Towards Vocabulary Development by Convention

Irlán Grangel-González, Lavdim Halilaj, Gökhan Coskun and Sören Auer
Enterprise Information Systems, University of Bonn, Bonn, Germany

Keywords: Vocabulary Development by Convention, Convention over Configuration.

Abstract: A major bottleneck for a wider deployment and use of ontologies and knowledge engineering techniques is the lack of established conventions along with cumbersome and inefficient support for vocabulary and ontology authoring. We argue, that the pragmatic development by convention paradigm well-accepted within software engineering, can be successfully applied for ontology engineering, too. However, the definition of a valid set of conventions requires broadly-accepted best-practices. In this regard, we empirically analyzed a number of popular vocabularies and ontology development efforts with respect to their use of guidelines and common practices. Based on this analysis, we identified the following main aspects of common practices: documentation, internationalization, naming, structure, reuse, validation and authoring. In this paper, these aspects are presented and discussed in detail. We propose a set of practices for each aspect and evaluate their relevance in a study with vocabulary developers. The overall goal is to pave the way for a new paradigm of vocabulary development similar to Software Development by Convention, which we name *Vocabulary Development by Convention*.

1 INTRODUCTION

Standards are powerful means to realize interoperability among heterogeneous systems. The process of defining them is usually as follows. Interested stakeholders come together and decide to define a new standard for a given case. They build a consortium which drives this process and they create a standardization organization or they create a subgroup within an existing one. In periodic meetings, representatives from the different stakeholders come together, communicate their particular needs, and try to find a consensus. If the outcome, which can be e.g a protocol, a component specification, or a vocabulary[1], is at a satisfying maturity level a specification document will be released. The adopters will implement this standard. Possibly, they need to adapt their already existing systems and the overall process is generally cumbersome and long lasting. The more stakeholders are involved and the more existing proprietary products are affected the more is this process exacerbated.

The dynamic World Wide Web, on the contrary, demonstrates that with a minimalistic standard set and flexible *de facto* standards interoperability is also possible to some extend. This is mainly enabled by focused applications and well documented specification pages. In some cases these *de facto* standards become real standards. However, the main idea is that they are not created in a top-down approach as in traditional standardization activities. Concretely, the implementation is not based on a predefined standard, but the standard is based on the adoption and the experience with existing implementations. That makes them more practical and avoid overly engineered standards like *CORBA* (Common Object Request Broker Architecture), *CAMEL* (Customised Applications for Mobile networks Enhanced Logic), etc. We consider this as a bottom-up approach for defining a standard.

Even within the Web context the danger of overly engineered standards is also given. The vision of the Semantic Web for example, caused the enthusiastic creation of standards like the *Web Ontology Language* and the *Rule Interchange Format* to represent knowledge and rules. It remains questionable if and when this standards will be really broadly adopted and if they are really practical enough to be used in various information systems. In contrast, positive examples likes *Schema.org*[2] clearly demonstrate that a practice-oriented approach is very effective. The definition, implementation and the usage is integrated pragmatically and not organized sequentially. In fact, the au-

[1] In this paper, we will use both terms 'vocabulary' and 'ontology' interchangeably.

[2] http://schema.org/

Grangel-González, I., Halilaj, L., Coskun, G. and Auer, S..
Towards Vocabulary Development by Convention.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 334-343
ISBN: 978-989-758-158-8

thors of this paper are convinced that being practice-oriented is the key success factor in this regard.

Therefore, we investigated into the applicability of the *Convention over Configuration* paradigm, which is very well-known and broadly adopted in software engineering, to vocabulary development. It aims at reducing the number of decisions that developers need to make, so they can focus on the main development. Inspired by this paradigm and the broad adoption of vocabularies like *Schema.org*, we propose a set of practices. These practices will represent the *Convention* which will be part of a new paradigm called *Vocabulary Development by Convention*. We derive these practices from the study of well-known vocabularies as well as our own experience in vocabulary development process. The bottom-up and pragmatically best-practice oriented technique which we applied in this study is presented in detail. In addition, we validate our approach by means of a survey with experts on the field.

The remainder of this paper is organized as follows. In section 2 we derive a set of practices to be applied in vocabulary development. In section 3 an analysis of the most widely vocabularies is presented. In section 4 we propose our set of best-practices for vocabulary engineering. We validate the impact of our approach by gathering opinions of vocabulary developers section 5 and compare our work with the current state of the art section 6. We conclude in section 7, shedding light on the critical aspects presented and providing an outlook to meaningful extensions of this work.

2 METHOD

Approaching a task can be done in two different ways. Top-down starts from the abstract and elaborates the concrete. Whereas bottom-up starts from the concrete level and continues towards the abstract. From a logic perspective, the former corresponds to deductive reasoning. It starts with known facts that are considered as premises and seeks for conclusions. The latter, on the contrary, starts with a given set of statements and looks for premises that caused them.

In the context of defining a methodology for vocabulary development, a top-down approach starts with the facts that are known about the expected outcome, namely the vocabulary. From the different characteristics of it, a possible creation process is derived. In the next steps, a list of roles is created and a set of tools are developed or selected, which can be used within the different steps of the overall process. In fact, most ontology engineering methodologies have been created by applying this approach.

