = i_s$
    else
      add i_s to the set of alternative IRIs $\mathbf{I}$
if g_b != *NULL* **then**
  return $\langle g_b, \boldsymbol{I} \rangle$
else
  return *No Acceptable representation*

RDF graphs and perform validation. Swagger[10] is used to provide friendly API documentation.

Figure 1 shows a request example where a client wants to obtain a representation of the resource identified by the URI <http://www.uniprot.org/taxonomy/3330> and conforming to the SHACL shape graph Listing 1. Note that the original server can only provide an HTML representation. The same request could be issued using curl using the Listing 2.

Listing 1: An example of a SHACL shape document

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix ex: <https://example.com/ontology#> .
...
ex:tagForComments a sh:NodeShape ;
  sh:targetSubjectsOf rdfs:comment ;
  sh:property [ sh:path rdfs:comment ;
                sh:qualifiedValueShape [ sh:languageIn ( ”en” ”fr” ) ;] ;
                sh:qualifiedMinCount 1 ; ].
```

[10]Swagger UI: https://swagger.io/tools/swagger-ui/

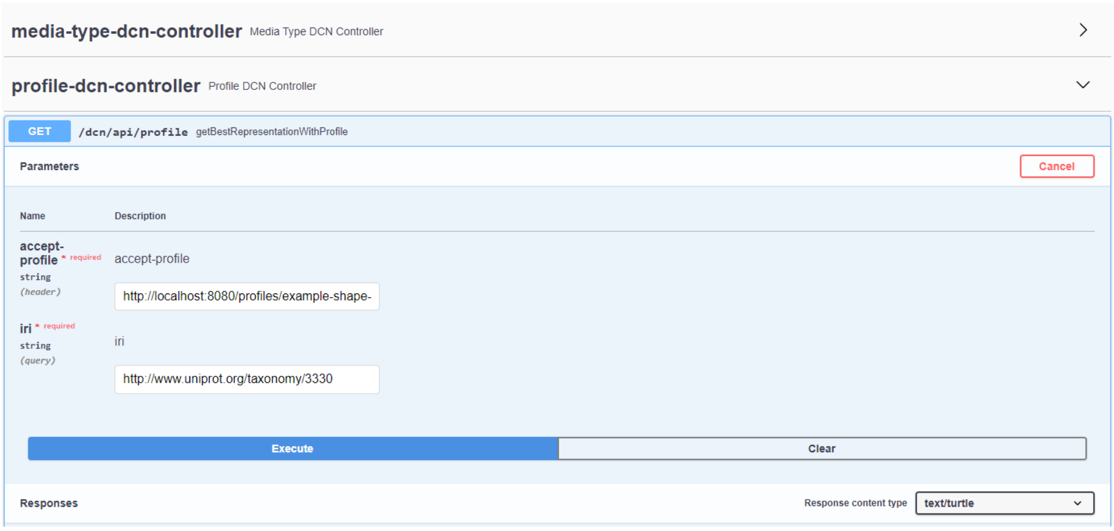

Figure 1: A request sent using the Swagger-UI to get a representation of a resource that validates a SHACL shape graph.

Listing 2: The same request sent in Figure 1 but using curl

```
curl -v http://localhost:8080/dcn/api/profile?iri=http://www.uniprot.org/taxonomy/3330 -H "accept-profile:
http://localhost:8080/profiles/example-shape-graph-1.ttl"
```

5. Experimentation

After having tested the proposed implementation in Section 4 with a few manually selected URIs, we carried out experiments in order to test the following hypotheses: (1) In case no acceptable representation is available at the initial URI (when negotiating some RDF serialisation, such as *text/turtle*, *application/rdf+xml* ...), our approach would increase the chance of finding an alternative utilising equivalence links when available. (2) The *sameAs.org* portal could be used as a reliable third party medium to find similar entities (e.g. it does not have an API call limit). This section provides details on the selected dataset, the experimental setup, and an analysis of the results obtained. The code, data and results of the experiments are publicly available[11].

