A Step Toward Semantic Content Negotiation

Yousouf Taghzouti[1,*], Danai Vachtsevanou[2], Simon Mayer[2] and Andrei Ciortea[2]

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

[2]Institute of Computer Science, University of St. Gallen, Rosenbergstrasse 30, 9000 St. Gallen, Switzerland.

Abstract

Content negotiation aims at enabling a server to provide a client with a representation of a resource that meets its needs. However, client and server might desire to negotiate constraints that go beyond the media type or language of the alternative representation. This is especially true in the Semantic Web, as a resource can be described with a single media type, but with different vocabularies (FOAF, schema.org, etc.), and may match specific patterns. In this paper, we propose an approach to increase the flexibility when negotiating a representation between client and server. Our approach follows the goals of the World Wide Web and uses a set of existing technologies: SHACL and profile-based negotiation. We define the mechanism (in terms of protocol and algorithm) for clients to announce their expectations and for servers to react and respond to them. We then explain, through a use case, how the same approach could be used in Web-based Multi-Agent Systems to help autonomous agents achieve their goals on the Web.

Keywords

Content negotiation, Multi-Agent Systems, SHACL, RDF, HTTP

1. Introduction

A resource on the Web is identified by a Uniform Resource Identifier (URI) and may have different alternative representations [1, Section 3.2]. Alternative representations may differ on several dimensions, such as media type or language. The purpose of *Content Negotiation* (CN) is that clients are enabled to request resource representations that they are able to understand, and successful CN provides a foundation for the client/server contract: If a client requested a specific representation and the server provides it, they have successfully established a common basis for their communication. CN plays a central role on the Web, and has done so since its earliest days, since the negotiation layer is a key component of the Web architecture [2]. The Hypertext Transfer Protocol (HTTP) comes with a set of headers that enable CN, the most common being *accept* for media type negotiation, and *accept-language* for language negotiation [1, Section

EKAW-C 2022: Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management

*Corresponding author.

yousouf.taghzouti@emse.fr (Y. Taghzouti); danai.vachtsevanou@unisg.ch (D. Vachtsevanou); simon.mayer@unisg.ch (S. Mayer); andrei.ciortea@unisg.ch (A. Ciortea)

https://youctagh.github.io/ (Y. Taghzouti); https://www.alexandria.unisg.ch/persons/8447/ (D. Vachtsevanou); https://www.alexandria.unisg.ch/persons/7778 (S. Mayer); http://andreiciortea.ro/ (A. Ciortea)

0000-0003-4509-9537 (Y. Taghzouti); 0000-0002-6697-0427 (D. Vachtsevanou); 0000-0001-6367-3454 (S. Mayer)

5.3]. The CN process consists of several stages that together contribute to the success of the mechanism:

- **Discovery:** In the first stage, the interest is in how a client discovers the resource to be negotiated and how the server announces its availability.
- **Request Formulation:** In this stage, the client communicates to the server the requested resource as well as the constraints and preferences according to a given set of dimensions.
- **Selection/Adaptation:** Next, the server selects an alternative representation and/or adapts it to fit the client's constraints.
- **Response Indication:** The server indicates the choice of representation and sends it to the client.
- **Response Interpretation:** Finally, the client interprets the received response and either continues or stops the negotiation process.

CN is also invaluable on the Semantic Web: Here, RDF is used for describing resources, and it is common to find different representations of a resource where each might utilise a set of vocabularies and conform to specific shapes. RDF representations can be serialised into several common serialisation formats (e.g., Turtle or JSON-LD), and hence CN may be used in the media type dimension.

However, beyond media types, clients might want to request a representation that uses specific vocabularies and conforms to specific patterns. This need for finer-grained CN goes beyond simply negotiating the media type and language, and comes with some challenges [3].