On the contrary, *Convention over Configuration* is a bottom-up software development paradigm, which aims at reducing the number of decisions that developers need to make. This approach inspired other works (Fiorelli et al., 2015; Meenakshi, 2015) due to its flexibility and success. A bottom-up approach to derive practices is supposed to start from the current state of practice and look for evidences that explain why people are doing what they are doing. The most common activities of the successful outcomes are then compiled as a set of best-practices. This is in fact the method we applied in this work. We advocate that there is no need for just another comprehensive methodology that is designed in detail in a top-down approach. Rather, we claim that in the meanwhile there are sufficient good examples to be analyzed and learnt from. For that reason, we empirically analyzed a number of popular vocabulary and ontology development efforts.

3 ANALYSIS OF WIDELY USED VOCABULARIES

We compiled a list of the 20 most widely used vocabularies. The selection was based on the following criteria. Firstly, a usage rate of more than 5% in all datasets of the Linked Data Cloud (Schmachtenberg et al., 2014) was considered. Based on this, 13 vocabularies were chosen. Secondly, we looked for recognized ontologies that contain best practices regarding documentation, dereferenceability and are used by independent data providers[3]. In this case, 3 ontologies were added. Finally, *Linked Open Vocabularies* (LOV)[4] was observed for mostly reused vocabularies. The outcome of this observation was 4 vocabularies. We considered these as the most successful vocabularies that build the ground for our analysis. We defined them as *authoritative vocabularies*. Therefore, *authoritative vocabularies* have been revised and used for many years and also the community recognized that they are built on good practices (Schober et al., 2009). For that reason we believe that studying them will provide a better understanding of the common features and best practices of current vocabulary development. In this regard, we wanted to understand important aspects of vocabulary creation such as reuse, internationalization, documentation and naming as well as the implicit structure of these vocabularies (e.g. use of logical axioms, prop-

[3] http://www.w3.org/wiki/Good_Ontologies

[4] http://lov.okfn.org/dataset/lov/

erty domain/range definitions).

With respect to **Reuse**, 80% of the vocabularies make use of vocabulary elements defined elsewhere and 57% reuse elements from at least two external ontologies. This shows a considerable presence of the reuse aspect in the studied cases. One of the most important aspects of **Internationalization** (I18n) is the support for multi-linguality. In vocabularies this can be implemented by providing textual values for properties such as `rdfs:label`, `rdfs:comment` in different languages (using different language tags for RDF string literals). In 70% of the vocabularies we encountered explicit English literals (*@en*). In 15% of cases we found a translation of the terms into other languages and the remaining 15% there were no explicit language tags used at all. Consequently, despite I18n being important for existing ontologies we discovered that the most common practice is to support only English.

Documentation refers to the addition of human readable labels and descriptions (using the properties `rdfs:label`, `rdfs:comment`) to the vocabulary elements (i.e. classes, properties and individuals). We encountered that `rdfs:label` or `rdfs:comment` are present in 86% of the cases. It is worth noting that the combination of the two above mentioned elements with `rdfs:isDefinedBy` is used with a frequency of 57%. Only in one case (i.e. 5%) we did not find any form of documentation. This shows that documentation (i.e. `rdfs:label`, `rdfs:comment` for commenting, and `rdfs:isDefinedBy` for linking definitions) is widely used by existing vocabularies.

Another important practice in vocabulary creation is the convention for **Naming** elements. The *CamelCase* notation was with 60% of the cases the most used one. In all other cases (i.e. 40%) no homogeneous naming convention could be identified. A combination of CamelCase notation, underscore or dash sign were used instead.

We performed an statistical analysis regarding the inclusion of **domain and range axioms** for properties. By using the *Shapiro-Wilk* test over the observations of the object properties, domain and range axioms we encountered that the data do not follow a normal distribution for these variables. Our hypothesis was that there is a correlation in the obtained data regarding the amount of object properties and the domain and range axioms. To check for a correlation, we computed the *Spearman* rank coefficient. For the amount of object properties and domain as well as range axioms we obtained a value of 0.91 and 0.95 respectively. This indicates a strong correlation between object properties as well as domain and range axioms. The results for various vocabularies are illustrated in Figure 1 and Figure 2. The y-axis is transformed to log scale for a better comprehension. We performed the same process between data properties and domain as well as range axioms. In this case, we obtained 0.93 for both. These observations favor the conclusion that object properties and data properties should contain domain and range axioms. We also calculate the percentage for **inverse properties** (60%) and **class disjointness** (50%). These data indicate that the above mentioned axioms should be more carefully analyzed regarding the domain but are still important when building a vocabulary.

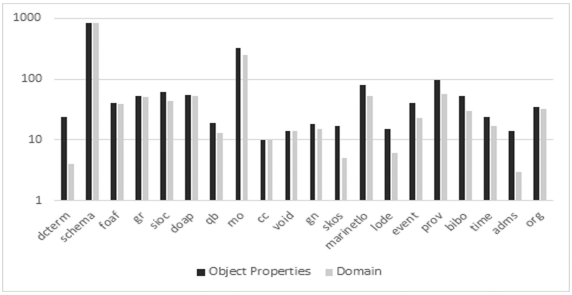

Figure 1: Relation of the amount of object properties and domain axioms.

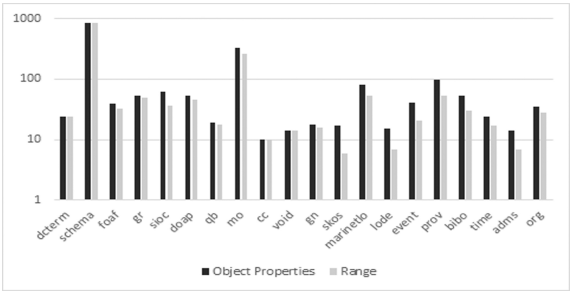

Figure 2: Relation of the amount of object properties and range axioms.

