



RENDER  
 FP7-ICT-2009-5  
 Contract no.: 257790  
 www.render-project.eu

## RENDER

### Deliverable D3.1.2

#### Evaluation of models for diversity-rich information

Editor:	Andreas Thalhammer, UIBK; Ioan Toma, UIBK; Fabian Flöck, KIT
Author(s):	Andreas Thalhammer, UIBK; Ioan Toma, UIBK; Fabian Flöck, KIT
Deliverable Nature:	Report (R)
Dissemination Level: (Confidentiality) <sup>1</sup>	Public (PU)
Contractual Delivery Date:	M18
Actual Delivery Date:	M18
Suggested Readers:	Example: Developers creating software components to be integrated into knowledge workspace, developers creating case study prototypes, etc.
Version:	V2.0
Keywords:	Knowledge Diversity Ontology, diversity models, evaluation, presentation

<sup>1</sup> Please indicate the dissemination level using one of the following codes:

• **PU** = Public • **PP** = Restricted to other programme participants (including the Commission Services) • **RE** = Restricted to a group specified by the consortium (including the Commission Services) • **CO** = Confidential, only for members of the consortium (including the Commission Services) • **Restreint UE** = Classified with the classification level "Restreint UE" according to Commission Decision 2001/844 and amendments • **Confidentiel UE** = Classified with the mention of the classification level "Confidentiel UE" according to Commission Decision 2001/844 and amendments • **Secret UE** = Classified with the mention of the classification level "Secret UE" according to Commission Decision 2001/844 and amendments

---

**Disclaimer**


---

This document contains material, which is the copyright of certain RENDER consortium parties, and may not be reproduced or copied without permission.

*In case of Public (PU):*

All RENDER consortium parties have agreed to full publication of this document.

*In case of Restricted to Programme (PP):*

All RENDER consortium parties have agreed to make this document available on request to other framework programme participants.

*In case of Restricted to Group (RE):*

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement. However, all RENDER consortium parties have agreed to make this document available to <group> / <purpose>.

*In case of Consortium confidential (CO):*

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the RENDER consortium as a whole, nor a certain party of the RENDER consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP 3 - Diversity representation and processing
Document Title:	D3.1.2 - Evaluation of models for diversity-rich information
Editor (Name, Affiliation)	Andreas Thalhammer, UIBK; Ioan Toma, UIBK; Fabian Flöck, KIT
Work package Leader (Name, affiliation)	Ioan Toma, UIBK
Estimation of PM spent on the deliverable:	6

**Copyright notice**

© 2010-2013 Participants in project RENDER

## Executive Summary

Rather than following the trend for more and more personalization – which Eli Pariser calls the “Race for Relevance” in his book “The Filter Bubble: What The Internet Is Hiding From You”<sup>2</sup> - the RENDER project focuses on enabling and retrieval of knowledge diversity. For this, we combine expertise in NLP (including named entity recognition), representation of knowledge diversity, storage, reasoning, and retrieval. Driving forces in this context are use cases provided by Telefonica, Wikipedia, and Google News. Each of these organizations provide the project with a different angle on knowledge diversity, which enables us to derive a data model that is as general as possible and at the same time as specific as necessary.

After presenting an initial version of the Knowledge Diversity Ontology (KDO v1) in D3.1.1, we continued to work on this important component that glues together the data layer with the presentation and application layers of the RENDER architecture.

KDO serves as a central data model for the development of the opinion and fact mining toolkit and the integration of extracted information into the storage layer.

In this deliverable we describe the presentation and the evaluation of the Knowledge Diversity Ontology version 2. For presentation, we use a novel approach for ontology documentation – Parrot, a RESTful service that unifies the specification and maintenance of an ontology into a single effort. For quality assurance, we evaluated the second version of KDO, pointing out current strengths and weaknesses. We also include a revised description of all classes and properties, as well as updated examples for modelling the motivating scenarios and for answering the competency questions with SPARQL.

---

<sup>2</sup> Viking, June 2011)

## Table of Contents

Executive Summary .....	3
Table of Contents .....	4
List of Figures.....	5
Abbreviations.....	7
Definitions .....	8
1 Introduction .....	9
2 Presentation of KDO.....	10
3 Evaluation of the Knowledge Diversity Ontology (KDO) .....	12
3.1 Ontology Evaluation Methodology .....	12
3.1.1 Criteria .....	12
3.2 KDO Evaluation Results .....	13
4 Conclusion .....	16
References.....	17
Annex A The Knowledge Diversity Ontology v2 .....	18

## List of Figures

Figure 1 Screenshot of the Parrot visualization of KDO. ....	10
Figure 2 Visualization of the core components of KDO version 2 .....	18

## List of Tables

Table 1: SPAQL queries answering the competency questions .....	27
---	----

## Abbreviations

<b>DCMI</b>	Dublin Core Metadata Initiative
<b>FOAF</b>	Friend Of A Friend
<b>HEO</b>	Human Emotions Ontology
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTML</b>	Hypertext Markup Language
<b>OWL</b>	Web Ontology Language
<b>KDO</b>	Knowledge Diversity Ontology
<b>NER</b>	Named Entity Recognition
<b>PRO</b>	Publishing Role Ontology
<b>REST</b>	Representational State Transfer
<b>RDF</b>	Resource Description Framework
<b>SIOC</b>	Semantically-Interlinked Online Communities
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>XML</b>	Extensible Markup Language

## Definitions

A glossary which introduces the terminology that is used in this deliverable can be found in Hasan et al. "Towards a Knowledge Diversity Model" [1].



# 1 Introduction

The technical architecture of RENDER is comprised by the following layers: data layer, application layer, and presentation layer. The data layer introduces raw unstructured or semi-structured datasets, crawlers, and repositories in order to collect and store a vast amount of processed and unprocessed information. In alignment to the use cases the raw data includes customer feedback, news or Wikipedia data. The application layer introduces the Fact Mining Toolkit (cf. D2.2.1), the Opinion Mining Toolkit (cf. D2.1.1), Diversity Services, and Ranking Services. After processing, the analyzed data is stored in a triple store that is part of the data layer. This triple store performs reasoning in order to infer new knowledge that is not explicitly contained in the original information source. This store is then accessed by the presentation layer to present the mined information in new, diversity-aware ways.

