# WShEx: A language to describe and validate Wikibase entities

Jose Emilio Labra-Gayo[1]

[1] WESO research group - University of Oviedo

Abstract
Wikidata is one of the most successful Semantic Web projects. Its underlying Wikibase data model departs from RDF with the inclusion of several features like qualifiers and references, built-in datatypes, etc. Those features are serialized to RDF for content negotiation, RDF dumps and in the SPARQL endpoint. Wikidata adopted the entity schemas namespace using the ShEx language to describe and validate the RDF serialization of Wikidata entities. In this paper we propose WShEx, a language inspired by ShEx that directly supports the Wikibase data model and can be used to describe and validate Wikibase entities. The paper presents the abstract syntax and formal semantics of the WShEx language.

## 1. Introduction

Wikidata has become one of the biggest projects which collect human knowledge in the form of linked data according to the Semantic Web view. It is collaboratively maintained both by humans and bots, which update the contents from external services or databases. The set of open source tools which run Wikidata is called Wikibase [1] and it makes possible to create other knowledge graphs following the same data model as Wikidata but with different content and purposes. The projects that are using Wikibase are called Wikibase instances and in this paper we will call the style of knowledge graphs obtained using Wikibase instances as Wikibase graphs.

Wikibase was initially created as a set of MediaWiki extensions which facilitated adoption by the Wikimedia community. Internally, Wikidata content is managed by a relational database [1]. In order to facilitate data analysis and querying, as well as integrating Wikidata within the semantic web ecosystem, it adopted a triple store [2] whose contents can be retrieved as RDF data according to the linked data principles and exported as an SPARQL endpoint through the Wikidata query service.

In 2019, Wikidata created a new namespace for entity schemas which can be used to describe and validate the RDF serialization of Wikidata entities using ShEx [2] Entity schemas offer a concise language to describe Wikibase entities. Users can create new schemas for different

*Wikidata'22: Wikidata workshop at ISWC 2022*
labra@uniovi.es (J. E. Labra-Gayo)
http://labra.weso.es (J. E. Labra-Gayo)
0000-0001-8907-5348 (J. E. Labra-Gayo)


[1] https://wikiba.se/

[2] The current technology is Blazegraphhttps://blazegraph.com/ but there are plans to replace it

purposes and there are ShEx-based tools that can be used to check if entities conform to entity schemas or visualize entity schemas. At the time of this writing there are more than 370 entity schemas created[3] but there are no evidences about their general adoption as part of the mainstream workflow employed by Wikidata users. Although there may be several reasons for this like the lack of better tool support, one aspect that can also affect this situation is that entity schemas describe the RDF serialization of entities, instead of their underlying Wikibase data model. This aspect makes entity schemas a bit more verbose and aggravates their usability. In this paper, we propose a new language called WShEx which is inspired by ShEx and can be used to directly describe and validate entities based on the Wikibase data model.

The first motivation for the development of WShEx was to create subsets of Wikidata in different domains using a concise and human-readable language. In order to process JSON-based Wikidata dumps, it was not possible to directly use entity schemas which describe the RDF serialization, so we developed WShEx, a language similar to ShEx that could be used to describe Wikibase data models directly. Some parts of this paper have been extracted from [3], a larger paper where we also describe the different subsetting techniques employed.

## 2. Wikibase data model

The Wikibase data model[4] is an abstract data model that can have different serializations like JSON and RDF. It is defined using UML data structures and a notation called Wikidata Object Notation.

Informally, the data model is formed from entities and statements about those entities. There are two main types of entities: item and properties[5]. An item is identified using a `Q` followed by a number and can represent any thing like an abstract of concrete concept. For example, Q80 represents Tim Berners-Lee in Wikidata. A property is identified by a `P` followed by a number and represents a relationship between an item and a value. For example, P19 represents the property *place of birth* in Wikidata. The values that can be associated to a property are constrained to belong to some specific datatype. There can be compound datatypes like geographical coordinates. Some of Wikibase datatypes are: quantities, dates and times, geographic locations and shapes, monolingual and multilingual texts, etc.

A statement consists of:

- A property which is usually denoted using a `P` followed by a number.
- A declaration about the possible value (in Wikibase terms, it is called a *snak*) which can be a specific value, a `no value` declaration or a `some value` declaration.
- A rank declaration which can be either `preferred`, `normal` or `deprecated`.
- Zero or more qualifiers which consist of a list of property-value pairs
- Zero or more references which consist of a list of property-value pairs.