4 VOCABULARY DEVELOPMENT PRACTICES

In this section, we provide a comprehensive list of practices for vocabulary development. These practices are also available at http://eis-bonn.github.io/vdbc/. We derived this list from our own experience in creating vocabularies like *SCORVoc*[5] and *MobiVoc*[6] in combination with the results of the aforementioned analysis in section 3. Documentation, structure and validation aspects are mainly derived based on our experience. The other

[5]http://purl.org/eis/vocab/scor
[6]http://www.mobivoc.org/

Table 1: Authoritative Vocabularies.

Name	Prefix	Domain
Friend Of A Friend http://xmlns.com/foaf/0.1/	*foaf*	Terms related to Persons (i.e. Agent, Document, Organization, etc).
Dublin Core ontology Terms http://purl.org/dc/terms/	*dcterms*	General metadata terms (i.e. Title, Creator, Date, Subject, etc).
WGS84 Geo Positioning http://www.w3.org/2003/01/geo/wgs84_pos#	geo	Represents longitude and altitude information in the WGS84 geodetic reference datum.
Socially Interconnected Online Communities ontology http://rdfs.org/sioc/ns#	*sioc*	Aspects of online community sites (i.e. Users, Posts, Forums, etc).
Simple Knowledge Organization System Namespace http://www.w3.org/2004/02/skos/core#	skos	Data model for sharing and linking knowledge organization systems.
Vocabulary of Interlinked Datasets http://rdfs.org/ns/void#	*void*	Metadata about RDF datasets (i.e. Dataset, Linkset, etc).
Biographical information http://vocab.org/bio/0.1/.html	*bio*	Biographical information about people, both living and dead.
Data Cube Vocabulary http://purl.org/linked-data/cube#	*qb*	Statistic data (i.e. Dimensions, Attributes, Measures, etc).
Vocabulary for Rich Site Summary http://purl.org/rss/1.0/	*rss*	Models the declaration for Rich Site Summary (RSS) 1.0.
Vocabulary for modeling abstracts things for people http://www.w3.org/2000/10/swap/pim/contact#	*w3con*	General concepts about people everyday life (i.e Address, Phone, etc).
Description of a Project http://usefulinc.com/ns/doap#	*doap*	Terms for Open Source Projects (i.e. Version, Repository, etc).
Bibliographic Ontology http://purl.org/ontology/bibo/	bibo	Citations and bibliographic references (i.e. quotes, books, articles, etc).
Data Catalog Vocabulary http://www.w3.org/ns/dcat#	dcat	Facilitate interoperability between data catalogs published on the Web.
Schema.org http://schema.org	*schema*	Broad schema of concepts (i.e. Events, Organization, Person, etc).
GoodRelations http://purl.org/goodrelations/v1	gr	E-Commerce related terms (i.e. Products, Services, Locations, etc).
Music Ontology http://purl.org/ontology/mo/	*mo*	Terms related to music (i.e. Artists, Albums, Tracks, etc).
Creative Commons schema http://creativecommons.org/ns	*cc*	Describes copyright licenses (i.e. License Properties, Work Properties, etc).
GeoNames http://www.geonames.org/ontology	gn	Geospatial semantic information (i.e. Population, PostalCode, etc).
MarineTLO ontology http://www.ics.forth.gr/isl/ontology/MarineTLO/	marinetlo	Marine domain (i.e. Species, Marine Animal, etc).
Event Ontology http://purl.org/NET/c4dm/event.owl	event	Describes reified events (i.e. Event, location, time, ect).

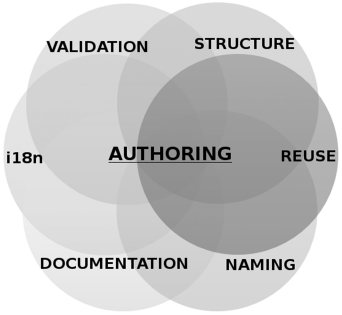

Figure 3: Main aspects of Vocabulary Authoring.

aspects are obtained from the conducted analysis in combination with the state of the art. These practices will serve as guidelines that help to focus on the most important aspects of vocabulary creation process. Therefore, it is expected to increase the efficiency of the collaboration and to improve the overall quality of the vocabulary. Figure 3 depicts the main aspects of our approach, which are described in detail in the remainder of this section. However, these guidelines are independent of the concrete development environment. They can be applied within various circumstance.

4.1 Reuse

Currently, in vocabulary construction, the reuse of existing terms is an aspect of vital importance (Poveda-Villalón, 2012; Pedrinaci et al., 2014). The main idea is not to create new terms but to utilize those that are present in the existing vocabularies and to avoid redundant work. Apart from saving time and investment costs, ontology reuse is expected to ensure a certain level of quality. The reason for this is that the longer an ontology exists and is reused, the more review processes it has gone through. Additionally, according to (Heath and Bizer, 2011) reuse is considered to be a best-practice in vocabulary construction. Therefore, in the following we discuss important practices regarding reuse.

