What Is Concept Drift and How to Measure It?

Shenghui Wang[1,2], Stefan Schlobach[2], and Michel Klein

[1] Department of Communication Science
[2] Department of Computer Science
Vrije Universiteit Amsterdam

Abstract. This paper studies concept drift over time. We first define the meaning of a concept in terms of intension, extension and label. We then introduce *concept drift* over time and two derived notions: *(in)stability* over a time period and *concept shift* between two time points. We apply our framework in three case-studies, one from communication science, on DBPedia, and one in the legal domain. We describe ways of identifying interesting changes in the meaning of concept within given application contexts. These case-studies illustrate the feasibility of our framework in analysing concept drift in knowledge organisation schemas of varying expressiveness.

1 Introduction

Knowledge organisation systems (KOS), such as formal ontologies (*e.g.* modelled in OWL), thesauri or taxonomies (*e.g.* described in SKOS) or other term classification schemes, play an crucial role in providing semantic interoperability in many domains and use cases. They have become critical to the Web of Data, for structured access of documents in libraries or patient records based on diagnostic information, and many more applications. In almost all modern types of KOS, *concepts* are the central constructs that are used to describe sets of objects with shared characteristics. Although it is widely recognised to be an oversimplification most current systems consider their underlying KOS to be stable over time. For many applications, this starts to be a critical problem, and this paper attempts to provide a first step towards a better understanding of what we call *concept drift*.

Problem description: As the world is continuously changing, concepts also change over time. That is, for example, a concept refers to different objects at different points in time. The term Government of the Netherlands refers to different people in 1999 and in 2009. Consider the concept Middle class which is interpreted very differently in various periods of time.[1]

To our knowledge there has been no formalisation of what concept drift actually means and implies. In order to identify different types of changes in concepts

[1] This drift in meaning occurs not only over time, but also over location, culture, *etc.* For ease of presentation we will mostly refer to drift in time, but significant parts of the framework should extend to other kinds of "contexts."

P. Cimiano and H.S. Pinto (Eds.): EKAW 2010, LNAI 6317, pp. 241–256, 2010.

and to understand the impact of concept drift, such a formalisation is critical. Therefore, this paper focuses on the following research questions:

RQ1 What is concept drift, and how to formalise it?
RQ2 Can we identify the impact of concept-drift?

Methodology: We provide a generic formalisation of the meaning of concepts in terms of label, intension and extension. These definitions are not intended to provide new philosophical insights, but aim at making existing accepted notions applicable in practice. For each of the three elements of concept meaning we define *concept drift* and study two important consequences: the *(in)stability* over a time period and *concept shift* between time points (where part of the meaning of a concept shifts to some other concept).

Experiments: We instantiate our framework in three case-studies, studying concept drift in a SKOS vocabulary used by communication scientists for political analysis, a general purpose RDFS ontology, DBpedia and a legal OWL ontology, LKIF-Core. We investigate the introduced mechanisms for studying concept drift in these three different KR models. Our experiments show the feasibility of both the formalisation and identification mechanisms by pointing to some examples of concept (in)stability and shift which were identified as relevant by collaborating domain experts.

Contributions: The paper should be read as an attempt to turn established (philosophical) insights into a general pragmatic framework. We believe that we also contribute to a better understanding of temporal change of meaning in formal knowledge organisation schemes and its impact in practical applications.[2] We motivate and define the crucial notions of *drift, shift* and *stability.* In three case-studies we show that our findings are relevant, most particular in our main case-study in communication science.

2 A Theory of Concept Drift

The meaning of concepts changes over time. Let us first commit to some basic definitions regarding the meaning of concepts. The *intension* of a concept are the properties implied by it, the *extension* the set of things it extends to. We also consider the *labelling* as a part of the meaning of a concept, as the way people reference a concept is crucial in studying concept drift. Labels do not refer to a unique identifier but to a natural language description used to convey the meaning of a concept from one human to another.[3]

[2] Discussions with "only" philosophical relevance will usually be dealt with in footnotes to improve the flow of the story-line.

[3] We try to be consistent with common philosophical approaches. We apply and formalise the standard distinction between intension and extension which goes back to [1]. We include, somewhat more unconventionally, the labelling in the meaning of a concept (in the tradition of the *signifier* [2]).

The meaning of concepts. Our definition of the meaning of a concept, and its drift, should be generic enough to be applied in different ontological frameworks. In this paper, we apply our idea to a set of concepts used for annotating documents in communication science as well as one RDFS ontology and one OWL ontology. We start out from a set of objects referred to as the universe of the domain, and a set of properties (unary predicates). Both universe and properties depend on the application and the formalism used.