[3] A directory for entity schemas can be seen at https://www.wikidata.org/wiki/Wikidata:Database_reports/EntitySchema_directory

[4] https://www.mediawiki.org/wiki/Wikibase/DataModel

[5] There are other types of entities like Lexemes which we omit for brevity

We define a formal model for Wikibase which is inspired from Multi-Attributed Relational Structures (MARS) [4]. For simplicity, we model only qualifiers omitting references, which could be represented in a similar way, don't handle the no-value and some-value snaks, and represent only one primititve data value $D$.

**Definition 1** (Wikibase graphs)**.** Given a mutually disjoint set of items $Q$, a set of properties $P$ and a set of data values $D$, a *Wikibase graph* is a tuple $\langle Q, P, D, \rho \rangle$ such that $\rho \subseteq E \times P \times V \times FinSet(P \times V)$ where $E = Q \cup P$ is the set of entities which can be subjects of a statement and $V = E \cup D$ is the set of possible values of a property.

In practice, Wikibase graphs also add the constraint that every item $q \in Q$ (or property $p \in P$) has a unique integer identifier $q^i \in \mathbb{N}$ ($p^i \in \mathbb{N}$). In the Wikibase data model, statements contain a list of property-values and the values can themselves be nodes from the graph. This is different from property graphs, where the set of vertices and the set of values are disjoint.

**Example 1** (Wikibase graph example)**.** We focus on a subset of Wikidata that models information about Tim Berners-lee (*timBl*, Q80) and some of its awards. *timBl* is an *instanceOf* (P31) *Human* (Q5). His *birthPlace* (P19) was *London* (Q84), whose *country* (P27) is *UK* (Q145) and his *birthDate* (P569) was 1955. He has as *employer* (P108) the *CERN*twice. The first one has *start* date (P580) 1980 and *end* date (P582) 1980, and the second one between 1984 and 1994. He was *awarded* (P166) with the *Princess of Asturias* (*PA*) award *togetherWith* (P1706) Vinton Cerf (*vintCerf*) [6], the award was given at the point in time (*pointTime*, P585) 2002. The *country* of that award is *Spain*and Vinton Cerf was born in New Haven (*NewHaven*):

$$
\begin{aligned}
Q &= \{ \; timBl, vintCerf, London, CERN, UK, Spain, PA, Human\} \\
P &= \{ \; birthDate, birthPlace, country, employer, awarded, \\
&\qquad start, end, pointTime, togetherWith, instanceOf\} \\
D &= \{ \; 1984, 1994, 1980, 1955\} \\
\rho &= \{ \; (timBl, instanceOf, Human, \{\}), \\
&\qquad (timBl, birthDate, 1955, \{\}), \\
&\qquad (timBl, birthPlace, London, \{\}), \\
&\qquad (timBl, employer, CERN, \{ \; start{:}1980, end{:}1980 \; \}), \\
&\qquad (timBl, employer, CERN, \{ \; start{:}1984, end{:}1994 \; \}), \\
&\qquad (timBl, awarded, PA, \{pointTime{:} \; 2002, togetherWith{:}vintCerf\}), \\
&\qquad (London, country, UK, \{\}), \\
&\qquad (vintCerf, instanceOf, Human, \{\}) \\
&\qquad (vintCerf, birthPlace, NewHaven, \{\}) \\
&\qquad (CERN, awarded, PA, \{ \; pointTime{:} \; 2013 \; \}) \\
&\qquad (PA, country, Spain, \{ \; \}) \; \}
\end{aligned}
$$

Figure 1 presents a possible visualization of the example Wikibase graph.