The Knowledge Diversity Ontology (KDO, cf. D3.1.1) enables to present **mined** information and to enable diversity-aware **storage, reasoning, and retrieval** of the processed information.

Currently, KDO is used in the following way:

The opinion and fact mining toolkit (cf. D2.1.1/D2.2.1) enables to process unstructured and semi-structured information in order to derive information such as topics, opinions, sentiment, and social dimensions of diversity. The Enrycher service<sup>3</sup> extracts meaningful information from text and creates links to the Reference Knowledge Stack (e.g. Open Directory Project (DMOZ)<sup>4</sup>, DBpedia<sup>5</sup>, Proton<sup>6</sup>, etc.). This information is then exported using KDO for all diversity aspects.

After the mined information is represented in RDF, it also needs to be stored in a triple store. There are two reasons for doing so: first, the data needs to be retrievable by a variety of presentation and analysis tools, second, a reasoning engine that works with the triple store can infer additional knowledge from the mined data. To enable reasoning, background knowledge has to be mined and merged. This is done by the FactForge<sup>7</sup> linked data exploration service. The resulting background knowledge, also called "Reference Knowledge Stack" serves as a key technology in order to derive new knowledge for the previously mined use case data (cf. D1.2.1).

KDO states the core data model of the RENDER technical architecture. As the partners want to stress the reuse of components developed in RENDER, the focus of this deliverable is on documentation and quality insurance for KDO. Therefore, this document introduces an HTML documentation that provides the presentation of the second version of KDO (Section 2). This documentation is included in the RENDER website<sup>8</sup> through an inline frame (iframe). Furthermore, this deliverable includes a thorough and complete evaluation of the second version of the Knowledge Diversity Ontology (Section 3). This evaluation follows a well known evaluation strategy that was conducted by staff independent from the editors of KDO. In the Appendix of this document we provide a description of the second version of KDO.

---

<sup>3</sup> <http://enrycher.ijs.si/>

<sup>4</sup> <http://www.dmoz.org/>

<sup>5</sup> <http://dbpedia.org/>

<sup>6</sup> <http://proton.semanticweb.org/>

<sup>7</sup> <http://factforge.net/>

<sup>8</sup> <http://render-project.eu>

## 2 Presentation of KDO

Providing an HTML representation of an ontology is common practice in the Semantic Web community. This ontology documentation should include an explanation of all classes, properties and individuals as well as meta information such as authors, version information and a change log.

Creating such a documentation manually is connected with a serious effort. Therefore, a variety of tools emerged that help to create such specifications (semi-) automatically.

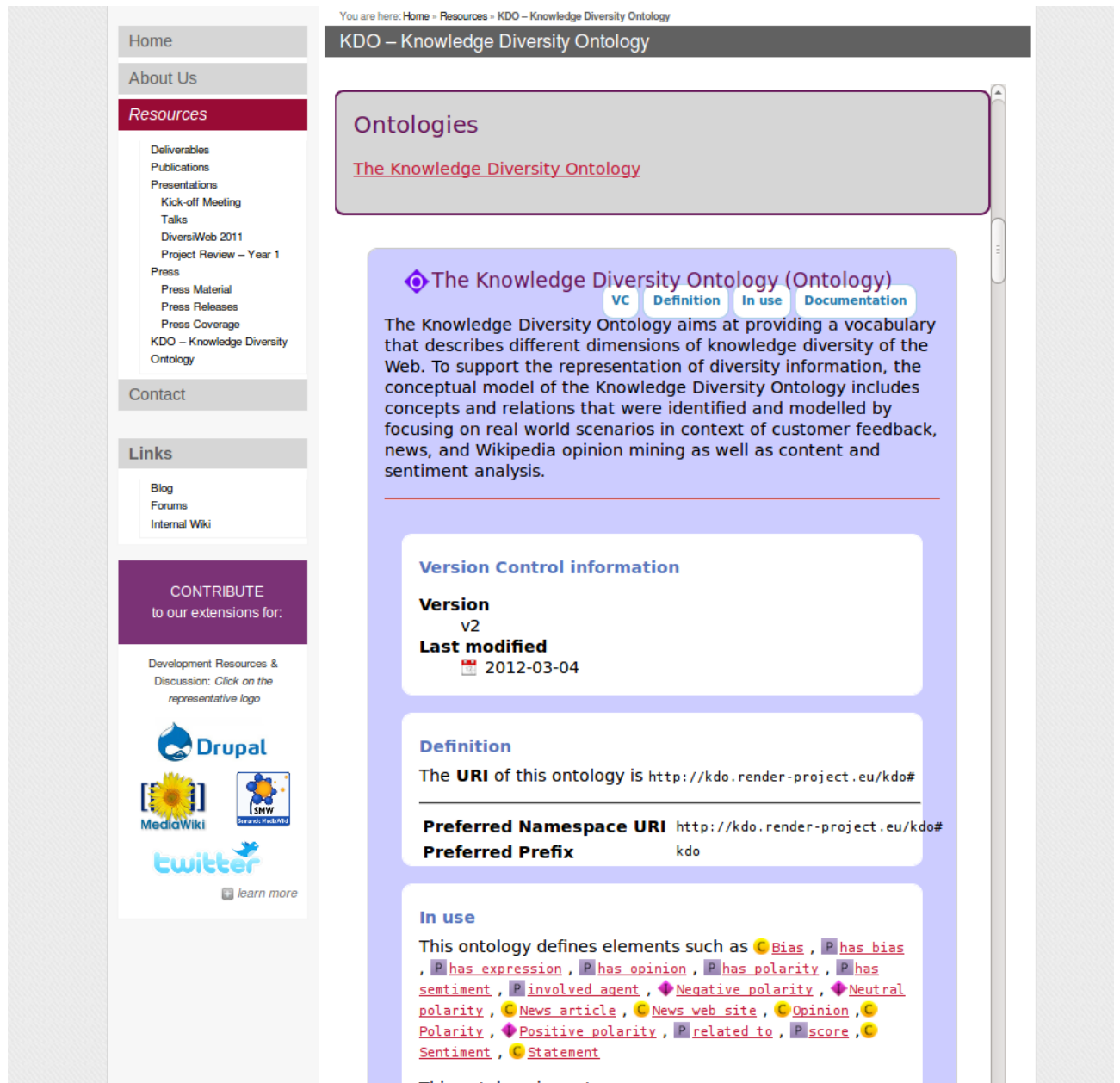


Figure 1 Screenshot of the Parrot visualization of KDO.