Definition 1. *The* meaning of a concept C *is a triple* $(label(C), int(C), ext(C))$, *where* $label(C)$ *is a string,* $int(C)$ *a set of properties (the* intension *of* C*), and* $ext(C)$ *a subset of the universe (the* extension *of* C*).*

All elements of the meaning of a concept can change. Still it makes sense to talk about a concept being the same over time. Our solution to this problem is based on the rigid part of the intension of concepts.[4] Formally, we assume that the intension of a concept C is the disjoint union of a rigid and a non-rigid set of properties (*i.e.* $(int_r(C) \cup int_{nr}(C)))$. This separation between rigid and non-rigid properties does not need to be explicitly specified, but rigidity is crucial for *identity* of a concept over time. Intuitively, this amounts to the assumption that a concept is uniquely identified through some core properties that do not change over time.[5]

Definition 2. *Two concepts* C_1 *and* C_2 *are considered identical if and only if, their rigid intension are equivalent,* i.e. , $int_r(C_1) = int_r(C_2)$.

Identity allows us to compare two variants of the same concept at different moments in time even if the meaning (either label, extension or the non-rigid part of its intension) has changed. We will assume that there is always only one variant of a concept at each moment in time, *i.e.* , only one concept at a moment can be identical to a concept at another moment.

Concept drift. If a concept at different times has the same meaning, there is no concept drift. A more subtle notion of concept drift, however, requires notions of similarity of meaning, which can be decomposed into intensional similarity, sim_{int}, which is calculated between sets of predicates, extensional similarity, sim_{ext}, between sets of objects, and label similarity, sim_{label}, between strings. Each similarity is a function with the range [0, 1], and a similarity value of 1 indicates an equality.

---

[4] Rigidity is discussed in Ontoclean [3] in a slightly different way. There rigidity is a meta-property of a concept that modellers should make explicit. We use it in a stronger way as an intrinsic and the only stable part of the meaning of a concept which otherwise can drift in various ways.

[5] This assumption implies that if the rigid core of a concept changes, the new concept will be a *different* concept. An example is demagogue which used to denote the concept of political leaders. The concept of a populist now referred to by the label "demagogue" is a different concept.

Definition 3. *A concept has extensionally drifted in two of its variants*[6] C' *and* C''*, if and only if,* $sim_{ext}(C', C'') \neq 1$*. Intensional and label drift are defined similarly.*

Concept shift and (in)stability. Concept drift happens regularly and even if it can be measured it is often difficult to grasp its impact. Therefore, we define other notions: *(in)stability* and *concept shift*. Both can be used to identify more drastic concept drift. The more the meaning of a concept drifts, the more *unstable* it becomes. Although there is no indication of when an unstable concept becomes *critically* unstable, instability can still be an interesting notion. First, one could define a threshold based on experience. Label similarity is often defined using edit distance and one could define lexically instability in terms of a high edit-distance. Another way of analysing concept drift over time is to compare the (average) stability of concepts. As a *relative* measure, it does not require any priori commitment (such as a threshold).

A special case of instability is when a concept becomes so unstable, that part of its meaning is more representative for a different concept rather than for itself. We call this *concept shift*.

Definition 4. *The meaning of a concept extensionally shifts between two of its variants* C' *and* C'' *if the extension of* C'' *is more similar to the extension of a non-identical concept rather than to the extension of* C'*. Intensional and label shift are defined similarly.*

Concept shift can have drastic consequences on the use of a concept in an application as some other concept has basically taken over its meaning.

Applying the framework. To apply our framework for concept drift in a specific use-case, the following steps are required:

1. to define intension, extension and a labelling function.
2. to define similarity functions over intension, extension and labels

Given that the mission of the Semantic Web includes giving meaning to resources on the Web, it could come as a suprise that defining intensions, extensions and even labelling functions is by no means trivial. The usual model-theoretic notions of extension and intension, *e.g.* for RDF(S) or OWL semantics are slightly misleading here, as they refer to specific models, whereas ontologies usually represent classes of models. In practice, one needs to define the relevant notions per use-case, where each such definition is an ontological commitment. In the following section, we will give such commitments for 3 different case-studies. It should be understood that such a commitment is never uncontroversial.

[6] This means that C' and C'' are identical but have different meaning at different moments in time.