P-R1 Reuse of Authoritative Vocabularies. We define authoritative vocabularies as vocabularies (cf. section 3) which are: (1) published by renowned standardization organizations; (2) used widely in a large number of other vocabularies; and (3) defined in a more domain independent way addressing more general concerns. Reusing authoritative vocabularies will increase the probability that data can be consumed by applications (Schober et al., 2009). Hence, these most widely used vocabularies should be considered as a first option for reuse (cf. Table 1).

P-R2 Reuse of Non-authoritative Vocabularies. Search online resources, such as vocabulary registries like *LOV*[7] and LODStats[8] or ontology search engines

[7] http://lov.okfn.org/dataset/lov/
[8] http://lodstats.aksw.org/

like *Swoogle*[9] and *Watson*[10] to find terms to reuse. The output of this process is a set of terms. For instance, by searching in LOV for a specific term the following information can be derived: (1) the number of datasets that uses it; (2) the number of occurrences of the term in all datasets; and (3) the reuse frequency of the vocabulary to which the term belongs (Pedrinaci et al., 2014). Also, the semantic description and definition of the term should be checked in order to verify whether it fits the intended use. The above information supports the decision process regarding to which terms are better candidates for reusing.

P-R3 Avoid Semantic Clashes. If the term has a *strong* semantic meaning for the domain, different from the existing ones, then a new element should be created.

P-R4 Individual Resource Reuse. Especially elements from authoritative vocabularies should be reused as individual vocabulary elements. For non-authoritative vocabularies a reuse of individual identifiers is less recommendable and the creation of own vocabulary elements with a possible alignment (cf. P-R6) or the reuse of larger modules (cf. P-R5) should be considered.

P-R5 Vocabulary Module Reuse. (Opposite of P-R4) Often vocabularies require certain basic structures such as addresses, persons, organizations, which are already defined in non-authoritative vocabularies. Such structures comprise usually the definition of one or several classes and a number of properties. If the conceptualizations match the complete reuse of a whole module should be considered.

P-R6 Establishing Alignments with Existing Vocabularies. Instead of the strong semantic commitment of reusing identifiers from non-authoritative vocabularies, alignments using `owl:sameAs`, `owl:equivalentClass`, `owl:equivalentProperty`, `rdfs:subClassOf`, `rdfs:subPropertyOf` can be established.

4.2 Vocabulary Structure

When a vocabulary grows in size and complexity the difficulty in the development and the maintenance processes increase. In this regard, modularization is a possible solution because it allows to divide huge vocabularies in logical and convenient way. Modularizing ontologies is an important aspect of vocabulary development (Suarez-Figueroa et al., 2012). (Poveda-Villalón, 2012) describes an ontology module as a loosely coupled and self-contained component of an ontology that keeps relationships with other ontology modules. Even though in some cases ontology modules are considered to be independent ontologies (dAquin et al., 2008), from the development perspective components are not treated as independent elements. Organizing a vocabulary in files where each file represents a module, is a way of managing modularity within the development process. Furthermore, some reports show that a module in a mid-sized vocabulary should contain between 200 and 300 lines of code (Schlicht and Stuckenschmidt, 2006). Since modularity depends on the overall size of the vocabulary, we propose the following three possibilities to structure and organize the files with respect to modularity.

P-S1 One File for the Whole Vocabulary. When the vocabulary is small (e.g. contains less than 300 lines of code) and represents a domain which cannot be divided in sub domains, it should be saved within one single file. If the number of contributors is relatively small and the domain of the vocabulary is very focused, organizing it into one single file might be possible, even if it exceeds 300 lines of code. However, if the comprehensibility is exacerbated, splitting it into different files should be considered (P-S2).

P-S2 Multiple Files. If the vocabulary contains more than 300 lines of code or if it covers a more complex domain, it should be organized into different subdomains. When the subdomains themselves are small enough they should be represented by different files within the parent folder. In this case, domain experts can contribute independently by modifying modules according to their field of expertise.

P-S3 Multiple Files and Folders. In case of very large vocabularies comprising complex domains, splitting the whole vocabulary into files is not sufficient. This would lead to a large amount of files within a single folder. Therefore, the subdomains should be represented by folders if they are large enough to be split into different components represented by different files. In this case, the folder and file structure should reflect the complex hierarchy of the overall domain.

4.3 Naming Conventions

Following naming conventions has a high impact in vocabulary development (Schober et al., 2012). Naming conventions help to avoid lexical inaccuracies and increase the robustness and exportability, specifically in cases when vocabularies should be interlinked and aligned with each other (Schober et al., 2009). The utilization of meaningful names increases the robustness of context-based text mining for automatic term

[9] http://swoogle.umbc.edu/

[10] http://watson.kmi.open.ac.uk/

recognition and ease the manual and automated integration of terminological artifacts (i.e. comparison, checking, alignment and mapping) (Svátek and Šváb-Zamazal, 2010; Schober et al., 2012).

Considering the literature on this topic (Schober et al., 2009; Montiel-Ponsoda et al., 2011) and the results of section 3 we propose some practices to be followed in the process of naming elements in vocabularies. For vocabulary construction, the use of the CamelCase notation is a considered as a best practice (Svátek et al., 2009). Our study also indicated the presence of this notation in 62% of the cases. Therefore, we propose the observation of this specific notation to be used in vocabulary construction.

P-N1 Concepts as Single Nouns. Name all concepts as single nouns using CamelCase notation (i.e. *PlanReturn*).

P-N2 Properties as Verb Senses. Name all properties as verb senses also following CamelCase approach. The name of an property should not normally be a plain noun phrase, in order to clearly distinct from class names (i.e. *hasProperty* or *isPropertyOf*).