In the following we will list the most common tools for generating an ontology documentation:

- Parrot<sup>9</sup>
- VocDoc<sup>10</sup>
- SpecGen<sup>11</sup>
- OWLDoc<sup>12</sup>, Ontology Browser<sup>13</sup>, Ontology Browser fork<sup>14</sup>
- Neologism<sup>15</sup>
- TopQuadrant<sup>16</sup>

Out of these tools, Parrot that was chosen to support the documentation of the KDO ontology. This documentation software was developed in context to the European project ONTORULE.<sup>17</sup> Parrot works as a RESTful service and creates an HTML ontology documentation out of four different kinds of input: URI of an online ontology, direct input, file upload and existing report. It is also possible to provide individual style sheets in order to adapt the look and feel. The main advantage of this Web Service approach to ontology documentation is the maintainability. As soon as the referenced OWL document changes, the documentation changes as well. This avoids inconsistencies that may occur in cases where the specification is not directly connected to the implementation (like it is the case with the most other tools).

For the documentation of the KDO ontology, we chose the “existing report” option and provide four files:

- <http://kdo.render-project.eu/kdo> - the actual OWL ontology
- <http://kdo.render-project.eu/info.html> - the specification about where to find the prologue and the appendix
- <http://kdo.render-project.eu/prologueappendix.html> - the file containing the prologue and the appendix
- <http://kdo.render-project.eu/renderstyle.css> - the style sheet to adapt the look and feel for the integration into the RENDER web site

Combined with the RESTful service of Parrot, the documentation is automatically created and available at:

<http://kdo.render-project.eu/>

Figure 1 shows a screenshot of the resulting documentation.

---

<sup>9</sup> <http://ontorule-project.eu/parrot/parrot>

<sup>10</sup> <http://kantenwerk.org/vocdoc/>

<sup>11</sup> <https://bitbucket.org/wikier/specgen/wiki/Home>

<sup>12</sup> <http://code.google.com/p/co-ode-owl-plugins/wiki/OWLDoc>

<sup>13</sup> <http://code.google.com/p/ontology-browser/>

<sup>14</sup> <https://bitbucket.org/kurzum/ontology-browser-fork>

<sup>15</sup> <http://neologism.deri.ie/>

<sup>16</sup> <http://topquadrant.com/>

<sup>17</sup> <http://ontorule-project.eu/>

## 3 Evaluation of the Knowledge Diversity Ontology (KDO)

In the following we explain what criteria were tested, which methods were employed in the testing and the results of those tests.

### 3.1 Ontology Evaluation Methodology

The evaluation of the KDO is based on the criteria and methods proposed by Vrandečić<sup>18</sup>, who establishes a framework of criteria distilled from an extensive literature review on ontology evaluation models. These criteria and the evaluation methods applied for the KDO are listed in the following.

The evaluation was carried out by one research group member of the partner KIT, who was not involved in the development of the KDO. It was reviewed internally by other KIT members not related to RENDER. As the criteria used and the methods proposed for testing them were strictly adhered to according to Vrandečić, high objectivity of the evaluation was maintained.

#### 3.1.1 Criteria

1. **Accuracy** “is a criteria [sic] that states if the axioms of the ontology comply to the knowledge of the stakeholders about the domain. A higher accuracy comes from correct definitions and descriptions of classes, properties and individuals.”

To test for accuracy, we a) manually checked the terminology of the KDO for ambivalence and compliance to the stakeholders domain knowledge and b) tested the competency questions defined as SPARQL in Annex A against the KDO via SPARQL queries (using the OWL consistency checker tool<sup>19</sup>) directly on the KDO RDF file.

2. **Adaptability** “measures how far the ontology anticipates its uses. An ontology should offer the conceptual foundation for a range of anticipated tasks [...]. It should be possible to extend and specialize the ontology monotonically,” i.e. without removing axioms.

Adaptability was assessed by hand and by checking for redundancy in the ontology, as redundancy of concepts heightens the effort to make adaptations to those concepts.

3. **Clarity** “measures how effectively the ontology communicates the intended meaning of the defined terms. Definitions should be objective and independent of the context. Names of elements should be understandable and unambiguous.” Clarity in short is about the readability and usability for humans.

Clarity was evaluated by assessing if the resources in the ontology are properly labelled (understandable) and if URIs are constructed in a consistent and comprehensible manner. The quality of the documentation was also considered.

4. **Completeness** “measures if the domain of interest is appropriately covered. All questions the ontology should be able to answer can be answered.”

Completeness was evaluated via the competency questions via SPARQL queries.

5. **Computational efficiency** “measures the ability of the used tools to work with the ontology [...]”

Computational complexity was analysed as well as the runtime for two reasoners, Pellet and the Fact++ plugin for Protege.

6. **Conciseness** “is the criteria [sic] that states if the ontology includes irrelevant elements with the regard to the domain to be covered [...] or redundant representations of the semantics.”

<sup>18</sup> Denny Vrandečić: „Ontology Evaluation“ – Dissertation at KIT Karlsruhe, 2010.

<sup>19</sup> Online tool available at <http://www.mindswap.org/2003/pellet/demo.shtml>, the RDF was loaded into the tool instead of querying the KDO endpoint.

We tested if the KDO contains redundant or otherwise superfluous concepts and blank nodes unnecessary to answer the competency questions.

7. **Consistency** “describes that the ontology does not include or allow for any contradictions.”

