# Semantic content negotiation for knowledge exchange in heterogeneous systems

Yousouf Taghzouti*
yousouf.taghzouti@emse.fr
Mines Saint-Étienne, Univ Clermont Auvergne, INP Clermont Auvergne, CNRS, UMR 6158 LIMOS
Saint-Étienne, France

## ABSTRACT

Resources on the Web are identified by uniform resource identifiers (URIs). Each resource can have several representations describing it that we call variants. A user (client) wanting a particular representation of that resource makes a request to its URI with a set of constraints. Content negotiation (CN) is the process by which a server matches the client's preferences to the set of variants the resource has. For a long time, the constraints that differentiated the variants were mainly media type and language. Later, variants were created to represent different evolutionary versions of the resource. However, today, with the use of semantic web technologies, especially RDF, a resource can be described using multiple vocabularies and ontologies and adhering to different web ontology language profiles. A finer negotiation of the content is therefore necessary, and the server response must be flexible, especially to answer a request if no representation perfectly validates all the constraints. In this work, we present the challenges associated with solving the fine-grained CN problem in a heterogeneous web environment. We present our research questions to address this problem and the proposed hypotheses. Finally, we describe our methodology, including an evaluation plan, and provide preliminary results obtained and the direction we intend to take from here.

## CCS CONCEPTS

• **Information systems → Web searching and information discovery**; **Information retrieval**.

## KEYWORDS

Content negotiation, semantic web, constraint.

**ACM Reference Format:**
Yousouf Taghzouti. 2022. Semantic content negotiation for knowledge exchange in heterogeneous systems. In *Proceedings of The Web Conference 2022 (WWW '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

*Supervised by Antoine Zimmermann and Maxime Lefrançois.





## 1 INTRODUCTION

Open, distributed, accessible and heterogeneous are some of the fundamental characteristics of the Web [4]. Although the fact that anyone can access the web from anywhere in the world has greatly contributed to its development and enrichment through its openness, this has had the opposite effect of having an abundance of web resources and difficulties in providing the best content for each client; a simple example is that of two people speaking different languages accessing the same resource, the server with the resource should be able to provide each client with an understandable version. To remedy this, a solution was devised from the start, having a negotiation layer between the client and the server [3]. It is discussed later in the Architecture of the World Wide Web document as one of the essential components of Web design [14, section 3.2].

Negotiation as a concept is a back and forth communication intended to reach an agreement when two or more parties have common interests and others opposed [9, p. 1]. Applied to the Web, it becomes the mechanism for selecting the appropriate representation when processing a request. In HTTP one can express and transmit constraints referred to as preferences in [8, section 5.3]. And with that, in addition to finding and transmitting information, it is possible to select more specific formats and languages.

With the mobile era, a new challenge emerged, the content already available was designed to fit computer screens, not phones. Once again, CN was needed to negotiate what was appropriate for these devices based on their capabilities [15].

Resources on the web were primarily intended for humans, not machines. Not having semantically understandable content hindered their full exploitation, but this changed with the introduction of semantic web languages to describe web content and provide a way for machines to understand it. This was achieved by first using the Resource Description Framework (RDF) [12]. Then, using OWL and a variety of vocabularies [11]. Nevertheless, this diversity revealed the need for fine-grained negotiation that goes beyond the simple format or language as it was.

In this PhD work, we are interested in how to make CN more refined in/with the semantic web. We want to use the semantic web for negotiation (1) and use negotiation when it comes to the semantic web (2).

**(1)** In negotiation, we use semantic web technologies, e.g. metadata describing a resource in different vocabularies.
**(2)** When we want to have specific semantic web resources, e.g. by requesting a resource using the profile vocabulary.

The remainder of this paper is structured as follows: Section 2 presents the problem statement, use cases motivating it, research questions and hypotheses, the scope of our research and the expected contributions, followed by the state of the art and related work in Section 3. Section 4 briefly describes the proposed approach, while Section 5 describes the planned research methodology. Preliminary results are described in Section 6, followed by the conclusion and future work in Section 7.