The *profile vocabulary* has been introduced as a novel way to define validation rules or semantic interpretation for representations [4]. A new header *accept-profile* has been introduced [5] to allow the client to pass its preference regarding a profile to the server. A *profile resource* may take many forms, where in this work we focus on SHACL documents (i.e., shapes graphs) – SHACL and ShEx are languages used to describe and validate RDF data graphs against a set of conditions [6, 7]. It is hence natural to use SHACL to convey a set of conditions to be validated by the returned representation during Semantic CN. We selected SHACL because of its broad acceptance as a W3C recommendation and because it defines the *sh:severity* property, which is instrumental for our purposes.

In this paper, we present a conformance-based approach to CN in the profile dimension. The alternative representations used are RDF sources, and SHACL documents are used to convey client constraints to be validated. In the same way as for traditional CN, we use HTTP headers as a means of communicating the to-be-negotiated dimension. In the following, we present our CN approach based on semantic validation, which allows the negotiation of variants (that use a set of vocabularies; that best validate a set of constraints) while taking into account client preferences for constraint priority. Then, we discuss how the same approach could be used in Web environments for Multi-Agent Systems to help (artificial) agents achieve their goals.

2. Content Negotiation based on Semantic Validation

While it is common to use quality parameters in HTTP requests to assign a relative preference to CN header fields to convey precedence [1, Section 12.4.2], using it in the profile dimension would give the ability to assign precedence to the entire SHACL document, which would be useful but not sufficient if one wants to express the order of parts of the shapes graph or to convey the severity of certain shape constraints (to convey that one would tolerate non-conformance with certain parts of the shapes graph).

The server may send an HTTP 406 (Not Acceptable) response code if it could not produce a representation matching the full list of acceptable constraints, e.g., acceptable media type [1, Section 15.5.7], alternatively a default representation is returned. While a general default representation is an option, in our approach we propose to respond with an available representation that is close to the requested one by calculating a violation rate score. This score takes into account the precedence expressed by the client through the severity property.

Pre-processing vs on-the-fly conformance checking of representations A server can have a fixed number of profiles it serves and goes through a pre-processing phase where it matches the available representations to the profiles to be served, for example: rep1 $\mapsto$ http://example.com/prof1 and rep2 $\mapsto$ http://example.com/prof2

When a client requests a representation conforming to, for example, http://example.com/prof1, the server searches for a corresponding representation and provides *res1*. Thus, the client must know the exact URI to identify the required profile. In the case of SHACL, even if the request includes the same patterns, e.g., http://example.com/prof1, but collected in a document identified by a different URI, e.g., http://example.com/my-custom-prof-uri, the server will not be able to provide the representation since the selection is performed by matching it with the available entries. However, if the selection strategy is *on-the-fly conformance*, checking the conformance of the available resources to the requested profile will result in the delivery of the representation *rep1*. One can think of a *hybrid* solution where the most commonly requested profiles are pre-processed and, in the selection process before applying the conformance checks checked if one of the entries matches the desired profile.

Full (strict) vs Partial (flexible) conformance When a client sends a list of constraints to the server and only one resource is to be returned, several scenarios can occur: all constraints are satisfied; some of the constraints are satisfied and some are not; none of the constraints are satisfied. The response strategies for each of these scenarios can be defined: at the server side, at the client side, or a combination of both. The behaviour could be to return an error message, typically a *406 Not Acceptable*, if some of the constraints are not met, or to return a default representation. This behavior is observable in the Apache HTTP server implementation[1]; for example, when considering constraints in different dimensions, such as media type and language. Apache's *ForceLanguagePriority Prefer* could be viewed as maintaining a strict conformance, while *ForceLanguagePriority Fallback* used with a *LanguagePriority* list provides a fallback strategy making it more flexible.

It is difficult to have a predefined default representation that fits all scenarios. Especially in the Semantic Web, if one considers defining profiles like SHACL documents for example, since there could be a very large number of possibilities (combination of shape constraints). We propose to have an algorithm to compute this fallback representation by serving a representation that

[1] https://httpd.apache.org/docs/current/content-negotiation.html

conforms to as many constraints as possible, making it as close as possible to the requested one and better than a default unrelated representation. Moreover, if this fallback is triggered, this must be made visible to the client. To help in this direction, the client would express its desire for such behaviour through the use of the *accept-conformance: strict | flexible* header, while using the SHACL property *sh:severity* with the values *sh:Violation*, *sh:Warning* or *sh:Info* in the definition of the SHACL document which would be used to indicate that the client expects a partial conformance strategy.