Consistency was checked using the Fact++ reasoner plugin of the Protégé ontology tool<sup>20</sup> and the Pellet<sup>21</sup> reasoner.

8. **Organizational fitness** describes “how easily an ontology can be deployed within an organization.”

This criterion was checked by a) aligning the development process of the KDO with the respective partners in RENDER and b) by gathering feedback from the partners on the KDOv2 as presented in this document.

## 3.2 KDO Evaluation Results

The KDO data was checked first for syntactical validity with the W3C’s RDF validation service<sup>22</sup> and did not produce any errors. It is also completely dereferenceable, as checked via the VAPOUR Linked Data validator<sup>23</sup>. The RDF:ALERTS validator<sup>24</sup> also did not return any warnings, checking for syntax errors, undefined classes or properties, inconsistencies, bogus inverse-functional property values, atypical use of core vocabularies and datatype errors.

### *Inconsistency of actual RDF to description*

- No inconsistencies were found examining the documentation and the OWL file.

### *Competency questions via SPARQL queries*

Competency questions as defined in Annex A were evaluated via SPARQL queries (using the OWL consistency checker tool) directly on the KDO RDF file. We report the results before discussing the criteria as introduced in Section 3.1:

- Nr.7 and 9 cannot be answered as the concepts with the prefix pro: are not defined or referred to in the KDO.
- Nr.8 cannot be answered as the question is deprecated.
- Nr.17 cannot be answered as question is deprecated.
- The following SPARQL queries can be answered: 1,2,3,4,5,6,10,11,12,13,14,15,16

### 1. Accuracy

The correct modelling of “reality” by the KDO is difficult to assess, as especially the terms “Opinion”, “Bias” and “Sentiment” are somewhat ambiguously used in common and even in scientific language. However, by not building up these concepts on top of each other but connecting them via kdo:Statement, no assertion is made about their relation to each other, aside from the fact that they are distinct and can be expressed in a statement of any form, which is a reasonable assumption. The model leaves it open to the end user how a kdo:Opinion or kdo:Bias are filled (any URI is possible). For Sentiment, there is a decimal score defined and three Polarity instances, positive, neutral and negative, which can be regarded as common value range and notations, respectively, in Sentiment Analysis. The introduction of kdo:NewsArticle is a useful extension of the sioc:Post subclasses, as no similar subclass is defined in the SIOC ontology. In

<sup>20</sup> <http://protege.stanford.edu/>

<sup>21</sup> <http://clarkparsia.com/pellet>

<sup>22</sup> <http://www.w3.org/RDF/Validator/>

<sup>23</sup> <http://validator.linkeddata.org/>

<sup>24</sup> <http://swse.deri.org/RDFAlerts/>

conclusion, the selected modelling seems to be a good choice to represent the heterogeneous terminology and concepts accurately.

Some of the competency questions defined in Section 3.4 could not be answered by the KDO (as SPARQL queries), which speaks against the accuracy of the KDO or a wrong formulation of competency questions.

## 2. Adaptability

The following definitions of classes are showing redundancy in the KDO, as they are already defined as classes in the respective ontologies:

```
<owl:Class rdf:about="http://rdfs.org/sioc/ns#Container"/>
<owl:Class rdf:about="http://rdfs.org/sioc/ns#Post"/>
<owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
<owl:Class rdf:about="http://xmlns.com/foaf/0.1/Agent"/>
```

These redundancies should be remedied in order to make the KDO slimmer and easier to edit.

There are no unnecessary blank nodes created and because of the independent structure of the main KDO classes to each other, an extension or other adaption is facilitated.

## 3. Clarity

We ran some tests employing the inspector modules of the Eyeball tool, which is designed “for checking RDF and OWL models for various common problems”, especially implausible definitions of concepts.<sup>25</sup> The URI inspector was used to check if every URI in the model is well-formed. The LITERAL inspector was used to check for syntactically correct language codes, syntactically correct datatypes and conformance of the lexical form of typed literals to their datatype. Both inspectors did not return warnings. The URIs are also logically constructed to be readable by human users. With the PREFIX inspector we identified the use of dc: as a prefix for the DublinCore Terms ontology instead of dcterms, with the latter being more common. This is, however, a neglectable fault. Moreover, the KDO is assigned as as base namespace and additionally given the prefix kdo:, which is redundant and may be confusing for users of the KDO.

A pointer to the RENDER homepage with the KDO documentation is given and the documentation itself is satisfactory in terms of comprehensiveness and ease of understanding.

## 4. Completeness

As assessed via the Eyeball inspectors AllTyped, Class and Property, all properties and classes are declared. All resources have an rdf:type property in the KDO (or the used schemas).

## 5. Complexity

The KDO is expressed in OWL Full with an DL expressivity of SHI(D). This results in a complexity of ExpTime-complete in regard to concept satisfiability and A-Box consistency.<sup>26</sup>

The Pellet reasoner runs over the KDO in 4573 ms<sup>27</sup>, the Fact++ plugin for Protégé needs less than 1 s, while the latter is not loading and checking imported ontologies.

## 6. Conciseness

---

<sup>25</sup> Eyeball is based on the Jena framework. <http://incubator.apache.org/jena/documentation/tools/eyeball-manual.html>

<sup>26</sup> According to <http://www.cs.man.ac.uk/~ezolin/dl/>

<sup>27</sup> Via the online service at <http://www.mindswap.org/2003/pellet/demo.shtml> and loading 3272 species, validating 1301.

The redundant concepts listed under “2. Adaptability” pose an impairment to the conciseness of the KDO as well. Apart from these, some concepts proved to be superfluous for answering the competency questions as proposed in Section 3.4, such as `kdo:relatedTo` and `kdo:involvedAgent`.

There are no excess blank nodes defined. The KDO only defines the necessary core concepts (plus one extension to SIOC with `kdo:NewsArticle`) and reuses existing ontologies for all remaining modeling needs, which makes it very light-weight and lean.