3 Case-Studies

3.1 Case Study 1: Concept Shift in Political Reporting

Communication scientists annotate various media content with concepts from controlled vocabularies (of increasing expressiveness) in order to quantitatively study the influence of the Media on the political processes. Such controlled vocabularies have recently been represented using the SKOS model [4]. Each concept has a preferred Label and possibly a few alternative Label which are the synonyms of this concept. One concept can be linked to the others using skos:broader, skos:narrowed and skos:related.

In this case study, we focus on five variants of a SKOS vocabulary of political concepts used during five most recent Dutch national election campaigns. All newspaper articles on Dutch politics during these campaign periods were manually annotated with the concepts from the particular variant of that year. In our case, these articles can be considered as the instantiation of the abstract political concepts.

Political concepts and their meaning. We now formally define our problem: for each election campaign $t \in \{1994, 1998, 2002, 2003, 2006\}$ we have a set Δ_t of sentences annotated by concepts from a SKOS vocabulary V_t.

The label of a concept is obtained using the SKOS Core labelling property skos:prefLabel. The extension $ext_s(C_t)$ of a concept $C_t \in V_t$ at time t is the set of all sentences annotated by C_t, *i.e.* ,

$$ext_s(C_t) = \{s \in \Delta_t \mid \mathsf{annotatedBy}\ C_t\}.$$

It is more difficult to formally define the intension of a concept, as there is no explicit intensional definition of the concepts available. We construct an explicit intension based on co-occurrence of concepts in annotations. For each concept C, we calculate the top K concepts $topKuse(C)$ which co-occur the most in the sentences they code in one moment in time. The properties we use to define the intension are based on "topicality", *i.e.* , a property $P_C(D)$ is true if, and only if, D is in the $topKuse$ relation with C. The intension of C is then the set of properties $int_s(C) = \{P_C(D) = \text{true}\}$, *i.e.* , the intension is in fact determined by all associated concepts.

Similarity of intension, extensions and labels. Similarity of labels can be determined through standard Levenshtein edit distance. In our case we define similarity as 1 minus the hyperbolic tangent of the original edit distance. For each concept C, we average over the Levenshtein edit distance $ed(label_s(C_t), label_s(C))$ between the preferred labels of all its variants C_t in each year.

Extensional similarity is usually determined by calculating the overlap of the extensions. In our case, this is not possible, as the set of sentences in different years are disjoint. In order to use disjoint extensions to measure the similarity of concepts from different years, we applied the mapping tool which was developed in [5]. For each sentence in Year1, the mapping tool first looks for the most

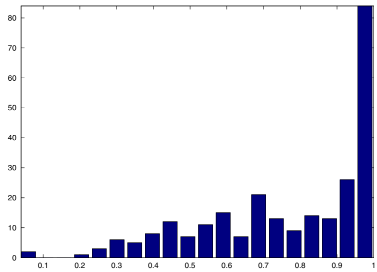

Fig. 1. Label stability: The X-axis gives the average label similarity over the years; the Y-axis the number of concepts with this average.

similar sentence in Year2. Then this sentence of Year1 is considered to be coded by the concept(s) with which its most similar sentence of Year2 is coded. In this way, two disjoint extensions become dually annotated, and we then measure the similarity between two concepts in terms of their common extensions.

Since the intension of a concept is determined by its associated concepts, the intensional similarity between two concepts is therefore determined by the set similarity between the sets of concepts with which they are associated. We use the Jaccard similarity for this purpose.

For each concept C, we calculated the above three kinds of similarity between all pairs of variants C_t in each year. We take the average as the measure of the corresponding *stability* of this concept over time, noted as S_{label}, S_{int} and S_{ext}.

Note that these similarity measures can only provide relative ranking whether one concept is more stable than another. A lower similarity indicates a higher instability. In the following experiments, we use automated methods to evaluate instability. Concept shift, on the other hand, is harder to quantify and we study it using selected examples.

Experiments to study concept drift in our political ontology. The identity problem is solved, in our case, by the manual concept mapping provided by a communication science expert, based on the rigid part of the intention. In this way, we are sure whether two concepts in different years are actually the two variants of a single concept. This enables us to investigate the concept drift in terms of the labels, intension and extension.

Identifying label instability and shift. Since the domain expert has indicated identical concepts across different years, our question is to see whether these "intensionally" identical concepts have "stable" labels.