Constraint precedence (priority, ordering) In the case of using HTTP headers as a means of conveying the constraints, the quality value "q-values" [1, Section 12.4.2] could be used to assign a relative weight to the preference for that associated content and express an order to the constraints across dimensions, (e.g., media types or languages, etc). The same technique could be used to order profiles [8], but a profile may be composed of multiple shapes, and a client may want to express a finer ordering. Similar to how we use *sh:severity* for conformance checking, we rely on using SHACL semantics of the *sh:severity* property, with the assumption that the priority *violation* > *warning* > *info*. This approach allows us to use this property to order and prioritise constraints on a higher granularity, in addition to the scoring of the alternative representation to be served.

3. Use Cases and Preliminary Implementation

In this section, we present two use cases and a preliminary implementation of our approach. The first use case illustrates CN in the profile dimension between a client and a server holding several representations of a target resource. In the negotiation, the client requests a representation conforming to a profile defined as a SHACL document. The second use case presents an application of our work to Web-based Multi-Agent Systems: We illustrate how profile-based CN can help autonomous agents achieve their goals in a more flexible manner.

3.1. Traditional content negotiation on the Web

Negotiation of profiles as 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 provided by the university portal, and these requirement are to be satisfied:

R1 - Shape importance: John needs a representation that conforms to specific shapes. Therefore, negotiating the vocabulary is not enough since he has to manually validate all the returned data graphs to verify that they conform to the desired profile.

R2 - Flexibility tolerance: In R1, 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.

R3 - Constraints inequality: 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.

Preliminary implementation To solve this use cases and achieve flexible CN in a practical way, SHACL was used. Specifically, the recently introduced header *accept-profile* is used to request a representation that validates a set of constraints in the form of SHACL documents [9]. 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 is returned depending on the server configuration.[2] 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. A functional demonstration of CN using profiles that takes the form of SHACL documents was developed. 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[3].

Concrete Incremental Example To illustrate our approach and show the improvements it would bring, we take a concrete example. All the Web resources used are available on Git[4]. In what follows we will use the Git URI as the base URI.

Let us assume that we have a resource at ./yousouf with different representations. A text: in English ./yousouf.en.txt containing the text: "Yousouf is a PhD student at MINE Saint-Étienne", and in French ./yousouf.fr.txt containing "Yousouf est un doctorant à MINE Saint-Étienne". We also have a JSON representation Listing 1, and also two RDF representations: one serialised in Turtle Listing 2, and the other as XML Listing 3.

Listing 1: A JSON representation

```
{ "name":"Yousouf", "occupation":"PhD student", "place":"MINE Saint- tienne " }
```

Listing 2: A Turtle representation

```
...
ex:yousouf a ex:Person;  ex:hasName "Yousouf"@en;
    ex:hasOccupation "PhD Student"@en;  ex:worksIn "MINE Saint-Etienne"@en .
```

[2]Multiple discussions within the Data Exchange Working Group to address these issues, e.g. https://github.com/w3c/dx-connegp/issues/5
[3]https://github.com/YoucTagh/flexible-cn
[4]https://github.com/YoucTagh/flexible-cn/blob/main/local-resources/

Listing 3: An XML representation

```
...
<rdf:Description rdf:about="http://example.org#yousouf">
    <rdf:type rdf:resource="http://example.org#Person"/>
    <ex:hasName xml:lang="en">Yousouf</ex:hasName>
    <ex:hasOccupation xml:lang="en">PhD Student</ex:hasOccupation>
    <ex:worksIn xml:lang="en">MINE Saint-Etienne</ex:worksIn>
</rdf:Description> ...
```

If a user has some preferences and wants to negotiate in the media type dimension, the HTTP *accept* header could be used with the values of the appropriate media types[5], in our example the value could take for example *text/plain, application/rdf+xml, text/turtle, application/json.* or in the language dimension by means of the *accept-language* header with the value of the appropriate language tag e.g. *en* or *fr*. In the Semantic Web, this approach is sufficient if one only wants to negotiate RDF serialization. However, a resource can be described with different vocabularies yet serialised in the same format, e.g. ./yousouf.ttl as shown in Listing 2 and ./yousouf_other_vocab.ttl in Listing 4.