The Wikibase data model supports 2 main export formats: JSON and RDF. The JSON serialization directly follows the Wikibase data model. It basically consists of an array of entities where each entity is a JSON object that captures all the local information about the entity: the labels,

[6] The award was really obtained by Tim Berners-Lee, Vinton Cerf, Robert Kahn and Lawrence Roberts, we included here only the first two for simplicity

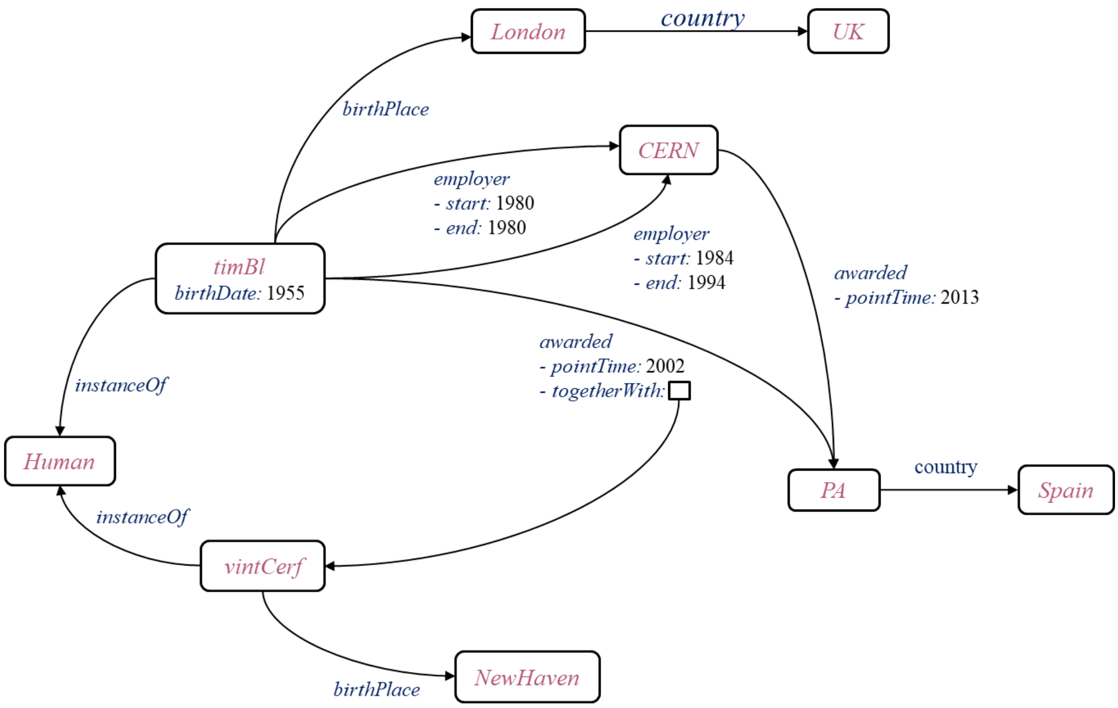


**Figure 1:** Visualization of example Wikibase graph

descriptions, aliases, sitelinks and statements that have the entity as subject. Each JSON object is represented in a single line. A remarkable feature of this encoding is that it captures the output neighborhood of every entity in a single line making it amenable to processing models that focus on local neighborhoods.

The RDF serialization[7] was designed with the goal of being able to represent all the structures of the Wikibase data model in RDF, maintaining compatibility with semantic web vocabularies like RDFS and OWL and avoiding the use of blank nodes [5].

**Example 2** (RDF serialization of a node)**.** As an example, a fragment of the information about Tim Berners-Lee that declares that he is an instance of Human, has birth place London, has birth date 1955 and has employer with value CERN between 1984 and 1994 is represented in RDF (Turtle) [8] as:

```
wd:Q80 rdf:type wikibase:Item ;
 wdt:P31 wd:Q5 ; # instance of = Human
 wdt:P19 wd:Q84 ; # birthplace = London
 wdt:P569 "1955-06-08T00:00:00Z"^^xsd:dateTime ; # birthDate
 wdt:P108 wd:Q42944 ; # employer = CERN
 p:P108 :Q80-4fe7940f .
```

[7]https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format
[8]The full Turtle serialization can be obtained at: https://www.wikidata.org/wiki/Special:EntityData/Q80.ttl

```
:Q80-4fe7940f rdf:type wikibase:Statement ;
 wikibase:rank wikibase:NormalRank ;
 ps:P108 wd:Q42944 ;
 pq:P580 "1984-01-01T00:00:00Z"^^xsd:dateTime ;
 pq:P582 "1994-01-01T00:00:00Z"^^xsd:dateTime .
```

The RDF serialization uses a direct arc to represent the preferred statement represented by prefix alias `wdt:` leaving the rest of the values of a property accessible through the namespaces `p:`, `ps:` and `pq:`. The reification model employed by Wikidata creates auxiliary nodes that represent each statement. In the previous example, the node `:Q80-4fe7940f` represents the statement which can be qualified with the start and end time.