## 2 PROBLEM

### 2.1 Problem Statement

A resource available under a specific Uniform Resource Identifier (URI) may have different representations which we call variants or alternatives as in [7, section 1.3]. CN is the mechanism for choosing the best representation from among the available variants. The client includes a set of constraints in the request while the server delivers a selected representation when it matches its own constraints [8].

The questions that immediately arise are: what are these constraints? how to express them and how to match them? Over the years, constraints have taken many forms, starting with format, language, and content encoding [8], through constraints related to the device's ability to process a certain representation [15] to being more complex to indicate a semantic interpretation [2].

For a server, in addition to the interpretation and matching of the constraints, another problem is the explanation of the choice, either of the selected representation, or of the alternatives provided that have been judged sufficiently close to the one requested.

### 2.2 Use case

*2.2.1 Negotiation of vocabulary.* Alice runs a portal that harvests and collects data describing ancient artefacts (creator, material used, date of creation, etc.) from different sources, such as museums and Wiki-data. She has noticed that each source uses different vocabularies, sometimes customised. Lately, she has received many requests to find a way to search for data in a specific vocabulary. Or to specify the desired vocabularies in an orderly fashion. For example, exposing creator data using a vocabulary such as FOAF (Friend Of A Friend), Schema.org or DCMI (Dublin Core Metadata Initiative).

Currently, the data graphs available on the APIs use the same media type *text/turtle*. The user has to query all the data graphs manually to select those that use the desired vocabulary.

*2.2.2 Negotiation of RDF shapes.* John is a researcher interested in the evolution of youth unemployment in different societies; he needs data in the form of RDF data graphs. To do this, he queries the data graphs available in various web APIs queried by the university portal. These scenarios are plausible:

**Scenario 1 - The shape is also important:** John needs a representation that conforms to a specific shape. Therefore, vocabulary negotiation is not enough because he would have to manually validate all returned data graphs with the desired vocabularies.

**Scenario 2 - Flexibility vs rigidity:** In scenario 1, the negotiation can be rigid in case John wants *all* the constraints to be valid, and prefers not to have an answer otherwise. Else, the negotiation can be flexible in case John agrees to receive a representation even if it does not satisfy all the constraints.

**Scenario 3 - Not all constraints are equally important:** For John, not all shape constraints have the same degree of importance. He therefore wants a way to express this importance for each constraint and to obtain the representation that minimises the violation rate by taking it into account

### 2.3 Research questions and hypotheses

In this research we aim to address the following questions:

- **RQ1** What are the characteristics of a CN use case, and how can they be compared and classified?
- **RQ2** How to formalise in a uniform way the different styles and dimensions of CN?
- **RQ3** How to use semantic validation to request the best representation that partially validates the request among a set of variants?
- **RQ4** How to assess feasibility and quality of the proposed algorithms and methods??

Our hypotheses are derived directly from the research questions:

- **H1** Creating a resource targeting CN documentation, in addition to providing CN use cases with their existing or potential solutions would encourage the use of CN.
- **H2** Semantic web technologies can contribute to the CN mechanism.
- **H3** SHACL is not only useful for validating RDF graphs but can be used to introduce some flexibility in the process of choosing the best representation.

### 2.4 Scope

At this stage of the thesis, we set the scope of the research to include CN based on user preferences and semantic interpretation, which means that for example, network conditions, device characteristics, device capabilities, battery status, monetary cost are excluded for now. However, although we are interested in the CN mechanism in general. once we instantiate our general model, we will do so primarily using RDF sources [5].

This scope is manifested in the formalisation, we started with the document class, then its sub-classes, however we are not interested at this stage in the speed of the data streams for example. It is worth mentioning that this scope will not prevent us from specifying use cases employing negotiation, for example, the use case of negotiating the frequency of updating things (data freshness) in a Web of Things (WoT) environment in the Content Negotiation Theoretical Framework (CNTF) resource that we implemented as part of this work. However we will not consider it when formalising the problem.