Listing 4: Another Turtle representation with different vocabularies

```
...
ex:yousouf a foaf:Agent;  foaf:name "Yousouf"@en;
    dc:type "Worker";  dc:title "PhD student";  org:memberOf ex:emse .
ex:emse a org:Organisation;  skos:prefLabel "MINE Saint-Etienne".
```

A user may want a response that validates a set of rules, e.g., the person's name must use the *foaf:name* property or the organisation of which that person is a member must use the Organisation vocabulary and the *skos:prefLabel* property must be specified. Our proposal for semantic CN takes advantage of validation languages, e.g., SHACL, to describe the required rules as shapes. The corresponding SHACL shape document is presented in Listing 5.

Listing 5: An example of a SHACL shape document

```
...
ex:PersonShape a sh:NodeShape ; sh:targetClass foaf:Agent ;
   sh:property [  sh:path foaf:name;  sh:minCount 1;  sh:datatype xsd:string; ].
```

In this approach, when requesting the resource, the user uses the *accept-profile* header and indicates the URI of the SHACL document as a value, e.g. ./yousouf-preferred-shape.ttl, which allows the server to engage in semantic CN taking into account the finer preferences specified by the user and proposing a more precise response.

3.2. Web-based Multi-Agent Systems Use Case

The advantages of (semantic) CN can be highlighted in systems where (i) clients may have diverse preferences of variants, (ii) servers cannot hold a priori knowledge about client preferences, (iii) clients cannot hold a priori knowledge about variants, and (iv) effective negotiation of variants severely impacts system efficiency. To this end, we use Hypermedia Multi-Agent Systems (Hypermedia MAS) [10] – a type of Web-based Multi-Agent Systems – as a basis

[5] https://www.iana.org/assignments/media-types/media-types.xhtml

for discussing the applicability of our approach to systems that exhibit the above-mentioned properties. Specifically, we present a use case where the functioning of a Signifier Exposure Mechanism [11] in Hypermedia MAS can depend on CN based on semantic validation.

A *Signifier Exposure Mechanism (SEM)* manages the dynamic exposure of signifiers in a MAS environment — where signifiers are cues revealing information to autonomous agents about how to interact with resources in their environment[6]. At run time, agents can look up signifiers through the SEM towards receiving information about how to interact in a manner that better fits their context and abilities. In this case, the SEM consists of a SHACL processor that validates RDF data representing signifiers. The validation of signifiers is based on SHACL documents that capture constraints set by autonomous agents, and whose location is specified by agents in the *accept-profile* header upon request. For example, constraints may relate to an agent's employed methods for reasoning about action, or its current goals. As a result, such constraints can be very dynamic, heterogeneous, and unexpected to the SEM designer. For this, signifier relevance, and consequently, the probability of signifier exposure cannot be by default set to 0 in the absence of strict conformance, but it should still increase for validation measures with better scores. By enabling the SEM to calculate validation scores based on the available signifiers and the specified profile, the signifier that conforms to the agent's context and abilities the most is considered to be the one that is the most relevant to the agent based on the current agent-environment situation, thus it is the one exposed (returned in response) to the agent.