The RDF serialization model is employed in Wikidata to follow the linked data principles that enable to have a logical URI of a concept separated from its representation in different formats like HTML, JSON or RDF. It is also employed by the Wikidata Query Service which allow users to retrieve data using the SPARQL endpoint [6, 7] and by RDF-based Wikidata dumps.

Wikidata adopted entity schemas using ShEx in a new namespace (schema entities start by letter `E` followed by a number). As an example, listing 1 presents an entity schema for researcher entities[9]. Lines 1–6 contain prefix declarations following Turtle tradition. Lines 8–24 declare a `<Researcher>` shape which in this case has 7 triple constraints. The first constraint (line 9) states that items that conform to `<Researcher>` must be instances of Humans. The next line declares that the values of birth place (`wdt:P19`) must conform to shape `<Place>` declared in line 25. Line 11 declares that the values of property `wdt:P569` must belong to datatype `xsd:dateTime`. The question mark indicates that they are optional. Line 12 declares that the values of property `wdt:P108` (employer) must conform to shape `<Organization>` which is defined in line 28 (it is empty in this case). The star at the end indicates that there can be zero or more statements about `wdt:P108`. Lines 13–17 declare the constraints on the qualifiers, in this case, that it is optional to have a `pq:P580` (start) time and a `pq:P582` end time statement. Notice that these declarations about qualifiers resemble the RDF serialization model which requires one to repeat the value of the property `p:P108` and `ps:P108` for the statement. Lines 18–23 follow a similar pattern.

## 3. WShEx

As we have seen in the previous section, entity schemas in ShEx require users to be aware of how qualifiers and references are serialized in RDF which can lead to some duplication in their definition making their definitions more verbose than necessary. Another problem of ShEx schemas is that they cannot be used to directly describe the contents of Wikidata dumps in JSON which follow the Wikibase data model. In order to solve those issues, we propose an variant of ShEx called WShEx that can be used describe and validate the Wikibase data model. In this way, WShEx can also be used to validate Wikibase dumps in JSON without requiring them to be serialized in RDF. Figure 2 represents the relationship between ShEx and WShEx.

[9]The entity schema has also been created in Wikidata as:E371

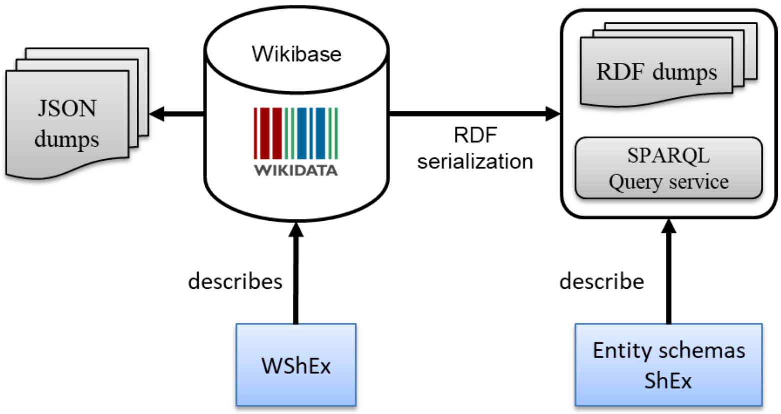


**Figure 2:** Relationship between ShEx, WShEx and Wikibase data model

In the following sections we present a simplified formal definition of WShEx adapted to Wikibase graphs as defined in 1 by presenting an abstract syntax followed by its semantic specification[10].

## 3.1. WShEx Abstract Syntax

A *WShEx Schema* is defined as a tuple $\langle L, \delta \rangle$ where $L$ set of shape labels, and $\delta : L \to S$ is a total function from labels to w-shape expressions where the set of shape expressions $se \in S$ is defined using the following abstract syntax:

| | | | |
|---|---|---|---|
| $se$ | ::= | cond | Basic boolean condition on nodes (node constraint) |
| | \| | $s$ | Shape |
| | \| | $se_1$ AND $se_2$ | Conjunction |
| | \| | $@l$ | Shape label reference for $l \in L$ |
| $s$ | ::= | CLOSED $s'$ | Closed shape |
| | \| | $s'$ | Open shape |
| $s'$ | ::= | $\{$ te $\}$ | Shape definition |
| $te$ | ::= | $te_1 ; te_2$ | Each of $te_1$ and $te_2$ |
| | \| | $te_1 \mid te_2$ | Some of $te_1$ or $te_2$ |
| | \| | $te*$ | Zero or more $te$ |
| | \| | $\_ \xrightarrow{p} @l\ qs$ | Triple constraint with predicate $p$ value conforming to $l$ and qualifier specifier $qs$ |
| | \| | $\epsilon$ | Empty triple expression |
| $qs$ | ::= | $\lfloor ps \rfloor$ | Open property specifier |
| | \| | $\lceil ps \rceil$ | Closed property specifier |
| $ps$ | ::= | ps , ps | *EachOf* property specifiers |
| | \| | $ps \mid ps$ | *OneOf* property specifiers |
| | \| | ps* | zero of more property specifiers |
| | \| | $\epsilon$ | Empty property specifier |
| | \| | p:$@l$ | Property $p$ with value conforming to shape $l$ |

[10]The specification of WShEx is published at https://www.weso.es/WShEx/

A ShEx schema that describes the Wikibase graph presented in example 1 can be defined as:

$$
\begin{array}{rcl}
L & = & \{ \quad Person, Place, Country, Organization, Date, Award \} \\
\delta(Person) & = & \{ \quad \llcorner \xrightarrow{birthDate} @Date; \llcorner \xrightarrow{birthPlace} @Place; \\
& & \quad \llcorner \xrightarrow{employer} @Organization \lfloor start : @Date, end : @Date \rfloor^* \\
& & \quad \llcorner \xrightarrow{awarded} @Award \lfloor pointTime : @Date, togetherWith : @Person \rfloor^* \\
& & \} \\
\delta(Place) & = & \{ \quad \llcorner \xrightarrow{country} @Country \} \\
\delta(Country) & = & \{ \quad \} \\
\delta(Award) & = & \{ \quad \llcorner \xrightarrow{country} @Country \} \\
\delta(Organization) & = & \{ \quad \} \\
\delta(\text{Date}) & = & \quad \in xsd : date
\end{array}
$$

It is possible to define a compact syntax for WShEx in a similar way to ShExC adding the symbols $\{ | \ldots | \}$ to declare open qualifier specifiers [11] and $[ | \ldots | ]$ for closed ones.

Listing 2 presents a WShEx example that corresponds with the entity schema presented in listing 1. It uses the default prefix declaration for items and properties. Lines 3–15 declare a `<Researcher>` shape with 5 triple constraints. The first one, declares that it must be instance of Human. The second one declares that they must have a *birth place* (`:P19`) that conforms with shape `<Place>` declared in line 16. The third constraint indicates that the values of *birth date* (`:P569`) must be `Time` values with a question mark indicating that it is optional. In WShEx it is possible to use Wikibase built-in datatypes like `Time`. Other values could be: `String`, `Item`, `Property`, `Quantity`, `MonolingualText`, `URL`, etc.[12]. Lines 8–10 contain *start* (`:P580`) and emphend (`:P582`) time qualifiers for the *employer* (`:P108`) statement. Similarly, lines 12–14 declare *point in time* (`:P585`) and *together with* (`:P1706`) qualifiers for the *award* statement.

[11] We adopted the same symbols as RDF-⋆ (https://w3c.github.io/rdf-star/cg-spec)

[12] See https://www.wikidata.org/wiki/Help:Data_type for a list of Wikibase built-in datatypes

```
PREFIX pq: <.../prop/qualifier/>
PREFIX ps: <.../prop/statement/>
PREFIX p: <.../prop/>
PREFIX wdt: <.../prop/direct/>
PREFIX wd: <.../entity/>
PREFIX xsd: <...XMLSchema#>

<Researcher> {
 wdt:P31 [ wd:Q5 ] ;
 wdt:P19 @<Place> ;
 wdt:P569 xsd:dateTime ? ;
 wdt:P108 @<Organization> * ;
 p:P108 {
  ps:P108 @<Organization> ;
  pq:P580 xsd:dateTime ? ;
  pq:P582 xsd:dateTime ?
 } * ;
 wdt:P166 @<Award> * ;
 p:P166 {
  ps:P166 @<Award> ;
  pq:P585 xsd:dateTime ? ;
  pq:P1706 @<Researcher> *
 } *
}
<Place> {
 wdt:P17 @<Country>
}
<Organization> {}
<Award> {
  wdt:P17 @<Country> ?
}
<Country> {}
```