P-N3 Short Names. Provide short and concise names for elements. When natural names contain more than three nouns, use the `rdfs:label` property with the long name and a short name for the element. For instance, for *ManageSupplyChainBusinessRules* use *BusinessRules* and set the full name in the label. In order to explain the context (i.e. Supply Chain), complement this label with the `skos:altLabel` (cf. subsubsection 4.7.1).

P-N4 Logical and Short Prefixes for Namespaces. Assign logical and short prefixes to namespaces, preferable, with no more than five letters (i.e. `foaf:XXX, skos:XXX`).

P-N5 Regular Space as Word Delimiters for Labeling Elements. For example, `rdfs:label "A Process that contains.."`.

P-N6 Avoid the Use of Conjunctions and Words with Ambiguous Meanings. Avoid names with "And", "Or", "Other", "Part", "Type", "Category", "Entity" and those related to datatypes like "Date" or "String".

P-N7 Use Positive Names. Avoid the use of negations. For instance, instead of *NoParkingAllowed* use *ParkingForbidden*.

P-N8 Respect the Names for Registered Products and Company Names. In those cases is not recommended the use of CamelNotation. Instead, the name of the company or product should be used as is (i.e. SAP, Daimler AG).

4.3.1 Dereferenceability

One of the four rules to be followed during vocabulary development is naming things with HTTP URIs[11]. Adopting HTTP URIs for identifying things is appropriate due to the following reasons: (1) it is simple to create global unique keys in a decentralized fashion and (2) the generated key is not used just as a name but also as an identifier.

By combining dereferenceability with content negotiation[12], the server will provide adequate content for a resource based on the type of request. There are three different strategies to make URIs of resources dereferenceable: (1) slash URIs; (2) hash URIs and (3) a combination between them[13].

P-D1 Use Slash URIs. When the client request a resource from server by providing its URIs, the server response will be *303 see other*. Slash URI should be used when dealing with large datasets. This makes the server to response only with requested resource. For example, the ChargingPoint resource is identified as follows *http://purl.org/net/mobivoc/ChargingPoint*. The URI of turtle representation of above resource is *http://purl.org/net/mobivoc/ChargingPoint.ttl* and the URI of html representation is *http://purl.org/net/mobivoc/ChargingPoint.html*.
In order to get information about ChargingPoint, the client provides URI and specify request type. In turn server response will be *303 see other* by redirecting to appropriate representation.

P-D2 Use Hash URIs. This solution is formed by including a fragment to the URIs as in the following format *URI#resource*. Use hash URIs when dealing with small datasets. This will reduce number of HTTP round trips. For instance, the URI of the *ScorVoc*[14] vocabulary is *http://purl.org/eis/vocab/scor*. The URI of the *Process* resource is *http://purl.org/eis/vocab/scor#Process*.

P-D3 Use Combination between Slash and Hash URIs. This allows a large dataset to be split into multiple fractions. Use this solution when datasets may grow to some point where it is not practical to serve all resources in single document(e.g. *http://purl.org/eis/vocab/scor/Process#this*).

4.4 Multilinguality

Providing multilingual ontologies is desirable but not an straightforward issue (Gracia et al., 2012). According to our empirical analysis in section 3 and with

[11] http://www.w3.org/DesignIssues/LinkedData.html
[12] http://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html
[13] http://www.w3.org/TR/cooluris
[14] http://purl.org/eis/vocab/scor

the aim to keep things simple we propose the following best-practices.

P-M1 Use English as the Main Language. Use English for every element and explicitly set with the *@en* notation.

P-M2 Multilinguality for other Languages. In order to add another language, use another line adding the same format for every element. The following example illustrates this practice with translations for the class *SupplyChain*.

```
scor:SupplyChain rdf:type owl:Class ;
    rdfs:label   "SupplyChain"@en;
    rdfs:comment "A Supply Chain is a ..."@en ;
    rdfs:label   "Lieferkette"@de;
    rdfs:comment "Eine Lieferkette ist ..."@de.
```

This approach should be followed with all the elements starting from the basics ones like `rdfs:label` and `rdfs:comment` but also for the external annotation properties (i.e. `skos:prefLabel`).

4.5 Documentation

Providing user friendly view of vocabularies for non-experts is crucial for integrating Semantic Web with everyday Web (Peroni et al., 2013). It facilitates contribution of domain experts during the development process. In addition, it helps other interested parts for easy use of vocabulary in later phases as well. There exists different tools for documentation generation. Basically, these tools requires that following information should be present for each resource to enable generation process.

P-Do1 Use of rdfs:label and rdfs:comment. Add a `rdfs:label` to every element setting the main name of the concept that is being represented and `rdfs:comment` to describe the context for which the element is created.

P-Do2 Generate Human-readable Documentation. Easy-to-use documentation is critical for the wide adoption of the vocabulary. There exist two different types of URIs (c.f. 4.3.1). If during vocabulary creation slash URIs are used for identifying resources then tools like *Schema.org* documentation generation should be used for documentation generation. Tools like *Parrot*[15] are appropriate if hash URIs or combination between slash and hash URIs are used for identifying resources.

4.6 Validation

Validation is an important aspect in the ontology development process (Poveda-Villalón et al., 2012). It analyzes whether ontology correctly represents the knowledge domain in accordance to user requirements and best practices (Gómez-Pérez et al., 2006; Kezadri and Pantel, 2010). Criteria used for validation activity are: (1) correctness; (2) completeness and (3) consistency (Suárez-Figueroa, 2010). With the purpose of addressing the above mentioned criteria, we propose the following practices.