#### 7. **Consistency**

Consistency checking with the Fact++ reasoner plugin of Protégé and with Pellet did not reveal any faults in terms of consistency, neither did a manual assessment.

#### 8. **Organizational fitness**

The KDO was developed in close synchronization with the RENDER use case and development partners and is therefore fitted to the use cases and ready to represent all resulting data from the RENDER data processing tools and architecture. Still, the lack of concepts for Tweets and Wikipedia edits (or possible other units apart from articles in Wikipedia) might prove to induce a need for extension of the KDO in the future. However, as the adaptability of the KDO is satisfactory, possible extensions, also for yet unknown use organizational users, are not difficult to implement. This is also achieved by the lean and rather generic specification of the KDO, which leaves margins for organizational adaption.

## 4 Conclusion

This deliverable provides the presentation and an evaluation of the second version of the Knowledge Diversity Ontology (KDO v2). For the reasons of completeness, in the appendix, we include a description of all classes and properties, as well as updated examples for modelling the motivating scenarios and for answering the competency questions with SPARQL.

For presentation, we explain why and how the team decided to use a novel approach for ontology documentation – Parrot, a RESTful service that makes the specification and maintenance of an ontology easy and merges these two tasks into a single effort. This also ensures consistency between the specification and the implementation.

For quality assurance, we also evaluated the second version of KDO and pointed out current weaknesses.

In conclusion, it can be stated that the development of a profound vocabulary is a continuous process that hardly is ever finished. This also goes along with the move in the digital world that bares the potential to redefine whole concepts from one day to another.

With the thorough documentation and the reached quality, the creators of KDO are confident that KDO will be reused by other developers and institutions.



## References

- [1] R. Hasan, K. Siorpaes, R. Krummenacher, and F. Flöck, "Towards a Knowledge Diversity Model," in First International Workshop on Knowledge Diversity on the Web (DiversiWeb 2011), 20th World Wide Web Conference (WWW 2011), 2011.

## Annex A The Knowledge Diversity Ontology v2

During the trials with the first version of the Knowledge Discovery Ontology (KDO v1), various issues were discovered (cf. Section 3). We addressed these issues accordingly and a reworked version of KDO emerged. The major difference to the previous version is the new flexibility in terms of meta and provenance information. While the previous schema was too stiff and did not support modelling of a variety of information, we now separate this information from the information that is relevant to diversity. Therefore, **kdo:Opinion** represented by its URI is used to contextualize meta and provenance information. All information that is relevant for representing the diversity of the content, like sentiment, polarities, and biases can still be modelled with the KDO. Figure 2 visualizes the core components of KDO v2.

For completeness, we revisit the description of all classes and properties (that was first presented in D3.1.1) respectively. Similarly, we also revisit the modelling and retrieval of KDO-represented information.

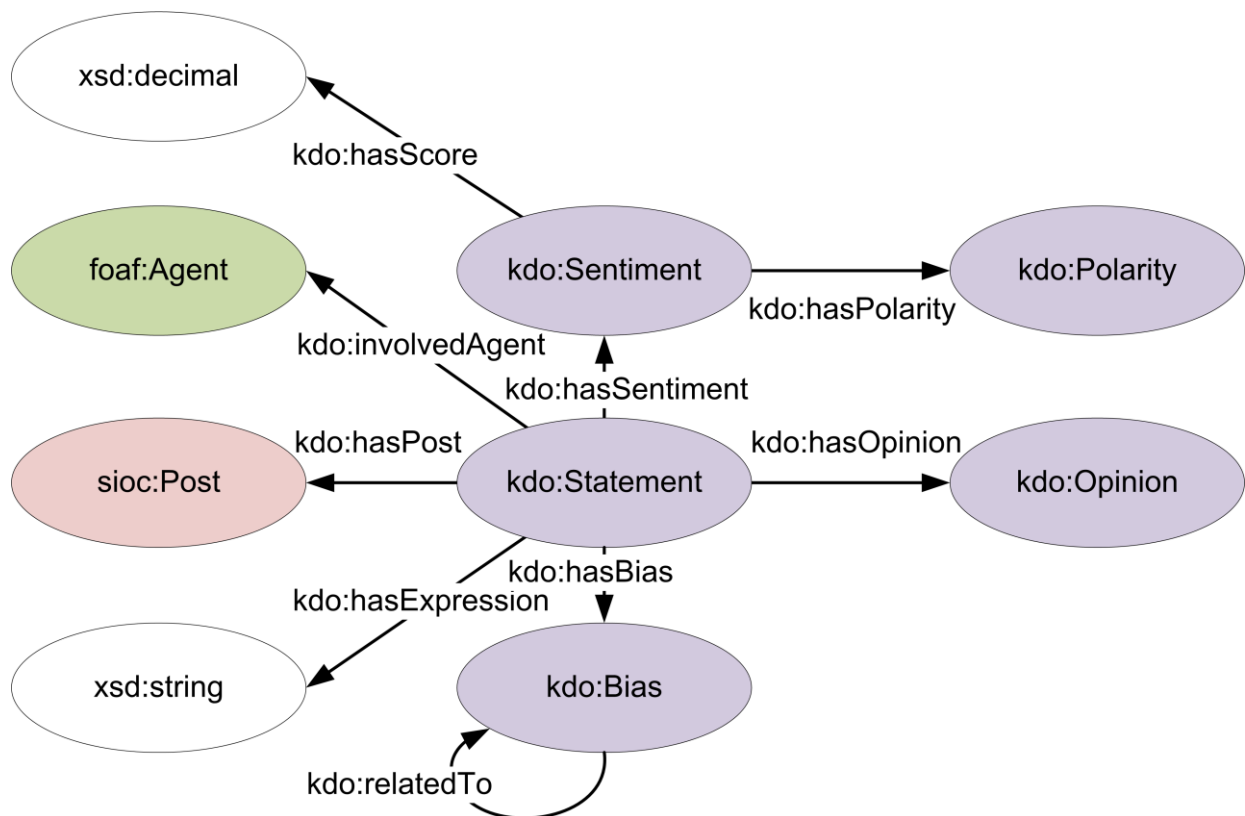


Figure 2 Visualization of the core components of KDO version 2

### KDO Classes (revisited)

We describe each of the classes below:

#### Class: Opinion

URI	http://kdo.render-project.eu/kdo#Opinion
Label	Opinion
Subclass of	-

The opinion class represents is used to contextualize<sup>28</sup> provenance, crawling other meta-information. A blank node may be used to represent opinions. Opinions are connected to statement through **kdo:hasOpinion**.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasOpinion [ a kdo:Opinion ;
    enrycher:crawlingDate "2012-02-20T01:03:00Z"^^xsd:dateTime ;
  ] .
```

### Class: Statement

URI	http://kdo.render-project.eu/kdo#Statement
Label	Statement
Subclass of	-

The statement class is the central class in the core of the KD ontology. It serves as a hub that connects sentiments (**kdo:hasSentiment**), opinions (**kdo:hasOpinion**), biases (**kdo:hasBias**), posts (**kdo:hasPost**), and agents (**kdo:involvedAgent**). Moreover, each statement may have an expression which is a string literal.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasOpinion [ a kdo:Opinion ;
    enrycher:crawlingDate "2012-02-20T01:03:00Z"^^xsd:dateTime ] ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity ;
    kdo:hasScore "0.9"^^xsd:decimal ] ;
  kdo:hasBias [ a kdo:Bias ;
    kdo:relatedTo _:0823wlej94 ] ;
  kdo:hasPost <#Post237398> ;
  kdo:involvedAgent <#Berkel> ;
  kdo:hasExpression "Movistar is the best tariff ever."^^xsd:string .
```

### Class: Sentiment

URI	http://kdo.render-project.eu/kdo#Sentiment
Label	Sentiment
Subclass of	-

The sentiment class represents the sentiments expressed by the opinions and posts. A sentiment can have (**kdo:hasPolarity**) different types of polarity (positive, negative, and neutral) and a decimal score (**kdo:hasScore**). It is in the range of **kdo:hasSentiment**.

An example is provided in the following:

<sup>28</sup> [http://en.wikipedia.org/wiki/Resource\\_Description\\_Framework#Statement\\_reification\\_and\\_context](http://en.wikipedia.org/wiki/Resource_Description_Framework#Statement_reification_and_context)

```
<#BerkelWikiOpinion> a kdo:Opinion ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity;
    kdo:hasScore "0.9"^^xsd:decimal] .
```

### Class: Bias

URI	http://kdo.render-project.eu/kdo#Bias
Label	Bias
Subclass of	-

A bias can be attached to a **kdo:Statement** and can therefore also relate to agents, posts, and opinions. More specifically, a bias can also relate to another bias.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasBias [a kdo:Bias ;
    kdo:relatedTo _:0823wlejr94 ] .
```

### Class: Polarity

URI	http://kdo.render-project.eu/kdo#Polarity
Label	Polarity
Subclass of	-

The **kdo:Polarity** class is used to express the polarity of sentiments. The predefined instances of this class are: **kdo:neutralPolarity**, **kdo:positivePolarity**, **kdo:negativePolarity**.

An example is provided in the following:

```
<#BerkelWikiStatement> a kdo:Statement ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity ] .
```

### Class: NewsArticle

URI	http://kdo.render-project.eu/kdo#NewsArticle
Label	News article
Subclass of	sioc:Post

The **kdo:NewsArticle** class represents news articles. The news article class is defined because there is no class to represent news articles in the SIOC ontology. The NewsArticle is a subclass of the **sioc:Post** class which is itself a subclass of the **foaf:Document**.

An example is provided in the following:

```
<http://www.bloomberg.com/article/c2lj3h3j4j4> a kdo:NewsArticle .
```

### Class: NewsWebsite

URI	http://kdo.render-project.eu/kdo#NewsWebsite
Label	News web site
Subclass of	sioc:Container

The **kdo:NewsWebsite** class represents news publisher containers that are commonly related to a single web site. The NewsWebsite class is defined because there is no container class to represent the news publisher web sites in the SIOC ontology. The NewsWebsite class is defined as a subclass of the **sioc:Container**.

An example is provided in the following:

```
<http://www.bloomberg.com/article/c2lj3h3j4j4> a kdo:NewsArticle ;
  sioc:hasContainer [ a kdo:NewsWebsite ;
    sioc:has_space <http://www.bloomberg.com/ ] .
```

## KDO Properties (revisited)

The properties of KDO are described below:

### Property: hasExpression

RI	http://kdo.render-project.eu/kdo#hasExpression
Label	has expression
Domain	kdo:Statement
Range	xsd:string

This property attaches a string literal to a **kdo:Statement**. This literal should be a text excerpt which states a statement and is extracted from a related post.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasExpression "Movistar is the best tariff ever."^^xsd:string .
```

### Property: hasSentiment

URI	http://kdo.render-project.eu/kdo#hasSentiment
Label	has sentiment
Domain	kdo:Statement
Range	kdo:Sentiment

The **kdo:hasSentiment** property is used to specify which **kdo:Sentiment** is attached to a **kdo:Statement**.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity;
    kdo:hasScore "0.9"^^xsd:decimal ] .
```

### Property: hasPolarity

URI	http://kdo.render-project.eu/kdo#hasPolarity
Label	has polarity
Domain	kdo:Sentiment
Range	kdo:Polarity

The **kdo:hasPolarity** is used to specify the polarity of a **kdo:Sentiment**. It has three individuals: **kdo:positivePolarity** representing the positive orientation, **kdo:negativePolarity** representing the negative orientation, **kdo:neutralPolarity** representing the neutral.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity;
    kdo:hasScore "0.9"^^xsd:decimal ] .
```

### Property: hasScore

URI	http://kdo.render-project.eu/kdo#hasScore
Label	has score
Domain	kdo:Sentiment
Range	xsd:decimal