Data Our data collection methodology is as follows: (1) collect 5+ sets of 75+ URIs. Each set of URIs has a homogeneous number of equivalent links. (2) Test if the *sameAs.org* service can handle and allow a large number of API calls. In addition, we want context-free entities, meaning that no assumptions are made about the type of data. To this end, we use the Wikidata identifiers for items[12]. This means that URI http://www.wikidata.org/entity/Q{id} requests are sent to the *sameAs.org* API by replacing the *{id}* part in the URI with an integer. In this

[11]Github repository: https://github.com/YoucTagh/decentralised-cn-experiment
[12]Wikidata identifiers: https://www.wikidata.org/wiki/Wikidata:Identifiers

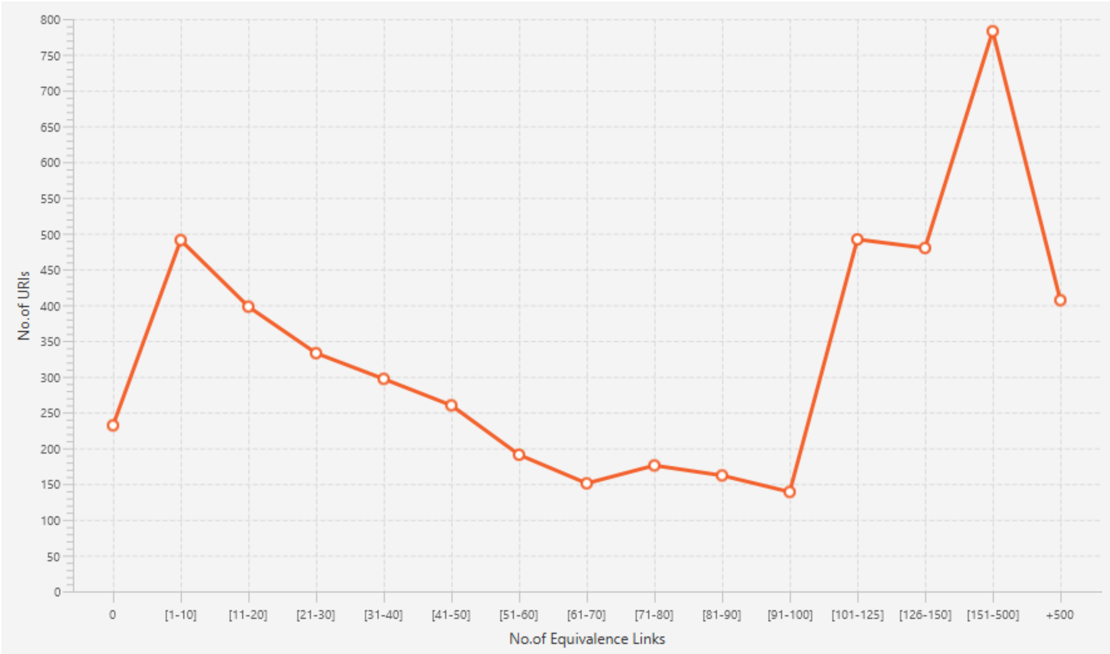

Figure 2: The statistics of the responses

experiment, the identifiers range from 1 to 5000. The statistics of the responses are provided in Figure 2.

From all the responses we create six subsets: $[1, 5]$, $[10, 15]$, $[25, 30]$, $[45, 50]$, $[70, 75]$, $[+100]$, each with 100 URIs. A subset $[x, y]$ means that all its URI elements have between x and y equivalence links. URIs are added to the relevant subset until it is full (i.e. first found first added).

Experimental Setup The code experiments are written in Java and the Apache Jena framework is used to manipulate RDF graphs and perform validation of SHACL shape graphs. The used machine has an Intel(R) Xeon(R) CPU E3-1505M v6 @3.00GHz processor with 16GB of RAM. For each of the six subsets created: *a)* start by deleting the wikidata URI from the equivalence links[13] *b)* randomly choose a URI from the equivalence links to be our initial URI[14] *c)* Request a representation that validates the constraints:

- **HTML:** request an HTML representation.
- **RDF:** request one of the following RDF representations (*Turtle, RDF+XML, N3*).
- **Turtle + SHACL:** request an RDF representation with a Turtle serialisation that validates a SHACL shape graph.

[13]The *sameAs.org* service sends back the equivalence links, if any, plus the requested URI.
[14]A seed is used in the random selection to allow reproducibility of the results.

Table 1
Decentralised Content Negotiation Experiment Results

Constraint	Metric	[1,5]	[10,15]	[25,30]	[45,50]	[70,75]	[+100]
HTML	Initial URI	5	20	22	31	20	17
	SameAs URI	13	79	78	69	80	83
	Avg Time	863 ms	944 ms	845 ms	871 ms	1077 ms	822 ms
RDF	Initial URI	53	44	37	40	29	25
	SameAs URI	47	56	63	60	71	75
	Avg Time	310 ms	381 ms	657 ms	480 ms	556 ms	393 ms
Turtle + SHACL	Initial URI	15	19	15	20	13	11
	SameAs URI	7	17	13	8	8	8
	Avg Time	277 ms	344 ms	410 ms	400 ms	460 ms	317 ms

If a representation is found, increment the *Initial URI* score. Otherwise, test whether any of the equivalent links validate the constraints. If so, increment the *SameAs URI* score. For each of the subsets and each constraint type, the average response time is calculated[15].