P-V1 Syntax Validation. When collaborating directly on the vocabulary source code, syntax checking is of paramount importance. Ideally, syntax checking is directly integrated into the editor and committing the code with errors is not possible. For example, tools like *Rapper*[16] or Web-based services such as the *RDF Validation Service*[17] or *OWL2 Validator*[18] can be used for finding common typos and syntax errors.

P-V2 Code-Smell Checking. Code smells are symptoms in the software source code that possibly indicate deeper problems. Similarly tools such as *OOPS*[19] can be used for vocabulary smell checking. OOPS is a Web-based tool for detecting common ontology pitfalls such as: (1) missing relationships; (2) using incorrectly ontology elements and (3) missing domain and range properties. The complete list of pitfalls that are detected by OOPS is presented in (Poveda-Villalón et al., 2012).

P-V3 Consistency Checking. Since we deal with lightweight ontologies it is not very likely to have axioms that produce semantic inconsistencies. Nevertheless, our analysis in section 3 showed that in authoritative vocabularies there are cases that lead to semantic inconsistencies (i.e. class disjointness). Handling inconsistencies impacts the quality of ontologies (Abburu, 2012). Tools like *Pellet*[20], *Fact++*[21], *Racer*[22], *HermiT*[23] or the Web based tool *ConsVISor*[24] should be used for consistency checking.

P-V4 Linked Data Validation. Tools such as *Vapour*[25] verify whether data are correctly published according Linked Data principles and the best publishing practices[26].

[15] https://bitbucket.org/fundacionctic/parrot/wiki/Home

[16] http://librdf.org/raptor/rapper.html

[17] http://www.w3.org/RDF/Validator/

[18] http://mowl-power.cs.man.ac.uk:8080/validator/

[19] http://oops.linkeddata.es/

[20] http://clarkparsia.com/pellet

[21] http://owl.man.ac.uk/factplusplus/

[22] https://github.com/ha-mo-we/Racer

[23] http://hermit-reasoner.com/

[24] http://vistology.com/OLD/www/consvisor.shtml

[25] http://validator.linkeddata.org/vapour

[26] http://http://www.w3.org/TR/swbp-vocab-pub/

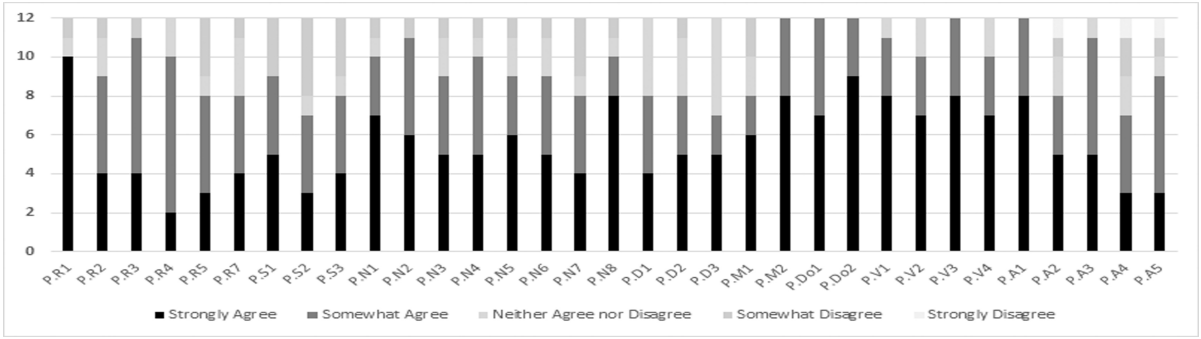

Figure 4: Evaluation results for the practices in Vocabulary Development Process.

4.7 Authoring

In section 3 we analysed common practices followed by vocabulary engineers (i.e. the creation of object properties and their associated domain and range axioms). Those practices are always domain dependent, but still can serve as general guidelines to be followed in the process of designing vocabularies.

P-A1 Domain and Range Definitions for Properties. When creating a property, consider to provide the associated domain and range definitions. This also means that in case of object properties the corresponding classes should be defined. In case of datatype properties, the range should be a suitable datatype.

P-A2 Avoid Inverse Properties. Create inverse properties only if it is strictly necessary to have bidirectional relations (i.e. `invalidated` and `wasInvalidatedBy`). Inverse properties affect the size as well as the complexity of the vocabulary.

P-A3 Use of Class Disjointness. Use class disjointness to logically avoid overlapping classes. Even though disjointness has been used in authoritative vocabularies, it should be carefully examined because it can easily lead to semantic inconsistencies.

4.7.1 Utilization of SKOS Vocabulary

The Simple Knowledge Organization System *SKOS* is a W3C recommendation for modeling vocabularies in the Web. *SKOS* is currently used by at least 478 vocabularies (Haslhofer et al., 2013). The utilization of some *SKOS* constructs is considered a best practice for declaring and documenting indexing terms (i.e. `skos:prefLabel`) and alternatives terms (i.e. `skos:altLabel`) (Manaf et al., 2012; Baker et al., 2013). Both above mentioned properties are subproperties of `rdfs:label`. *SKOS* provides a more detailed notion of the labeling concept, which can be useful for better descriptions of the terms.