Figure 1 shows the histogram of the label stability values, indicating that most (over 80) concepts have very stable labels ($S_{\ label}(C) = 1$) over these years. For example, all five variants of the concept Asielzoekers (asylum seekers) has exactly the same label. However, some concepts do have very unstable labels. According to the domain expert, the following 6 concepts are intensionally identical:

1994_sjo_creawetsto → 1998_wcorruptie (corruption) → 1998_rbeursfraude (stock fraud) →2002_belangenverstrengeling (conflict of interest) →
2003_corruptie (corruption)→ 2006_fraude_en_corruptie (fraud and corruption)

The instability of its label over the years is striking, and it results in a label stability value of $S_{label} = 0.47$.

Obviously, concepts with the same label across different years can have different intensions or, more precisely, the difference of their non-rigid parts of intension is so big that the whole concept shifts to another concept with a different label, which we call *label shift*. For example, when newspapers used "openbaarheid" (openness) in 2002, they meant "public sphere," which is an area in social life where people can get together and freely discuss and identify societal problems, and through that discussion influence political action. While, in 2003, they uses the same label to mean "open government" in particular. These two concepts with the same label are actually different concepts. Therefore, Concept 1994_openbaarheid has a shift in label as its label shifts to another concept, according to Definition 4. Therefore we claim that the meaning of Concept openbaarheid shifts between 1994 and 1998.

Identifying extensional shift. As stated before, by looking at the extension of the concepts, we can detect whether the extension of one concept shifts over time or whether two concepts shift towards each other. For one concept, we calculate the similarity between its extension and the extension of the concepts from the following year. The concept with the highest similarity is considered to be the *extensionally identical* one. If this identity is consistent with the intensional identity provided by the domain experts, then there is no concept shift; otherwise, the concept has shifted to some other concept(s).

In Table 1, we compare the "extensionally" identical concepts and the "intensionally" identical ones. The first column gives the number of concepts which have found their extensionally identical concepts and also have an "intensionally" identical concept according to our domain expert. The second column is the number of concepts when extensional and intensional identities are consistent. As it shows, the consistency between these two identities is rather low.

We have to take into account the reliability of the calculated extensional similarity. Since it uses an approximation of the real extension, we therefore relax the criterion of identity consistency by introducing the parameter K: if the intensionally identical concept is within the top K most extensionally similar concepts, then we consider the two identities to be consistent. The third column

Table 1. Extensional shift vs intensional stability

Year	Extensional	Consistent	Consistent_at_5
1994–1998	41	8	19
1998–2002	43	9	16
2002–2003	23	8	11
2003–2006	92	35	58

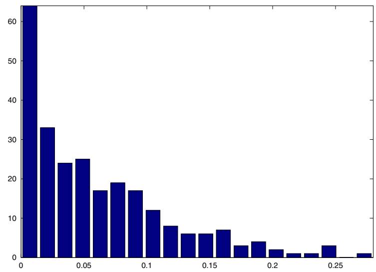

Fig. 2. Intensional stability: the X-axis lists the average intensional similarity between variants of a concept The Y-axis shows the number of concepts with this average.

gives the number of consistent concepts if $K = 5$. The degree of consistency increases, however, there is still a big inconsistency between these two identities.

As said, the validity of this identity is unfortunately not guaranteed. The traditional evaluation method is to compare it with some gold standard provided by human experts. However, it is not applicable here since both identities are under investigation in our study. Because of the lack of reliable extensional identity, we could not measure the (in)stability at the concept level as we did in terms of labels.

Although an automated stability analysis is not possible, we can still manually identify some real extensional shift. For example, according to the domain expert, 2003_kinderopvang (childcare) is the same as 2006_kinderopvang. However, 2006_gratis_kinderopvang (free childcare) is more similar to 2003_kinderopvang in terms of their extension. In this case, we say 2003_kinderopvang has shifted its meaning towards a more specific topic, namely free childcare. This is also confirmed by the post-hoc analysis of our domain experts.

Identifying intensional stability and shift. As we described earlier, we use the association to other concepts of the same year as an indicator of the intension of a concept. By measuring the similarity between such associations, we can get some information about how intensionally stable one concept is over the years.

Figure 2 gives the histogram of the measure of intensional stability (S_{int}). Now most concepts have a rather unstable intension over the years. On one hand, comparing to the label stability (Figure 1), few concepts have a stable intension, which suggests that the label of concepts is more stable then their intension. On the other hand, the instability of intension is far beyond our expectation, which suggests that we should look for another way of formalising concept intension, as the reliability of the results provided by the current formalisation is doubtful.

Nevertheless, Figure 3 and Figure 4 respectively give an example of very unstable and very stable concepts.[7] Here, the red links are the intentionally identical concepts provided by domain experts, the black links are the association links (*i.e.* , the concepts which co-occur the most to annotated sentences) and

[7] For convenience, we translate all the Dutch labels into English.