Listing 1: Example of ShEx-based Entity schema

```
PREFIX : <.../entity/>

<Researcher> {
 :P31 [ :Q5 ] ;
 :P19 @<Place> ;
 :P569 Time ? ;
 :P108 @<Organization>
  {| :P580 Time ? ,
    :P582 Time ?
  |} * ;
 :P166 @<Award>
  {| :P585 Time ? ,
    :P1706 @<Researcher> ?
  |} *
}
<Place> {
 :P17 @<Country>
}
<Organization> {}
<Award> {
 :P17 @<Country>
}
<Country> {}
```

Listing 2: Example of WShEx schema

### 3.2. Semantics

In order to define the semantics of WShEx we employ a conformance relation parameterized by a shape assignment $G, n, \tau \vDash se$ with the meaning that node $n$ in graph $G$ conforms to shape expression $se$ with shape assignment $\tau$ according to the rules 1.

We also define a conformance relation $G, ts, \tau \Vdash te$ which declares that the triples $ts$ in graph $G$ conform to the triple expression $te$ with the shape assignment $\tau$ using the rules 2 which takes

$$\textit{Cond}\frac{cond(n) = true}{G, n, \tau \vDash cond} \qquad \textit{AND}\frac{G, n, \tau \vDash se_1 \quad G, n, \tau \vDash se_2}{G, n, \tau \vDash se_1 \text{ AND } se_2}$$

$$\textit{ClosedShape}\frac{neighs(n, G) = ts \quad G, ts, \tau \Vdash s'}{G, n, \tau \vDash_{\texttt{CLOSED}} s'}$$

$$\textit{OpenShape}\frac{ts = \{\langle x, p, y\rangle \in neighs(n, G) \mid p \in preds(te)\} \quad G, ts, \tau \Vdash s'}{G, n, \tau \vDash s'}$$

**Table 1**
Inference rules for WShEx shape expressions

into account qualifier specifiers.

$$\textit{EachOf}\frac{(ts_1, ts_2) \in part(ts) \quad G, ts_1, \tau \Vdash te_1 \quad G, ts_2, \tau \Vdash te_2}{G, ts, \tau \Vdash te_1; te_2}$$

$$\textit{OneOf}_1\frac{G, ts, \tau \Vdash te_1}{G, ts, \tau \Vdash te_1 \mid te_2} \qquad \textit{OneOf}_2\frac{G, ts, \tau \Vdash te_2}{G, ts, \tau \Vdash te_1 \mid te_2}$$

$$\textit{Star}_1\frac{}{G, \emptyset, \tau \Vdash te*}$$

$$\textit{Star}_2\frac{(ts_1, ts_2) \in part(ts) \quad G, ts_1, \tau \Vdash te \quad G, ts_2, \tau \Vdash te*}{G, ts, \tau \Vdash te*}$$

$$\textit{TripleConstraint}\frac{ts = \{\langle x, p, y, s\rangle\} \quad G, y, \tau \vDash @l \quad G, s, \tau \vdash qs}{G, ts, \tau \Vdash \sqcup \xrightarrow{p} @l\ qs}$$

**Table 2**
Inference rules for WShEx triple expressions

Finally, the conformance relationship $G, s, \tau \vdash qs$ between a graph $G$ a set $s \in P \times V$ of property-value elements, a shape assignment $\tau$ and a qualifier specifier $qs$ is defined with the rules presented in table 3.

## 3.3. Implementation and use cases

WShEx is currently implemented as a module[13] inside the ShEx-s project[14], a Scala implementation of ShEx.

The implementation includes a matcher for Entity documents which can be used to validate Wikidata dumps in JSON format. In fact, the initial motivation for WShEx was the possibility to validate JSON dumps instead of RDF dumps to create Wikidata subsets [3].