P-A4 Provide `skos:prefLabel` to Complement the Labeling of Concepts. Use `skos:prefLabel` in combination with `rdfs:label` to complement the semantic label of the element. For instance, `skos:prefLabel` might describe a shorter definition for a concept than `rdfs:label`.

P-A5 Use `skos:altLabel` to Describe Variations of the Elements. Add complementary descriptions for the elements such as acronyms, abbreviations, spelling variants, and irregular plural/singular forms by using `skos:altLabel`.

5 SURVEY AND RESULTS DISCUSSION

With the goal to validate the proposed practices we performed a survey for vocabulary developers.[27] The experience in the selected group is as follows, 58% have up to two years and 41% from two to five years. The *Likert Scale* (Boone and Boone, 2012) was used to collect the opinions. Figure 4 depicts the results of the survey. Generally, all practices have received good evaluations regarding to the opinion of experts. The authoring aspect was the most controversial one. The practice **P-A2**, received some negative opinion due to the existing debate regarding the use of inverse properties [28]. The results of **P-A4, P-A5** show that even *SKOS* as a generally accepted standard still is not well received for a certain group of vocabulary developers.

6 RELATED WORK

Currently, there exist several methodologies for de-

[27] https://goo.gl/X8otxe

[28] https://lists.w3.org/Archives/Public/public-vocabs/2014Apr/0200.html

Table 2: Comparison with existing approaches regarding the Aspects for Collaborative Vocabulary Development.

	Reuse	Structure	Naming	i18n	Documentation	Validation	Authoring
METHONTOLOGY(Fernández-López et al., 1997)	Yes	No	No	Yes	Yes	Yes	No
Constructing Reusable Ontologies (Annamalai and Sterling, 2003)	Yes	No	No	No	No	No	No
DILIGENT (Pinto et al., 2004)	Yes	Yes	Yes	No	No	No	No
On-To-Knowledge (Sure et al., 2004)	No	Yes	Yes	No	No	Yes	No
RapidOWL (Auer and Herre, 2007)	No	No	Yes	No	No	No	Yes
JEOE (Di Maio, 2011)	Yes	No	No	No	Yes	Yes	Yes
Linked Data Patterns (Dodds and Davis, 2011)	Yes	No	No	Yes	Yes	No	Yes
NeOn Methodology (Suárez-Figueroa, 2010)	Yes	Yes	No	Yes	No	Yes	No
Methodology for semantic model development (Zeginis et al., 2013)	Yes	No	No	No	No	Yes	No

veloping ontologies (Fernández-López et al., 1997; Pinto et al., 2004; Auer and Herre, 2007; Di Maio, 2011; Suarez-Figueroa et al., 2012; Zeginis et al., 2013). Generally, the methodologies cover the main aspects for ontology development with a top-down approach. Specific practices to address how to perform the ontology engineering process regarding the reusing, multilinguality, modularization are still missing. On the other hand, there exist some guidelines and best practices for vocabulary development (Annamalai and Sterling, 2003; Dodds and Davis, 2011). Despite these guidelines follow a bottom-up approach they do not cover all the aspects mentioned in this paper. One central characteristic of our practices is that they will support the developers when taking design decisions for vocabulary creation. Thus, they can be seen as pragmatic design criteria to build vocabularies. Despite we recognized the design criteria presented in (Gruber, 1995) our goal is to support the development process by defining specific tasks and how those task can be realized. Table 2 shows some of the existing guidelines and methodologies for developing vocabularies regarding the aspects covered in our approach. To the best of our knowledge, there is no existing work that comprises all the mentioned aspect for Vocabulary Development in a usable and pragmatic way.

7 CONCLUSION

In this paper, we considered creating standards through heavyweight processes within standardization organizations as the legacy approach to tackle the problem of data integration among heterogeneous systems. Driven by the success of the Web vocabularies and the ever-increasing awareness of the importance of data, we advocate that it is now time to rethink how this problem should be addressed. For this reason, we identified the most important vocabularies, which we call authoritative vocabularies. We analyzed their commonalities in terms of the key aspects reuse, structure, documentation, multilinguality, naming, validation and some practices regarding authoring. These aspect were identified as the most important ones during our own work on vocabulary creation. The rationale for this analysis was to derive best-practices and conventions for vocabulary development. The overall goal is to pave the way for a new paradigm of vocabulary development similar to Software Development by Convention, which we name *Vocabulary Development by Convention*. The applied bottom-up approach is in contrast to related work in the field of knowledge and ontology engineering. Usually, methodologies are designed in a top-down approach, without considering guidelines that are used to realize specific aspects in the development process. In this regard, they are similar to standardization activities which tend to be over-engineered and lead to the wasting resources. Regarding future work, we plan to extend the results presented in this paper in various directions. Initially, we envision to study all existing vocabularies in LOV. The purpose is to generalize our results as well as to include different indicators related to vocabulary structure like the correlation with respect to number of classes, properties etc. This action will lead us to a better understanding of the existing practices of vocabulary creation and, as an outcome, derive better conventions. Additionally, we plan to create a tool to automatically support some of the practices that we have proposed in this paper. Finally, we want to create a light-weight *Vocabulary Development by Convention* methodology that includes practices for collecting the domain knowledge.

ACKNOWLEDGEMENTS

This work has been supported by German BmBF project LUCID (http://www.lucid-project.org).

REFERENCES

Abburu, S. (2012). A survey on ontology reasoners and comparison. *Int. Journal of Computer Applications*, 57(17):33–39.