This property indicates the score of a **kdo:Sentiment**. This score is denoted as a decimal. However, the range can vary and depends on the implementation of the sentiment extraction process.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity;
    kdo:hasScore "0.9"^^xsd:decimal ] .
```

### Property: hasOpinion

URI	http://kdo.render-project.eu/kdo#hasOpinion
-----	---

Label	has opinion
Domain	kdo:Statement
Range	kdo:Opinion

Relates a **kdo:Statement** to a **kdo:Opinion**.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasOpinion [ a kdo:Opinion ;
    enrycher:crawlingDate "2012-02-20T01:03:00Z"^^xsd:dateTime ;
  ] .
```

### Property: involvedAgent

URI	http://kdo.render-project.eu/kdo#:involvedAgent
Label	Involved agent
Domain	kdo: Statement
Range	foaf:Agent

The **kdo:involvedAgent** property is used to specify the holder of an opinion. An opinion holder is represented by the **foaf:Agent** class.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:invlvedAgent <#Berkel> ;
```

### Property: hasPost

URI	http://kdo.render-project.eu/kdo#hasPost
Label	has post
Domain	kdo:Statement
Range	sioc:Post

The property **kdo:hasPost** relates a **kdo:Statement** to a post (**sioc:Post**).

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  kdo:hasPost <#Post237398> .
```

### Property: hasBias

URI	http://kdo.render-project.eu/kdo#hasBias
Label	has bias

Domain	kdo:Statement
Range	kdo:Bias

The **kdo:hasBias** property is used associate a **kdo:Bias** with a **kdo:Statement**.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  hasBias [ a kdo:Bias ;
    kdo:relatedTo <#anotherBias> ;
  ] .
```

### Property: relatedTo

URI	http://kdo.render-project.eu/kdo#relatedTo
Label	related to
Domain	kdo:Bias
Range	kdo:Bias

Relates one bias to another.

An example is provided in the following:

```
<#Statement20234789> a kdo:Statement ;
  hasBias [ a kdo:Bias ;
    kdo:relatedTo <#anotherBias> ;
  ] .
```

## Modelling motivating scenarios (revisited)

In D3.1.1 we introduced seven motivating scenarios. These scenarios are strongly related to the use cases and serve as a foundation of the ontology development task. Three motivating scenarios were chosen as an example for modelling data with KDO. Each of those scenarios is related to one of the use cases.

At this point we will revisit the modelling of the motivating scenarios. In the following we will show how the motivating scenarios are modelled with KDO v2.

An important difference between KDO v1 and v2 is that provenance and meta information such as crawling information, publisher, language, publishing date, and authors are stored in the context of a **kdo:Opinion** node. Therefore, the opinion node serves as a container for such information. Although most triple stores support context, this concept is not foreseen in the serializations of RDF. As a consequence, we will add an opinion node to each of the following statements. The reader can assume that using this node as context will provide provenance and meta information.

First, we model **Scenario 1** (cf. section 2.1.1 of D3.1.1) that relates to the first of the “News Articles” use cases:

```
<#FerdinandMaier> a foaf:Person ;
  foaf:firstName "Ferdinand" ;
  foaf:lastName "Maier" .
```



```

<#JamesMcDonald> a foaf:Person ;
  foaf:firstName "James" ;
  foaf:lastName "McDonald" .

<#GGC> a foaf:Organization ;
  foaf:name "GGC" .

<#FMW> a foaf:Organization ;
  foaf:name "FMW" .

<#Bigat> a foaf:Organization ;
  foaf:name "Bigat" .

<http://www.bloomberg.com/article/c2lj3h3j4j4> a kdo:NewsArticle;
  sioc:content "(...) I conclude this article with the following statement: the takeover will happen and benefits both parties." ;
  pro:isRelatedToRoleInTime [ a pro:RoleInTime ;
    pro:withRole pro:author ;
    pro:isRoleHeldBy <#JamesMcDonald> ] ;
  pro:isRelatedToRoleInTime [ a pro:RoleInTime ;
    pro:withRole pro:publisher ;
    pro:isRoleHeldBy <#GGC> ] ;
  sioc:hasContainer [ a kdo:NewsWebsite ;
    sioc:has_space <http://www.bloomberg.com/> ]
  sioc:topic <http://sws.geonames.org/3165524/> ;
  sioc:topic <#FerdinandMaier> ;
  sioc:topic <#FMW> ;
  sioc:topic <#Bigat>.

<#Statement1> a kdo:Statement ;
  kdo:hasExpression "the takeover will happen and benefits both parties" ;
  kdo:hasPost <http://www.bloomberg.com/article/c2lj3h3j4j4> ;
  kdo:hasOpinion [ a kdo:Opinion ] ;
  kdo:involvedAgent <#JamesMcDonald> ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:positivePolarity] .

```

### Listing 1: Scenario 1 modelled with the KDO v2 (Turtle serialization)

For the use case that relates to the collaborative knowledge bases, we chose **Scenario 4** (cf. section 2.2.2 of D3.1.1):

```

<http://es.wikipedia.org/wiki/Crisis_del_euro_en_2010> a sioc:WikiArticle;
  a sioc:WikiArticle;
  sioc:content "La Crisis de la deuda soberana de la Eurozona, también llamada crisis del euro o crisis de la zona euro, es una serie de acontecimientos que han venido afectando negativamente desde principios del año 2010 a los 16 estados miembros de la UE que conforman la Eurozona o Zona Euro, esto es, que han adoptado el euro como moneda única y componen entre sí una unión monetaria plurinacional en el seno de la Unión Europea. (...)" ;
  sioc:topic <http://sws.geonames.org/2963597/ > ;

```

```

sioc:topic <http://sws.geonames.org/390903/ > .

<http://de.wikipedia.org/wiki/Euro-Krise> a sioc:WikiArticle;
  sioc:content "Mit Staatsschuldenkrise im Euroraum (auch Schuldenkrise oder Euro-Krise genannt) werden Verschuldungskrisen einiger Mitgliedstaaten der Eurozone bezeichnet, die den Zahlungsverpflichtungen aus ihrer Verschuldung ohne Unterstützung Dritter nicht mehr nachkommen können. (...)" ;
  sioc:topic <http://sws.geonames.org/390903/> ;
  sioc:topic <http://sws.geonames.org/2510769/> ;
  sioc:topic <http://sws.geonames.org/2264397/> .