### 2.5 Expected Contributions

The expected contributions of this research include:

- **C1** A web application that lists, categorises and links CN techniques, use cases and related work. As well as acting as a dissemination platform for our future contributions.
- **C2** A formalisation of the CN problem from broad (document-based negotiation) to specific (RDF Graph).
- **C3** A solution for fine-grained CN using semantic validation languages, such as SHACL.
- **C4** Implementation and validation of the proposed solution.

## 3 STATE OF THE ART

CN has been proposed as an essential layer of the Web architecture since the beginning [3], and has been implemented in the HTTP protocol by providing the means to negotiate variants through headers among others: *accept* to express a constraint on the format of the representation, and *accept-language* to select the preferred language [8]. As well as a set of response status codes to be used to indicate if a specific HTTP request has been completed, but in most cases these codes are generic and do not provide any explanation.

With HTTP, different styles of CN have emerged, including proactive CN, which makes the server responsible for choosing the best alternative, and reactive CN, in which the server provides a list of alternatives and it is up to the client to choose the best one. Also worth mentioning is transparent CN [13], which allows proxies to choose on behalf of the server by taking advantage of the *vary* HTTP header [8].

Unfortunately, the basic HTTP headers are not sufficient, but the protocol can be extended with other headers. Examples of custom headers are *prefer* to request that certain behaviours be used by a server [24], *accept-presentation* to negotiate the RDF presentation [17], *accept-schema* to request how the resource should be structured [26]. This approach solves the problem but is not scalable considering the fact that a new header is created for each new requirement. In addition, interoperability is not achieved since the new header must be known in advance.

Capabilities are another dimension that must be negotiated. For the user agent the *user-agent* header was used but when mobile devices came on the scene a new approach was needed, CC/PP (Composite Capabilities/Preference Profiles) and UAProfile were introduced to solve this problem [15, 21]. But even today, as the use of the Web of Things grows, negotiation must take into account their limitations, such as low processor and battery power.

RDF is intended to describe resources on the web using vocabularies and ontologies. And to negotiate these representations, the Dataset Exchange Working Group proposed the profile vocabulary [2], and ways to negotiate profiles that could take the form of constraint language resources e.g. SHACL or ShEx [16, 23, 27, 28]. However, this lacks flexibility, for example in the case where a server has to choose between two representations that only partially validate a profile.

To address the performance and privacy impacts of sending headers that are not reliably used in processing a client request, a recent proposal is Client Hints [10]. An *Accept-CH* response header is introduced that servers can use to announce their use of request headers for proactive CN. However, in our opinion, the adoption of headers contributing to CN is insufficient due to the lack of a guidance resource. This resource should document the approaches and features of NC, and present them in a digestible way in the context of use cases. The latter would include their respective or potential solutions and pointers to other links for further investigation.

## 4 PROPOSED APPROACH

In this section, we discuss the approach taken to address the different research questions, which includes a theoretical framework that takes the form of a web application, a formalisation and an algorithm to increase the flexibility of CN.

### 4.1 Content Negotiation Theoretical Framework

We believe that CN is one of the pillars of the Web and deserves to be studied in more depth. We plan to start with a well-designed documentation and presentation for newcomers who want to learn about CN, its features and use cases, or for experts who want to keep up with the latest techniques and technologies used [**RQ1**]. To this end, we propose to create a theoretical framework to categorise and benchmark different use cases of CN, and we want to materialise this in a resource taking the form of a web application that we present in the following section.

CNTF (Content Negotiation Theoretical Framework) is a web application that categorises the characteristics of CN into different CN groups: style, dimension, etc. CNTF aims to collect use cases of CN, highlight existing solutions if available, or suggest plausible ways to advance them. These are the main objectives of the CNTF. Later, it will be used for the dissemination of our new propositions to advance CN. CNTF attempts to meet these requirements:

**Navigable design:** CNTF should have a navigable design that includes the ability to move from one concept to another, e.g., from a use case to the CN dimension or style being used via links.

**Extensible:** One of the main contrasts between a traditional survey document and the CNTF resource is that it should be extensible by allowing for the addition of new CN concepts, e.g., a new dimension, and be up-to-date with different terminology and definitions.