Annamalai, M. and Sterling, L. (2003). Guidelines for constructing reusable domain ontologies. In *OAS*, pages 71–74.

Auer, S. and Herre, H. (2007). Rapidowlan agile knowledge engineering methodology. In *Perspectives of systems informatics*, pages 424–430. Springer.

Baker, T., Bechhofer, S., Isaac, A., Miles, A., Schreiber, G., and Summers, E. (2013). Key choices in the design of simple knowledge organization system (skos). *Journal of Web Semantics*, 20:35–49.

Boone, H. N. and Boone, D. A. (2012). Analyzing likert data. *Journal of Extension*, 50(2):1–5.

dAquin, M., Haase, P., Rudolph, S., Euzenat, J., Zimmermann, A., Dzbor, M., Iglesias, M., Jacques, Y., Caracciolo, C., Aranda, C. B., et al. (2008). Neon formalisms for modularization: Syntax, semantics, algebra. *Deliverable D1*, 1.

Di Maio, P. (2011). 'just enough'ontology engineering. In *Int. Conf. on Web Intelligence, Mining and Semantics*, page 8. ACM.

Dodds, L. and Davis, I. (2011). Linked data patterns. *Online: http://patterns.dataincubator.org/book*.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

Fiorelli, M., Pazienza, M. T., and Stellato, A. (2015). A flexible approach to semantic annotation systems for web content. *Intelligent Systems in Accounting, Finance and Management*, 22(1):65–79.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2006). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media.

Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P., and McCrae, J. (2012). Challenges for the multilingual web of data. *Journal of Web Semantics*, 11:63–71.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *Int. journal of human-computer studies*, 43(5):907–928.

Haslhofer, B., Martins, F., and Magalhães, J. (2013). Using skos vocabularies for improving web search. In *22nd Int. Conf. on World Wide Web companion*, pages 1253–1258.

Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

Kezadri, M. and Pantel, M. (2010). First steps toward a verification and validation ontology. In *KEOD*, pages 440–444.

Manaf, N. A. A., Bechhofer, S., and Stevens, R. (2012). The current state of skos vocabularies on the web. In *The Semantic Web: Research and Applications*, pages 270–284. Springer.

Meenakshi, S. (2015). Ruby on rails â [euro]" an agile developer's framework. *Int. Journal of Computer Applications*, 112(1).

Montiel-Ponsoda, E., Vila Suero, D., Villazón-Terrazas, B., Dunsire, G., Escolano Rodríguez, E., and Gómez-Pérez, A. (2011). Style guidelines for naming and labeling ontologies in the multilingual web.

Pedrinaci, C., Cardoso, J., and Leidig, T. (2014). Linked usdl: a vocabulary for web-scale service trading. In *The Semantic Web: Trends and Challenges*, pages 68–82. Springer.

Peroni, S., Shotton, D., and Vitali, F. (2013). Tools for the automatic generation of ontology documentation: a task-based evaluation. *Int. Journal on Semantic Web and Information Systems (IJSWIS)*, 9(1):21–44.

Pinto, H. S., Staab, S., and Tempich, C. (2004). Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *16th European Conf. on Artificial Intelligence (ECAI 2004)*, volume 110, page 393.

Poveda-Villalón, M. (2012). A reuse-based lightweight method for developing linked data ontologies and vocabularies. In *The Semantic Web: Research and Applications*, pages 833–837. Springer.

Poveda-Villalón, M., Suárez-Figueroa, M. C., and Gómez-Pérez, A. (2012). Validating ontologies with oops! In *Knowledge Engineering and Knowledge Management*, pages 267–281. Springer.

Schlicht, A. and Stuckenschmidt, H. (2006). H.: Towards structural criteria for ontology modularization. In *ISWC 2006 Workshop on Modular Ontologies*. Citeseer.

Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *ISWC 2014*, pages 245–260. Springer.

Schober, D., Smith, B., Lewis, S. E., Kusnierczyk, W., Lomax, J., Mungall, C., Taylor, C. F., Rocca-Serra, P., and Sansone, S.-A. (2009). Survey-based naming conventions for use in obo foundry ontology development. *BMC bioinformatics*, 10(1):125.

Schober, D., Tudose, I., Svatek, V., and Boeker, M. (2012). Ontocheck: verifying ontology naming conventions and metadata completeness in protégé 4. *J. Biomedical Semantics*, 3(S-2):S4.

Suárez-Figueroa, M. C. (2010). *NeOn Methodology for building ontology networks: specification, scheduling and reuse*. PhD thesis, Informatica.

Suarez-Figueroa, M. C., Gómez-Pérez, A., and Fernandez-Lopez, M. (2012). The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer.

Sure, Y., Staab, S., and Studer, R. (2004). On-to-knowledge methodology (otkm). In *Handbook on ontologies*, pages 117–132. Springer.

Svátek, V. and Šváb-Zamazal, O. (2010). Entity naming in semantic web ontologies: Design patterns and empirical observations. Znalosti.

Svátek, V., Šváb-Zamazal, O., and Presutti, V. (2009). Ontology naming pattern sauce for (human and computer) gourmets. In *Workshop on Ontology Patterns*, pages 171–178.

Zeginis, D., Hasnain, A., Loutas, N., Deus, H., Fox, R., and Tarabanisa, K. (2013). A collaborative methodology for developing a semantic model for interlinking cancer chemoprevention linked-data sources. *Semantic Web Journal*.