<#SpanishWikiStatement> a kdo:Statement ;
  kdo:hasExpression "The financial crisis mainly affects Greece and Ireland" ;
  kdo:hasPost <http://es.wikipedia.org/wiki/Crisis_del_euro_en_2010> ;
  kdo:hasOpinion [a kdo:Opinion] .

<#berkel> a foaf:Agent ;
  foaf:nick "berkel" .

<#BerkelStatement> a kdo:Statement ;
  kdo:hasPost <http://de.wikipedia.org/wiki/Euro-Krise> ;
  kdo:hasSentiment [ a kdo:Sentiment ;
    kdo:hasPolarity kdo:negativePolarity ] .
  kdo:involvedAgent <#berkel> ;
  kdo:hasOpinion [a kdo:Opinion] .

```

**Listing 2: Scenario 4 modelled with the KDO v2 (Turtle serialization)**

For the use case that relates to the customer feedback, we chose **Scenario 5** (cf. section 2.3.1 of D3.1.1):

```

<#max_1977_1> a foaf:Agent ;
  foaf:nick "max_1977_1" .

<#tom294> a foaf:Agent ;
  foaf:nick "tom294" .

<#onlineUsers> a foaf:Group ;
  rdfs:label "Online Users" ;
  foaf:member <#max_1977_1> ;
  foaf:member <#tom294> .

<http://twitter.com/#!/some_user1/status/94027610476126208> a sioc:Microblog ;
  sioc:topic <http://www.dmoz.org/Regional/Europe/Spain/Communications/Wireless/Tarifa_Autónomos> .

<http://twitter.com/#!/some_user2/status/6524389758437534> a sioc:Microblog ;
  sioc:topic <http://www.dmoz.org/Regional/Europe/Spain/Communications/Wireless/Tarifa_Autónomos> .

<#TarifaStatement1> a kdo:Statement ;
  kdo:hasPost <http://twitter.com/#!/some_user1/status/94027610476126208> ;

```

```
kdo:hasExpression "Yo recomiendo la tarifa para autónomos" ;
kdo:hasSentiment [ a kdo:Sentiment;
  kdo:hasPolarity kdo:positivePolarity ] ;
kdo:involvedAgent <#tom294> ;
kdo:hasOpinion [a kdo:Opinion] .

<#TarifaStatement2> a kdo: Statement ;
kdo:hasPost <http://twitter.com/#!/some_user2/status/6524389758437534> ;
kdo:hasExpression "No contratéis la tarifa para autónomos, es un robo!" ;
kdo:hasSentiment [ a kdo:Sentiment ;
  kdo:hasPolarity kdo:negativePolarity ] ;
kdo:involvedAgent <#max_1977_1> ;
kdo:hasOpinion [a kdo:Opinion] .
```

**Listing 3: Scenario 5 modelled with the KDO**

## Answering competency questions with SPARQL (revisited)

In the following table, we exemplify for each of the competency questions defined in section 3.1 of D3.1.1 how the questioned item can be retrieved using a SPARQL query with KDO v2.

**Table 1: SPAQL queries answering the competency questions**

Question Number	SPARQL
1	<pre>SELECT ?uri WHERE {   ?uri rdf:type sioc:Post }</pre>
2	<pre>SELECT ?source WHERE {   &lt;document&gt; sioc:has_container ?container.   ?container sioc:has_space ?source }</pre>
3	<pre>SELECT ?type WHERE {   &lt;document&gt; rdf:type ?type}</pre>
4	<pre>SELECT ?topic WHERE {   &lt;document&gt; sioc:topic ?topic }</pre>
5	<pre>SELECT ?language WHERE {   &lt;document&gt; dcterms:language ?language }</pre>
6	<pre>SELECT ?date WHERE {   &lt;document&gt; dcterms:created ?date }</pre>
7	<pre>SELECT ?contributor, ?role WHERE {</pre>

	<pre> &lt;document&gt; pro:isRelatedToRoleInTime ?roleInTime. ?roleInTime pro:withRole ?role. ?roleInTime pro:isRoleHeldBy ?contributor } </pre>
<b>8</b>	<b>Deprecated</b>
<b>9</b>	<pre> SELECT ?contributor, ?group WHERE {   &lt;document&gt; pro:isRelatedToRoleInTime ?roleInTime.   ?roleInTime pro:isRoleHeldBy ?contributor.   ?group foaf:member ?contributor } </pre>
<b>10</b>	<pre> SELECT ?bias WHERE {   ?statement kdo:hasPost &lt;document&gt; .   ?statement kdo:hasBias ?bias } </pre>
<b>11</b>	<pre> SELECT ?version WHERE {   &lt;document&gt; dcterms:hasVersion ?version } </pre>
<b>12</b>	<pre> SELECT ?opinion WHERE {   ?statement kdo:hasPost &lt;document&gt; .   ?statement kdo:hasOpinion ?opinion } </pre>
<b>13</b>	<pre> SELECT ?opinionHolder WHERE {   ?statement kdo:hasOpinion &lt;opinion&gt;.   ?statement kdo:involvedAgent ?opinionHolder } </pre>
<b>14</b>	<pre> SELECT ?sentiment WHERE {   ?statement kdo:hasOpinion &lt;opinion&gt; .   ?statement kdo:hasSentiment ?sentiment } </pre>
<b>15</b>	<pre> SELECT ?sentiment WHERE {   ?statement kdo:hasPost &lt;document&gt; .   ?statement kdo:hasSentiment ?sentiment } </pre>
<b>16</b>	<pre> SELECT ?polarity WHERE {   &lt;sentiment&gt; kdo:hasPolarity ?polarity } </pre>
<b>17</b>	<b>Deprecated</b>
<b>18</b>	<pre> SELECT ?expression WHERE { </pre>

	<pre>?statement kdo:hasOpinion &lt;opinion&gt; . ?statement kdo:hasExpression ?expression }</pre>
--	---