One practical aspect that appeared is the need of a converter between Entity schemas defined in ShEx and WShEx. We have already implemented a first version of this converter. This

[13]https://github.com/weso/shex-s/tree/master/modules/wshex
[14]https://www.weso.es/shex-s/

$$OpenQs\frac{s' = \{(p,v) \in s | p \in preds(ps)\} \quad G,s',\tau \vdash ps}{G,s,\tau \vdash \lfloor ps \rfloor} \qquad CloseQs\frac{G,s,\tau \vdash ps}{G,s,\tau \vdash \lceil ps \rceil}$$

$$EachOfQs\frac{G,s,\tau \vdash ps_1 \quad G,s,\tau \vdash ps_2}{G,s,\tau \vdash ps_1\ ,\ ps_2}$$

$$OneOfQs_1\frac{G,s,\tau \vdash ps_1}{G,s,\tau \vdash ps_1 \mid ps_2} \qquad OneOfQs_2\frac{G,s,\tau \vdash ps_2}{G,s,\tau \vdash ps_1 \mid ps_2}$$

$$StarQs_1\frac{}{G,\emptyset,\tau \vdash ps*} \qquad StarQs_2\frac{(s_1,s_2) \in part(s) \quad G,s_1,\tau \vdash ps \quad G,s_2,\tau \vdash ps*}{G,s,\tau \vdash ps*}$$

$$EmptyQs\frac{}{G,\emptyset,\tau \vdash \epsilon} \qquad PropertyQs\frac{s = \{(p,v)\} \quad G,v,\tau \vDash @l}{G,s,\tau \vdash p\ :\ @l}$$

**Table 3**
Inference rules for WShEx qualifier expressions

approach allows to leverage the existing entity schemas which are defined in ShEx, convert them to WShEx and use them to process Wikibase JSON dumps.

## 4. Related work

Our definition of Wikibase graphs was inspired by the formal definitions used for knowledge graphs in books like [8], which define two main data models: directed labeled graphs, which resemble RDF-based graphs and property graphs. We were also inspired by MARS (Multi-Attributed Relational Structures) [4], which present a a generalized notion of property graphs adapted to Wikidata. In that paper, they also define MAPL (Multi-Attributed Predicate Logic) as a logical formalism that can be used for ontological reasoning.

Since the appearance of ShEx in 2014, there has been a lot of interest about RDF validation and description. In 2017, the data shapes working group proposed SHACL (Shapes Constraint Language) as a W3C recommendation [9]. Although SHACL can be used to describe RDF, its main purpose is to validate and check constraints about RDF data. ShEx was adopted by Wikidata in 2019 to define entity schemas [10]. We consider that ShEx adapts better to describe data models than SHACL, which is more focused on constraint violations. A comparison between both is provided in [11] while in [12], a simple language is defined that can be used as a common subset of both.

Improving quality of Knowledge graphs in general, and Wikidata in particular, has been the focus of some research like [13, 14, 15].

## 5. Conclusions and future work

WShEx can be used to describe and validate Wikibase graphs using an extension of Shape Expressions that handle also qualifiers. We consider that WShEx schemas are more succint and adapt better to the Wikibase data model. The language has been partially implemented and is being used to create Wikidata subsets from JSON dumps. Future work is still necessary to finish the implementation including a full grammar for the compact syntax, more complete support for other Wikibase constructs like labels, descriptions, aliases, other built-in datatypes, ranks, etc. and implement a full validator for Wikibase graphs based on WShEx.


## Acknowledgments

This work has been partially funded by the Project ANGLIRU: *ANGLIRU: Applying kNowledge Graphs to research data interoperabiLIty and ReUsability*, code: PID2020-117912RB and by the Alfred P. Sloan Foundation under grant number G-2021-17106 for the development of Scholia.


## References


[1] S. Malyshev, M. Krötzsch, L. González, J. Gonsior, A. Bielefeldt, Getting the most out of Wikidata: Semantic technology usage in Wikipedia's knowledge graph, in: The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, Springer, 2018, pp. 376–394.

[2] E. Prud'hommeaux, J. E. Labra Gayo, H. Solbrig, Shape Expressions: An RDF Validation and Transformation Language, in: H. Sack, A. Filipowska, J. Lehmann, S. Hellmann (Eds.), Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014, Leipzig, Germany, September 4-5, 2014, ACM Press, 2014, pp. 32–40. doi:10.1145/2660517.2660523.

[3] J. E. Labra Gayo, Creating knowledge graphs subsets using shape expressions, 2021. URL: https://arxiv.org/abs/2110.11709. doi:10.48550/ARXIV.2110.11709.