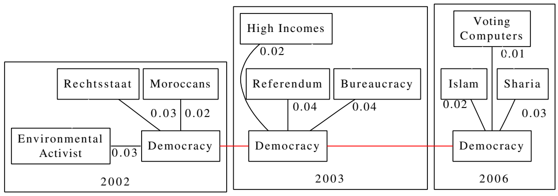

Fig. 3. Intension of concept Democracy in 3 years, with average drift of ($S_{int} = 0.02$)

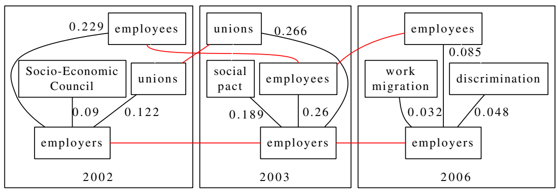

Fig. 4. Intension of concept Employers in 3 years, with average drift of ($S_{int} = 0.15$)

the number next to the links are the strength of the association. As the S_{int} value indicates, Concept Employers is rather stable over the years, while Concept Democracy seems to shift its meaning in different years. This observations are also consistent with the political reality. An interesting point observed from these two figures is that, when one concept is stable, its closely associated concepts tend to be stable too, while the concepts closely associated with an unstable concept tend to be also unstable.

3.2 Case-Study 2: Concept Drift in DBpedia

DBpedia[8] is probably the most successful ontology currently linked within the Linked-Open Data (LOD) cloud. It combines a hand-crafted class hierarchy with automatically generated instance data taken from the Wikipedia effort. Through its high quality and huge coverage DBpedia is now the most strongly linked dataset within the LOD. For the sake of this research we consider the the DBpedia ontology in RDFS [6], *i.e.* , we ignore the (very few) OWL operators used in the model. RDF(S) and its underlying semantics is a very common modeling framework, which makes DBpedia an interesting object of study as it is almost exclusively modeled in RDF(S).

RDF(S) concepts and their meaning. RDFS comes with a specific labeling relation rdfs:label. Furthermore, RDF is equipped with a rdf:type relation that relates objects with classes. It seems natural to define the extension of an RDF class to be the set of all instances in the rdf:type relation. The intension of a class is on the other hand not specifically defined in RDF(S). We have chosen a simple

[8] `http://wiki.dbpedia.org/`

Table 2. Four version of the DBpedia ontology

Version	#Concept	#Resource
3.5	255	1,477,377
3.4	204	1,161,678
3.3	174	1,054,199
3.2	174	875,273

approach which focusses on the "semantic" operators in RDFS with fixed semantics, more precisely rdfs:subclass, rdfs:range, rdfs:domain. The intension of a concept C is then simply the set of all triples with C in the subject or object position of these three types of triples.

Let us define the meaning of a DBpedia concept formally. We will call the combination of the DBpedia terminology T[9] and the explicit type information as well as the relations translated from Wikipedia, the DBpedia ontology.

Definition 5. *Let O be the DBpedia ontology,* i.e. *a set of triples* (s, p, o)*, and* O^* *the semantic closure of* O*. The* rdf-label $lab_r(C)$ *of* C *is defined as the object of the* $(C, \text{rdfs:label}, o)$*. The* rdf-extension $ext_r(C)$ *of* C *is defined as the set of resources* r *such that* $(r \text{ rdf : type } C) \in O^*$*. The* rdf-intension $int_r(C)$ *of* C *is defined as the set of all triples* $(C, p, o) \in O^*$ *in* O *where* $p =$rdfs:subclass *and* (s, p, C)*, where* $p \in \{$rdfs:subclass, rdfs:domain, rdfs:range$\}$.

Please note that this definition is just one possible choice of ontological commitment regarding the meaning of a concept in RDF(S).

Similarity relations between labels are defined on the basis of string similarity. Similarity between intension and extension, which are just sets of resources and triples respectively, can easily be defined through the set similarity (*e.g.* Jaccard).

Experiments to study concept drift in DBpedia. We studied the four latest versions of the same DBpedia ontology, namely, 3.5, 3.4, 3.3 and 3.2. The Table 2 gives some general information of these four versions. These versions were uploaded into independent RDF repositories.

The identity problem is solved by the use of unique URI references that remain stable over these versions. Our basic assumption is that two concepts in different versions are actually the two variants of a single concept. This enables us to investigate concept shift and instability in terms of their labels, intension and extension.