**Categorisable:** The CNTF must provide the means to categorise different CN use cases and techniques to allow for comparative evaluation. It should also have a well thought out grouping and modelling of different CN concepts to facilitate understanding.

**Maintainable:** CNTF should promote maintainability by adjusting the model used, e.g., community-recommended vocabularies, and by taking into account feedback and comments provided by CNTF users to clarify and rectify the content.

### 4.2 A bottom-up formalisation of CN

Another way to study semantic CN is to have more precision and rigour in its formalisation, for this we suggest a bottom-up formalisation of CN [**RQ2**]. Although there have been attempts to formalise CN, it has been done mainly to fit a certain context, such as content adaptation [19]. In our path to formalise CN in semantic contexts, we plan to take a step-by-step approach from basic web document negotiation to specialised graph document negotiation using semantic validation languages e.g. SHACL.

The goal is to define in a formal way what is a client request with its constraints, what a server must provide as an answer, what are some of the strategies to choose the representation to return. Then, we progressively add complexity such as how to express a redirection when the server cannot satisfy the request but can provide a URI that would help. And later on, we can define the protocols formally.

### 4.3 A step towards semantic CN

To solve the use cases already presented and achieve flexible CN in a practical way [**RQ3, RQ4**], we use SHACL. Specifically, we use the recently introduced header *accept-profile* to request a variant that validates a set of constraints in the form of SHACL documents [27]. A client makes a request with a SHACL document, the server in the traditional procedure has the set of profiles corresponding to the variants, if a variant conforms to the requested profile it is served otherwise a 406 (Not Acceptable), a 404 (Not found) or a 300 (Multiple Choices) status code are returned depending on the server configuration[1]. In our approach, we propose to validate on the fly the requested profile with the list of available variants, and to provide the closest variant if several partially validate the constraints. We developed a simple algorithm to show how this could be achieved:

The algorithm takes as input a list of SHACL document URIs $S$ that represent the client's constraints, each with an optional numeric value $q_s$ indicating the preference of that profile. The server has a list of data graphs (variants) $G$ from which to choose. The output of this algorithm is a data graph that has the minimum number of violations. For each of the SHACL documents $s \in S$, we validate the available data graphs and record the number of tested constraints $n_c$ as well as the valid constraints $v_c$. Then, we compute the validation measure with the formula:

$$v_m \leftarrow \frac{v_c}{Max(1, n_c)} \times q_s$$

Once each pair (shape document, data graph) has a validation measure, we deliver the one with the best score.

Another approach to increasing the flexibility of CN is constraint containment [1, 18, 22, 25]. Taking the SHACL document as constraints, we can say that a shape is contained in another shape if every node in a graph satisfying the constraints of the first shape also satisfies the constraints of the second. Let us consider three shape graphs ($SG$):

$SG_1 = \{S_1, S_2, S_3\}, SG_2 = \{S_1, S_2, S_3, S_4, S_5\}, SG_3 = \{S_1, S_3, S_6\}$

We can imagine three negotiation cases:

**Case 1:** the client request $SG_1$ which is contained in $SG_2$ because all the $S_n$'s in $SG_1$ are contained in $SG_2$.

**Case 2:** the client request $SG_2$ which is partially contained in $SG_1$ because some $S_n$ of $SG_2$ are contained in $SG_2$ but $SG_1$ has no additional $S_n$.

**Case 3:** the client request $SG_1$ or $SG_3$. In both cases we have extra $S_n$, $SG_1$ is partially contained in $SG_3$ and $SG_3$ is partially contained in $SG_1$.

The following server behaviours can answer these cases:

**Case 1:** As the requested $SG$ is fully contained in the available $SG$, we validate the data with the requested one and return it.

**Case 2:** Since the requested $SG$ is partially contained in the available $SG$, we can either validate the additional shapes to the data graph or return only the data graph with the available $SG$ applied with precision and recall.