To explain how our approach could be applied in a Web-based Multi-Agent System, we adjust the scenario of SEM usage presented in [11] towards enabling signifier exposure based on semantic validation. Based on the scenario, a Hypermedia MAS features a 3x3 maze grid – an environmental entity with which agents can interact by moving across the rooms of the maze. An SEM is deployed in the environment for modulating which signifiers are exposed to an agent in every system state, taking into consideration the complete set of available signifiers (and their variants) and the agent's constraints. he scenario phases are adjusted as follows:

1. An agent is registered to the maze in room 1, and has the objective of moving to room 9.
2. The agent publishes a SHACL document that describes the shape of signifiers that are relevant to the agent's current state and objective. Specifically, signifiers are considered relevant if they reveal information about possible interactions for agents that a) are positioned in a specific room of the maze (initially, in room 1), and b) have the objective of moving to room 9.
3. For retrieving information about how to interact with the maze, the agent requests from the SEM a new set of signifiers, and adds the URL of the SHACL document in the *accept-profile* header.
4. The SEM retrieves the SHACL document that is identified by the provided URL, and computes a validation measure for every pair formed by the shape document and an object from the set of available signifiers.
5. The SEM returns to the agent the URL of the signifier with the best score, i.e. the one that conforms the most to the agent's current context. Initially, the SEM returns a signifier that describes how the agent can move to room 2 of the maze.

[6]For example, signifiers exposed in the environment of a Hypermedia MAS describe the hypermedia controls that can be applied to Web resources [11].

6. The agent retrieves and interprets the exposed signifier identified by the URL, and proceeds to interact with the maze based on the signifier, i.e. the agent moves to room 2.
7. Steps 2-6 are repeated until the agent decides that its objective has been achieved, i.e. the agent has moved to room 9.

4. Conclusion and perspective

This article provides an overview of how basic Web technologies can be used or enhanced to provide more meaningful and useful services. It also relates high-level use cases to low-level implementation details. We presented our initial investigation into the use of semantic validation to improve the flexibility of CN. The results are very encouraging even though the representations and shapes graphs were small and synthetic. We plan to try this approach in the MAS use case described above with real data and shapes graphs in order to test its applicability.

Acknowledgments

We would like to thank Prof. Antoine Zimmermann, Prof. Maxime Lefrançois and Prof. Ruben Verborgh for their valuable feedback and support. This research was partially funded by the Swiss National Science Foundation under grant No. 189474 (HyperAgents), and the research support program SeReCo of the Franco-German University.

References

[1] R. T. Fielding, M. Nottingham, J. F. Reschke, HTTP Semantics, RFC 9110, IETF, 2022.
[2] T. Berners-Lee, R. Cailliau, J. Groff, B. Pollermann, World-Wide Web: The Information Universe, ENRAP 2 (1992) 74–82.
[3] R. Verborgh, Your JSON is not my JSON – A case for more fine-grained content negotiation, in: Proceedings of the Workshop on Smart Descriptions & Smarter Vocabularies, 2016.
[4] R. Atkinson, N. Car, The Profiles Vocabulary, W3C Working Group Note, W3C, 2019.
[5] L. Svensson, R. Verborgh, H. V. de Sompel, Indicating, Discovering, Negotiating, and Writing Profiled Representations, Internet-Draft, Internet Engineering Task Force, 2021.
[6] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3C Recommendation, W3C, 2017.
[7] E. Prud'hommeaux, I. Boneva, J. Labra Gayo, G. Kellogg, Shape Expressions Language 2.1, W3C Community Group Report, W3C, 2019.
[8] L. Svensson, An http Header for Metadata Schema Negotiation, in: W3C Workshop on Smart Descriptions & Smarter Vocabularies (SDSVoc), W3C, 2016.
[9] L. Svensson, R. Atkinson, N. Car, Content Negotiation by Profile, W3C Working Draft 26 November 2019, W3C Working Draft, W3C, 2019.
[10] A. Ciortea, O. Boissier, A. Ricci, Engineering world-wide multi-agent systems with hypermedia, in: International Workshop on Engineering Multi-Agent Systems, 2018.
[11] J. Lemee, D. Vachtsevanou, S. Mayer, A. Ciortea, Signifiers for affordance-driven multi-agent systems, in: International Workshop on Engineering Multi-Agent Systems, 2022.