[4] M. Marx, M. Krötzsch, V. Thost, Logic on MARS: ontologies for generalised property graphs, in: C. Sierra (Ed.), Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17), International Joint Conferences on Artificial Intelligence, 2017, pp. 1188–1194. doi:10.24963/ijcai.2017/165.

[5] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, D. Vrandečić, Introducing Wikidata to the Linked Data Web, in: P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. A. Knoblock, D. Vrandečić, P. T. Groth, N. F. Noy, K. Janowicz, C. A. Goble (Eds.), Proceedings of the 13th International Semantic Web Conference (ISWC 2014), volume 8796 of *LNCS*, Springer, 2014, pp. 50–65. doi:10.1007/978-3-319-11964-9_4.

[6] A. Bielefeldt, J. Gonsior, M. Krötzsch, Practical linked data access via SPARQL: the case of wikidata, in: T. Berners-Lee, S. Capadisli, S. Dietze, A. Hogan, K. Janowicz, J. Lehmann (Eds.), Proceedings of the WWW2018 Workshop on Linked Data on the Web (LDOW-18), volume 2073 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018.

[7] S. Malyshev, M. Krötzsch, L. González, J. Gonsior, A. Bielefeldt, Getting the Most out of Wikidata: Semantic Technology Usage in Wikipedia's Knowledge Graph, in: D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee, E. Simperl (Eds.), Proceedings of the 17th International Semantic Web Conference (ISWC'18), volume 11137 of *LNCS*, Springer, 2018, pp. 376–394.

[8] A. Hogan, E. Blomqvist, M. Cochez, C. Dámato, G. D. Melo, C. Gutierrez, S. Kirrane,

J. E. Labra Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, ACM Computing Surveys 54 (2021) 1–37. doi:10.1145/3447772.

[9] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3C Recommendation 20 July 2017, W3C Recommendation, World Wide Web Consortium, 2017. URL: https://www.w3.org/TR/2017/REC-shacl-20170720/.

[10] K. Thornton, H. Solbrig, G. S. Stupp, J. E. Labra Gayo, D. Mietchen, E. Prud'hommeaux, A. Waagmeester, Using Shape Expressions (ShEx) to Share RDF Data Models and to Guide Curation with Rigorous Validation, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 606–620. doi:10.1007/978-3-030-21348-0_39.

[11] J. E. Labra Gayo, E. Prud'hommeaux, I. Boneva, D. Kontokostas, Validating RDF Data, volume 7 of *Synthesis Lectures on the Semantic Web: Theory and Technology*, Morgan & Claypool, 2017. URL: https://doi.org/10.2200/s00786ed1v01y201707wbe016. doi:10.2200/s00786ed1v01y201707wbe016.

[12] J. E. Labra Gayo, H. García-González, D. Fernández-Alvarez, E. Prud'hommeaux, Challenges in RDF Validation, in: G. Alor-Hernández, J. L. Sánchez-Cervantes, A. Rodríguez-González, R. Valencia-García (Eds.), Current Trends in Semantic Web Technologies: Theory and Practice, Studies in Computational Intelligence, Springer, 2019, pp. 121–151. doi:10.1007/978-3-030-06149-4_6.

[13] A. Piscopo, E. Simperl, What we talk about when we talk about wikidata quality, in: Proceedings of the 15th International Symposium on Open Collaboration, ACM, 2019, pp. 1–11. doi:10.1145/3306446.3340822.

[14] H. Turki, D. Jemielniak, M. A. Hadj Taieb, J. E. Labra Gayo, M. Ben Aouicha, M. Banat, T. Shafee, E. Prud'hommeaux, T. Lubiana, D. Das, D. Mietchen, Using logical constraints to validate statistical information about disease outbreaks in collaborative knowledge graphs: the case of covid-19 epidemiology in wikidata, PeerJ Computer Science 8 (2022) e1085. URL: https://doi.org/10.7717/peerj-cs.1085. doi:10.7717/peerj-cs.1085.

[15] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe, P. Szekely, A Study of the Quality of Wikidata, Journal of Web Semantics 72 (2022) 100679. URL: https://doi.org/10.1016/j.websem.2021.100679. doi:https://doi.org/10.1016/j.websem.2021.100679.