**Case 3:** Compute the precision and recall between the requested shape graph and the available one. Then build a new $SG$ including only the shapes with a positive containment test. Finally, send it back to the user with the original shape graphs along with the precision and recall measures.

[1]Multiple discussions within the Data Exchange Working Group to address these issues, e.g. https://github.com/w3c/dx-connegp/issues/5

## 5 METHODOLOGY

The methodology adopted in the development of this doctoral work adheres to the following tasks:

(1) Investigation of the state of the art of research relevant to the identified problem. This includes the study of the literature on CN techniques, formulation and validation of constraints in the Semantic Web domain.
(2) Formalisation of the CN problem.
(3) Creation of the structure and categories of the CNTF.
(4) Collecting CN techniques motivated by use cases and adding them to the CNTF.
(5) Provide algorithms for flexible CN and inject the implementations into an implementation test space of CNTF.

After conducting a thorough state of the art, we opted to create a web application that supports the categorisation and comparative evaluation of CN features. We collected use cases, and identified the styles, dimensions and requirements of these use cases.

The flexible CN techniques will first be evaluated using a synthetic dataset and profiles, followed by real data for future work. Performance and scalability will be measured using response time for a set of predefined graphs and data profiles. Response time will also be measured by filling the data graphs with more and more triples, in order to compare performance with an increasing number of triples (e.g. 500, 1000, 1500 triples). The same measurement technique will be applied for an increasing number of constraints in the profiles by adding shapes in the SHACL case.

## 6 RESULTS

### 6.1 CNTF Web Application

A web application has been developed[2], which at the time of writing classifies CN into categories, a set of CN use cases, styles, dimensions have been collected and added to the resource and linked to each others, please refer to the *updates* section to perceive the list of features already added and those to come.

### 6.2 A bottom-up formalisation of CN

A first version of the formalisation of document CN has been done and will be added to the CNTF eventually. On the server side, a URI is associated with documents having each a quality value. To illustrate, consider the scenario in which a server has two representations of a resource $d_1$ in Turtle with a preference of 0.7 and $d_2$ in plain text with a preference of 0.9. A client wants to request a Turtle representation. From the the set of all documents $\mathcal{D}$, we have $d_1$ and $d_2$. The preference of the client for a document is indicated by a quality value $q$ in the range $[0, 1]$[3]. We define the set $C$ of preferences as:

$$c \in C \mid c : \mathcal{D} \rightarrow [0, 1] \tag{1}$$

In practice, a client typically assigns a q-value to a *type* of documents. This can be modelled as a preference $c$ such that $c(d) = q$ for

[2]https://ci.mines-stetienne.fr/cntf

[3]In RFC7231 [8, Section 5.3.1]. 0 means not acceptable, 0.001 is the least preferred and 1 is the most preferred

all documents $d$ of the requested type, and $c(d) = 0$ for all the others. E.g., if a request has the header `accept: text/turtle;q=0.9`, this can be translated into the preference $c(d) = 0.9$ for all Turtle documents, and $c(d) = 0$ for all others.

We model the Web as a function from the set of IRIs $\mathcal{U}$ to the set of constraints $C$, formally:

$$\mathcal{W} : \mathcal{U} \to C$$

The server from our scenario can serve two documents:

$$\mathcal{W}(u) : \begin{cases} d_1 \mapsto 0.7 \\ d_2 \mapsto 0.9 \\ \mathcal{D} - \{d_1, d_2\} \to 0 \end{cases} \tag{2}$$

We model the response as a document, in reality the response contains additional information such as headers. In our model, *RES* is a function defined as:

$$RES : \mathcal{REQ} \to \mathcal{D} \tag{3}$$

The response depends on the strategy of the server. Let us consider a request from a client to an IRI $u$ and with a constraint $c_1$ defined as $\langle u, c_1 \rangle \in \mathcal{REQ}$. And a server that can serve a response from the requested IRI $u$ but that has also a constraint $c_2$, so we can write $c_2 = \mathcal{W}(u)$. We define three possible responses:

$$RES_s : \langle u, c_1 \rangle \to argmax(\{\mathcal{W}(u)(d) | d \in \mathcal{D} \wedge \mathcal{W}(u)(d) \neq 0\})$$