Although most DBpedia concepts define their label using rdfs:label, these labels are mostly equivalent to the local-names of their URIrefs. These abels are strongly related to identify, and thus less interesting to study label drift.

For each concept C at year t, we built its extension $ext_r C_t$ (*i.e.* , the set of instances) and the intension $int_r C_t$ (*i.e.* , the set of related triples). The Jaccard similarity[10] was measured between the intension and extension of the different

[9] The terminology is called differently in the different DBpedia versions, but usually something like `dbpedia-ontoloy.owl`.

[10] `http://en.wikipedia.org/wiki/Jaccard_index`

Table 3. The top 5 most stable and ustable DBpedia concepts in terms of their extension and intension (of the 167 concepts present in all four versions)

Rank	Extensional	Intensional	Rank	Extensional	Intensional
1	Planet	SportsEvent	163	OfficeHolder	Vein
2	Road	FormulaOneRacer	164	Politician	BasketballPlayer
3	Infrastructure	WineRegion	165	City	EthnicGroup
4	Cyclist	Cleric	166	College	Band
5	LunarCrater	WrestlingEvent	167	ChemicalCompound	BritishRoyalty

variants of the same concept. In the end, we calculate the measure of intensional stability ($S_{int}(C)$) and extensional stability ($S_{ext}(C)$). Table 3 gives the top 5 most stable and the last 5 least stable concepts from these two aspects.

Concept Politician is considered extensionally very unstable. This can easily be confirmed by the change of the sheer amount of instances. In Version 3.2, it has only 476 instances, while in Version 3.5, it has already 19,285 instances. The extension of this concept clearly has expanded significantly. However, low stability not necessarily leads to concept shift. For example, although growing, the extension of Politician is always the most similar to the extension of its following variant.

Concept City is also very unstable extensionally. In Version 3.4 it has indeed shifted to another concept in Version 3.5, Settlement. These two concepts share more than 73% instances, which causes a high extensional similarity between them, which is higher than the similarity between the two variants of City. We found that Settlement appeared only in version 3.5. For some reason, most of the instances of City in version 3.4 have been transferred to Settlement. This poses an interesting question to the modeler whether this is intentional or by mistake.

Similarly, studying the intensional stability and shifts also gives insight to the evolution of the intensional semantics of a concept. For example, the intensionally very unstable concept, EthnicGroup, has been involved in the rdfs:domain and rdfs:range of a continuously changing set of properties. This indicates this concepts are related to different concepts in different versions, which contributes to the intensional instability. Furthermore, real shifts happened to some concepts. The identified extensional shift, from City to Settlement is also found to be an intensional shift, that is, these two concepts not only share a lot of instances, but their intensional definitions are very similar too. This double-confirmation is valuable because it may well indicate a genuine concept shift.

3.3 Case-Study 3: Concept Drift in LKIF-Core

In this section we look at the evolution of concepts in the LKIF-Core, an OWL ontology of basic legal concepts.[11] This ontology has also been continuously developed, and uses most of OWL's expressiveness.

[11] http://ontology.leibnizcenter.org/trac/wiki/LKIFCore

OWL concepts and their meaning. The formal meaning of a concept in an OWL DL ontology is often far more explicitly defined than in other formalisms. The intension of a concept could potentially be defined as the set of all possible DL concepts that are equivalent to it *wrt.* the ontology. Unfortunately, this is only possible in so-called definitorical terminologies, and difficult to calculate even if possible. We will approximate this set in our experiments (rather coarsely) by using the OWLIM[12] interpretation of LKIF, and consider finite sets of consequences (triples-chains). As OWL ontologies are specifications of sets of possible models, there is no unique notion of the extension of concepts. However, once committed to a particular model, DL semantics provide the formal instance-of relation to specify the extension of a concept.

The LKIF ontology does not come with instances, so that we do not consider extensional drift in this paper.

Let us define the meaning of concepts in LKIF.

Definition 6. *Let O to be the LKIF-OWL ontology and O^* denote the OWLIM inferred semantic closure. The owl-label $lab_o(C)$ of C is defined as the object of the $(C, \mathsf{rdfs{:}label}, o)$. The owl-intension $int_o(C)$ of C is defined:*

1. *all triples $(C, p, o) \in O^*$ and $(s, p, C) \in O^*$*
2. *all triples in chains $\{(C, p_1, o_1) \circ (s_2, p_2, o_2) \circ \ldots, \circ(s_n, p_n, o_n)\}$ where $s_k = o_k - 1$, plus*
3. *all triples in chains $\{(s_1, p_1, o_1) \circ (s_2, p_2, o_2), \circ, \ldots, \circ(s_n, p_n, C)\}$ where $s_{k+1} = o_k$ being blank nodes.*

Again, the above definition is only one possible ontological commitment regarding the meaning of an OWL concept. Based on such definition, the similarity of label and intension can be calculated using set similarity, as done for the DBpedia case.