Results The results of the experiments are depicted in Table 1. They show that, at best, only 30% of the randomly selected URIs have a representation that validates the *HTML* constraint, while up to 50% validate the *RDF* constraint (we believe that this augmentation is rational, since the new URIs are found through equivalence links that are expressed in RDF) and, understandably, at most 20% validate the *Turtle + SHACL* constraint[16]. This low percentage is due to either a broken URI or a non-conforming constraint. For each of the subsets we see a non-zero value of the *SameAs URI* score, ranging from 69% to 83% for the *HTML* constraint in all subsets except the first with only 13% this is partly due to the low number of equivalence links only $[1, 5]$. For the *RDF* constraint, the contribution is $56 - 75\%$, and for the *Turtle + SHACL* constraint, the addition is noteworthy at $7 - 17\%$.

Experiment Reproducibility To reproduce the experiment, a main command line interface is provided, when launched a menu with the available options is displayed. The first option will start the data collection process. The second option will use the collected data to start the experiment process. The third option is available to produce a human readable version of the results.

6. Related Work

CN has been proposed as an essential layer of the Web architecture since its creation [12]. The HTTP protocol was designed to allow the use of CN in different formats[17], and its benefits

[15]For constraint (1) and (2), requests are sent using the HTTP HEAD method.
[16]The SHACL constraints express that the representation with *rdfs:label* and *rdfs:comment* should have at least one language tag in French or English (Available in the aforementioned Github repository).
[17]Format negotiation in the initial W3 project: http://info.cern.ch/hypertext/WWW/DesignIssues/Formats.html

were outlined at an early stage in the HTTP negotiation algorithm[18] (e.g. "it allows a generic resource to exist which refers to many different specific resources, specific e.g. by language, format, etc."). CN has stood the test of time as its use is encouraged in various places [13, 14, 11], and it has evolved over the years (e.g. CN is possible in multiple additional dimensions [3]).

RDF is intended to describe resources on the Web using vocabularies and ontologies, and this knowledge can be captured in knowledge graphs. Tools such as *RDF Browser*[19] allow for the negotiation and visualisation of such a representation, albeit relying solely on the negotiation of media types. The Data Exchange Working Group has proposed the profile vocabulary [15], and ways to negotiate in the profile dimension [16]. A profile can play different roles, one of which is validation and is created using a constraint language (e.g. SHACL [9]]). A separate Internet Engineering Task Force (IETF) effort is investigating approaches to specify, discover, negotiate, and write profiled representations [17].

Equivalence links, in particular *owl:sameAs*, have been addressed in formal studies, where a quantitative analysis of *owl:sameAs* as well as a dataset[20] has been proposed [18]. A survey on identity management in the Web of Data has also been carried out, where *the problem of sameAs* has been highlighted [19]. Other alternative equivalence links exist and could also be used to discover potential representations, including: (1) *skos:exactMatch* and *skos:closeMatch* from the *SKOS* ontology[21] (2) *wdt:P2888* by *Wikidata*[22] (3) *umbel:isLike* in *Upper Mapping and Binding Exchange Layer ontology*[23] (4) *schema:sameAs* in *Schema.org*. (5) and finally the predicates introduced in the *similarity ontology* [20]. The sameAs portal is a service that provides equivalent URIs, and is available as a website or through an API[24].

In our work, we propose to use equivalence links for the specific application of enabling CN in a decentralised context. To this end, we leverage for example *owl:sameAs* to discover potential acceptable representations and proceed to their validation.

7. Conclusion

In this paper we propose an approach to achieve decentralised content negotiation using Semantic Web technologies. We first describe a motivating use case and extract the problem to be addressed. We then present the methodology and how we used equivalence links such as *owl:sameAs* to discover potential representations. In that section, we present two algorithms for negotiating in two dimensions, the first being the media type for negotiating Web documents, and the second being the profile for negotiating RDF sources that conform to a SHACL shape graph. We then present the implementation we developed to verify the feasibility of our approach based on the presented algorithms. We have carried out an evaluation and the results show that our approach contributes to the overall increase in the availability of resource representations. In our work we only consider one hop, which means that we only check the

[18] HTTP negotiation algorithm (1992): https://www.w3.org/Protocols/HTTP/Negotiation.html
[19] RDF Browser Addon: https://addons.mozilla.org/en-US/firefox/addon/rdf-browser/
[20] The Extended SameAs network (ESameNet) dataset.)
[21] SKOS: https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html
[22] In Wikidata the property P2888 is "Exact match": https://www.wikidata.org/wiki/Property:P2888
[23] Umbel: https://lov.linkeddata.es/dataset/lov/vocabs/umbel
[24] SameAs portal: http://sameas.org/

direct representation *owl:sameAs*, therefore our idea for future work is to test incrementally more hops to study their efficiency and eventually extract an optimal number of hops.