Using the $RES_s$ function, the server serves the response that maximises its constraints, not taking into consideration the client's constraints.

$$RES_c : \langle u, c_1 \rangle \to argmax(\{c_1(d) | d \in \mathcal{D} \wedge c_1(d) \neq 0 \wedge \mathcal{W}(u)(d) \neq 0\})$$

Using the $RES_c$ function, the server serves the response that maximises the constraints of the client. i.e., $RES_{noAnswer} = \emptyset$

The server does not send any response if there is a conflict between its constraints and the client constraints.

### 6.3 A step towards semantic CN

Another result worth mentioning is a functional demonstration of CN using profiles that takes the form of SHACL documents. The implementation was done using Java, and Spring Framework to handle queries and intercept query headers. Jena Framework was used to handle RDF graphs and SHACL document and validation. Once we have validated the scalability of the results, we will add the implementation to CNTF.

## 7 CONCLUSIONS AND FUTURE WORK

CN is very important and is a fundamental mechanism of the Web. The benefits and necessity to leverage CN are highlighted by the Spatial Data on the Web [29] working group, as well as the Web Data Best Practices [6] working group in their best practices documents.

In this paper, we analyse the state of existing work in CN and point to a new direction: the use of validation languages to express finer constraints. We believe that in real-world applications, these fine-grained constraints, when explored appropriately, would make the negotiation process much more flexible. As we have proposed the CNTF resource, the general formalisation of the negotiation process, and an algorithm for constraint negotiation using SHACL. Our future work will focus on enriching CNTF with more use cases and finalising the features to satisfy the above mentioned requirements, as well as extending the formalisation to include negotiation of specific documents such as RDF graphs using profile negotiation. The results of this study can be used to validate and assert that SHACL can be used to introduce flexibility into the process of choosing the best representation.

We plan to follow these steps for our future work:

(1) Continue to evaluate and extend profile negotiation.
(2) Enrich the formalisation to include proposed solutions.
(3) Investigate constraint containment more in depth [1, 18, 22, 25], then proposes an algorithm taking advantage of it.
(4) Take a real world use case of a portal like Europeana[4] that serves content from different sources.
(5) Explore the use case of web of things/hypermedia CN in a smart building environment.
(6) Examine content adaptation [20].