Experiments to study concept drift in LKIF. We studied 4 major versions of LKIF, namely, 1.0, 1.0.2, 1.0.3 and 1.1. Similarly, the local-name in the URI reference is used as the identity of one concept. Unfortunately, the rdfs:label actually was rarely used; only 4 concepts specify their labels which stay constant for all variants. Therefore, we focus on the intensional stability and shift, which we calculate as before based on Jaccard similarity. All concepts were ranked according to its intensional stability. The ranked list has been confirmed by one of the developer of LKIF to be consistent with his expectations.

By comparing the intension of concepts between different versions, we were also able to find true concept shifts, listed in Table 4.

As Table 4 shows, some intensional shift corresponding to shifting in modules, for example, Speech_Act is no longer a general action, instead it belongs to the expression module for describing, propositions and propositional attitudes (belief, intention), qualifications, statements and media. While the other kind of shift, for example, Mental_Concept to Mental_Entity, were confirmed to be a renaming operation.

[12] `http://www.ontotext.com/owlim/`

Table 4. Examples of confirmed intensional shift in LKIF-Core

lkif1.0:action.owl#Speech_Act	lkif1.0.2:expression.owl#Speech_Act
lkif1.0:action.owl#Termination	lkif1.0.2:process.owl#Termination
lkif1.0.2:lkif-top.owl#Mental_Concept	lkif1.0.3:lkif-top.owl#Mental_Entity
lkif1.0.2:lkif-top.owl#Physical_Concept	lkif1.0.3:lkif-top.owl#Physical_Entity

Our three case studies have shown the feasibility of both the formalisation and identification mechanisms in analysing concept drift in knowledge organisation schemas of varying expressiveness.

4 Related Work

Let us first look at research in other domains that is related to our notion of *concept drift.* In historical linguistics, *semantic shift* describes the evolution of word usage. Each word has multiple senses and connotations which can be added, removed or altered over time. Semantic change is a change in one meaning of a word. Semantic shift can be triggered by different forces and have different types [7], but this interpretation of "semantic" does not say anything about the meaning and change of the underlying concepts.

In machine learning *concept drift* addresses a similar problem [8]. The term *concept* refers to the quantity that a learning model is trying to predict, *i.e.* the variable. *Concept drift* is the situation in which the statistical properties of the target concept change over time. This requires regular updates in the predicting model itself. A special case is *virtual concept drift* [9] (or *sampling shift*), in which the meaning of a concept does not change, while in the latter case only the data distribution changes. An example is the concept "spam", for which the meaning does not change, but the data distribution (*i.e.* the relative frequency of the properties) is changing.

In 1994, Klenner and Hahn [10] discuss exactly the problem of concept drift because of evolving notions over time, however, not in the context of Semantic Web applications but for technical standards. As a mechanism for updating static value restrictions or integrity constraints, they propose an automatic procedure. This generates a generational stratification of the underlying level of generic concepts in terms of concept versions; single instances are then related to their associated concept version. The procedure exploits a so called progress model—provided by an expert—which describes in qualitative terms the regularities of foreseeable changes of attributes in a domain. Versions are then detected by measuring the change in values of attributes of instances.

With the goal of detecting concept drift and the occurrence of new concepts in a domain, Fanizzi *et.al.*describe the use of a conceptual clustering technique based on unsupervised learning [11]. In their approach, a clustering method is used to hierarchically organize groups of similar instances. Concept drift is detected by finding new individuals that are too far apart from existing clusters, but that together do not form a new cluster. If the unclustered instances do form a cluster, a new concept has occurred.

Takahira Yamaguchi also discusses concept drift in the context of ontologies [12]. This paper focuses on constructing domain ontologies starting from a hierarchically structured set of domain concepts without concept definitions (machine readable dictionary). This initial ontology is then refined by adding domain-specific knowledge; the places in the ontology that have to change because of this new knowledge are seen as places where concept drift occurs. Two specific strategies for such changes are presented. The paper does not define concept drift but merely uses this term for a step in an ontology construction methodology.