References

[1] I. Jacobs, N. Walsh, Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004, W3C Recommendation, W3C, 2004. URL: http://www.w3.org/TR/2004/REC-webarch-20041215/.

[2] Y. Taghzouti, D. Vachtsevanou, S. Mayer, A. Ciortea, A step toward semantic content negotiation, in: 23rd International Conference on Knowledge Engineering and Knowledge Management, 2022.

[3] Y. Taghzouti, A. Zimmermann, M. Lefrançois, Content Negotiation on the Web: State of the Art, Technical Report abs/2204.10097, CoRR, 2022. URL: https://arxiv.org/abs/2204.10097.

[4] H. Van de Sompel, M. L. Nelson, R. Sanderson, HTTP Framework for Time-Based Access to Resource States - Memento, RFC 7089, IETF, 2013. URL: https://www.rfc-editor.org/rfc/rfc7089.txt.

[5] M. Kelly, S. Alam, M. L. Nelson, M. C. Weigle, Client-Assisted Memento Aggregation Using the Prefer Header, in: M. Klein, E. A. Fox, Z. Xie (Eds.), Proceedings of the 2018 Web Archiving & Digital Libraries Workshop (WADL 2018), June 6, 2018, Fort Worth, Texas, USA, Virginia Tech University Libraries, 2018, pp. 5–6. URL: https://vtechworks.lib.vt.edu/bitstream/handle/10919/97988/WADL2018.pdf#page=5.

[6] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (2001) 34–43. URL: https://www.scientificamerican.com/article/the-semantic-web/.

[7] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, L. Tran, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation, W3C, 2004. URL: https://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[8] M. H. Butler, Implementing content negotiation using CC/PP and WAP UAProf, Technical Report HPL-2001-190, HP Lab, 2001. URL: https://www.hpl.hp.com/techreports/2001/HPL-2001-190.html.

[9] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3C Recommendation, W3C, 2017. URL: https://www.w3.org/TR/2017/REC-shacl-20170720/.

[10] E. Prud'hommeaux, I. Boneva, J. E. Labra Gayo, G. Kellogg, Shape Expressions Language 2.1, W3C Community Group Report, W3C, 2019. URL: http://shex.io/shex-semantics-20191008/.

[11] R. T. Fielding, M. Nottingham, J. F. Reschke, HTTP Semantics, RFC 9110, IETF, 2022. URL: http://www.ietf.org/rfc/rfc9110.txt.

[12] T. Berners-Lee, R. Cailliau, J.-F. Groff, B. Pollermann, World-Wide Web: The Information Universe, Electronic Networking: Research, Applications and Policy 2 (1992) 74–82. URL: https://doi.org/10.1108/eb047254.

[13] L. Sauermann, R. Cyganiak, Cool URIs for the Semantic Web, W3C Note, World Wide Web Consortium, 2008. URL: https://www.w3.org/TR/cooluris/.

[14] B. Farias Lóscio, C. Burle, N. Calegari, Data on the Web Best Practices, W3C Recommendation, W3C, 2017. URL: https://www.w3.org/TR/2017/REC-dwbp-20170131/.

[15] R. Atkinson, N. J. Car, The Profiles Vocabulary, W3C Working Group Note, W3C, 2019. URL: https://www.w3.org/TR/2019/NOTE-dx-prof-20191218/.

[16] L. G. Svensson, R. Atkinson, N. J. Car, Content Negotiation by Profile, W3C Working Draft, W3C, 2019. URL: https://www.w3.org/TR/2019/WD-dx-prof-conneg-20191126/.

[17] L. G. Svensson, R. Verborgh, H. Van de Sompel, Indicating, Discovering, Negotiating, and Writing Profiled Representations, Internet-Draft, Internet Engineering Task Force, 2021. URL: https://www.ietf.org/archive/id/draft-svensson-profiled-representations-01.txt.

[18] L. Ding, J. Shinavier, Z. Shangguan, D. L. McGuinness, SameAs Networks and Beyond: Analyzing Deployment Status and Implications of owl:sameAs in Linked Data, in: The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I, 2010, pp. 145–160. URL: https://doi.org/10.1007/978-3-642-17746-0_10.

[19] J. Raad, N. Pernelle, F. Saïs, W. Beek, F. van Harmelen, The sameas problem: A survey on identity management in the web of data, CoRR abs/1907.10528 (2019). URL: http://arxiv.org/abs/1907.10528.

[20] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, H. S. Thompson, When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data, in: The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I, 2010, pp. 305–320. URL: https://doi.org/10.1007/978-3-642-17746-0_20.