## REFERENCES


[1] A. Abbas, P. Genevès, C. Roisin, and N. Layaïda. 2017. SPARQL Query Containment with ShEx Constraints. In *Proc. 21st European Conference on Advances in Databases and Information Systems (LNCS, Vol. 10509)*. Springer, 343–356.
[2] R. Atkinson and N. Car. 2019. *The Profiles Vocabulary, W3C Working Group Note 18 December 2019*. W3C Working Group Note. W3C.
[3] T. Berners-Lee, R. Cailliau, J. Groff, and B. Pollermann. 1992. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy* 2, 1 (1992), 74–82.
[4] N. Choudhury. 2014. World Wide Web and Its Journey from Web 1.0 to Web 4.0. *Int. Journal of Comp. Sci. and Information Tech.* 5, 6 (2014), 8096–8100.
[5] R. Cyganiak, D. Wood, and M. Lanthaler. 2014. *RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014*. W3C Recommendation. W3C.
[6] B. Farias Lóscio, C. Burle, and N. Calegari. 2017. *Data on the Web Best Practices, W3C Recommendation 31 January 2017*. W3C Recommendation. W3C.
[7] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, L. Masinter, P. Leach, and T. Berners-Lee. 1999. *Hypertext Transfer Protocol - HTTP/1.1*. RFC 2616. IETF.
[8] R. Fielding and J. Reschke. 2014. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. RFC 7231. IETF.
[9] R. Fisher, W. Ury, and B. Patton. 2011. *Getting to yes: Negotiating agreement without giving in*. Penguin.
[10] I. Grigorik and Y. Weiss. 2021. *HTTP Client Hints*. RFC 8942. IETF.
[11] W3C OWL Working Group. 2012. *OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11 December 2012*. W3C Recommendation. W3C.
[12] P. Hayes and P. Patel-Schneider. 2014. *RDF 1.1 Semantics, W3C Recommendation 25 February 2014*. W3C Recommendation. W3C.
[13] K. Holtman and A. Mutz. 1998. *Transparent Content Negotiation in HTTP*. Technical Report. IETF.
[14] I. Jacobs and N. Walsh. 2004. *Architecture of the World Wide Web, Volume One, W3C Recommendation 15 December 2004*. W3C Recommendation. W3C.
[15] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. Butler, and L. Tran. 2004. *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation 15 February 2004*. W3C Recommendation. W3C.
[16] H. Knublauch and D. Kontokostas. 2017. *Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017*. W3C Recommendation. W3C.
[17] M. Lefrançois. 2018. RDF presentation and correct content conveyance for legacy services and the web of things. In *Proceedings of the 8th International Conference on the Internet of Things, IOT 2018, Santa Barbara, CA, USA, October 15-18, 2018*. ACM Press, 43:1–43:8.
[18] M. Leinberger, P. Seifer, T. Rienstra, R. Lämmel, and S. Staab. 2020. Deciding SHACL Shape Containment Through Description Logics Reasoning. In *The Semantic Web - ISWC 2020 - 19th ISWC, Athens, Greece, November 2-6, 2020, Proceedings, Part I (LNCS, Vol. 12506)*. Springer, 366–383.
[19] S. Lerouge. 2006. *Personalizing quality aspects for video communication in constrained heterogeneous environments*. Ph.D. Dissertation. Ghent University.
[20] W. Lum and F. Lau. 2003. User-Centric Content Negotiation for Effective Adaptation Service in Mobile Computing. *IEEE Trans. on Soft. Eng.* 29, 12 (2003), 1100–1111.
[21] Open Mobile Alliance. 2003. *User Agent Profile*. Technical Report. Open Mobile Alliance.


[4] https://www.europeana.eu/

[22] P. Pareti, G. Konstantinidis, F. Mogavero, and T. Norman. 2020. SHACL Satisfiability and Containment. In *The Semantic Web - ISWC 2020 - 19th ISWC, Athens, Greece, November 2-6, 2020, Proceedings, Part I (LNCS, Vol. 12506)*. Springer, 474–493.
[23] E. Prud'hommeaux, I. Boneva, J. Labra Gayo, and G. Kellogg. 2019. *Shape Expressions Language 2.1, Final Community Group Report 8 October 2019.* W3C Community Group Report. W3C.
[24] J. Snell. 2014. *Prefer Header for HTTP.* RFC 7240. IETF.
[25] S. Staworko and P. Wieczorek. 2019. Containment of Shape Expression Schemas for RDF. In *Proc. 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems.* ACM Press, 303–319.
[26] L. Svensson. 2016. An http Header for Metadata Schema Negotiation. In *W3C Workshop on Smart Descriptions & Smarter Vocabularies (SDSVoc).* W3C, 5 pages.
[27] L. Svensson, R. Atkinson, and N. Car. 2019. *Content Negotiation by Profile, W3C Working Draft 26 November 2019.* W3C Working Draft. W3C.
[28] L. Svensson, R. Verborgh, and H. Van de Sompel. 2021. *Indicating, Discovering, Negotiating, and Writing Profiled Representations.* Internet draft. IETF.
[29] L. van den Brink, P. Barnaghi, J. Tandy, G. Atemezing, R. Atkinson, B. Cochrane, Y. Fathy, R. García-Castro, A. Haller, A. Harth, K. Janowicz, S. Kolozali, B. van Leeuwen, M. Lefrançois, J. Lieberman, A. Perego, D. Le Phuoc, B. Roberts, K. Taylor, and R. Troncy. 2019. Best practices for publishing, retrieving, and using spatial data on the web. *Semantic Web Journal* 10, 1 (2019), 95–114.