In the Semantic Web community, the problem addressed here is related to a broader problem of *ontology change*, which refers to the "problem of deciding the modifications to perform upon an ontology in response to a certain need for change as well as the implementation of these modifications and the management of their effects in depending data, services, applications, agents or other elements" [13]. The existing research on ontology evolution and versioning mainly addresses this problem at the macro ontological level, that is, the effect of certain change operations over the ontology elements, including concepts, relations and instances, as well as the interoperability issue between different variants (versions) over time. For example, [14] formally defines ontology perspectives, which describe the relation between versions of an ontology and its extension. An exception might be the work of [15] who study the intensional change of concepts in different versions of ontologies,

On the specific meaning of change for specific concepts not much has been done. In [16] a series of "concept signatures" extracted from the textual definitions of the same concept at different time are used detect drifts. This definition-based method is applicable if there is rich definitions of concepts and the definitions are constantly modified. In the Ontoclean framework [3] the meta-properties identity and rigidity are defined with relate to the stability of a concept. In [17], a distinction is made between the "specification" and the "conceptualisation" of a concept.

5 Conclusion

More and more applications critically depend on some kind of concept schemes for the semantic interoperability of their data. However, although it is recognised by many as a critical problem, the continuous change in meaning of concepts (called drift in this paper) has not yet received the attention it deserves in the ontology modelling community. Despite the significant efforts that have gone into topics such as ontology evolution, semantic versioning or temporal modelling and reasoning, most tools are still based on static representations. The existing ontology versioning frameworks focus on the interoperability between versions and data. There is not yet a formal framework for concept drift, nor an implementation for identifying significant concept drift.

This paper attempts to close this gap by introducing a theoretical foundation for the notions **drift, shift** and **stability** over time. We show that the proposed mechanisms are useful in practical applications modeled in SKOS, RDFS and OWL respectively. The results of our evaluation are preliminary, but encouraging: although intensional drift is difficult to study because the concepts are often not formally defined, the detected concept shift and stability ordering on concepts gives useful information for the domain experts.

Future research will be directed in two directions: first, in cooperation with Communication Scientists working on political reporting and legal experts we will apply the proposed methods in more in-depth studies on meaning change. With the experience that will be gained in at least one of these additional use-cases we plan to create a generic implementation for analysing concept drift.

References

1. Frege, F.L.G.: über sinn und bedeutung. Zeitschrift für Philosophie und philosophische Kritik 100, 25–50 (1892)
2. Saussure, F.D.: Course in General Linguistics. Open Court Classics (1986) (reedition)
3. Guarino, N., Welty, C.: Evaluating ontological decisions with ontoclean. Commun. ACM 45, 61–65 (2002)
4. Isaac, A., Summers, E.: SKOS Primer. W3C Group Note (2009)
5. Schopman, B., Wang, S., Schlobach, S.: Deriving concept mappings through instance mappings. In: Proceedings of the 3rd Asian Semantic Web Conference, Bangkok, Thailand (2008)
6. W3C: RDFS, `http://www.w3.org/TR/rdf-schema/`
7. Bloomfield, L.: Language. Allen and Unwin (1933)
8. Tsymbal, A.: The problem of concept drift: definitions and related work. Technical Report TCD-CS-2004-15, Computer Science Department, Trinity College Dublin, Ireland (2004)
9. Widmer, G., Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 227–243. Springer, Heidelberg (1993)
10. Klenner, M., Hahn, U.: Concept versioning: A methodology for tracking evolutionary concept drift in dynamic concept systems. In: Proc. of ECAI 1994, pp. 473–477. Wiley, Chichester (1994)
11. Fanizzi, N., d'Amato, C., Esposito, F.: Conceptual clustering and its application to concept drift and novelty detection. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 318–332. Springer, Heidelberg (2008)
12. Yamaguchi, T.: Constructing domain ontologies based on concept drift analysis. In: IJCAI 1999. Workshop on Ontologies and Problem-Solving Methods (1999)
13. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: classification and survey. Knowledge Eng. Review 23, 117–152 (2008)

14. Heflin, J., Pan, Z.: A model theoretic semantics for ontology versioning. In: Third International Semantic Web Conference, pp. 62–76. Springer, Heidelberg (2004)
15. Huang, Z., Stuckenschmidt, H.: Reasoning with multi-version ontologies: A temporal logic approach. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 398–412. Springer, Heidelberg (2005)
16. Gulla, J.A., Solskinnsbakk, G., Myrseth, P., Haderlein, V., Cerrato, O.: Semantic drift in ontologies. In: Proceedings of 6th International Conference on Web Information Systems and Technologies, Valencia, Spain (April 2010)
17. Klein, M.: Change Management for Distributed Ontologies. PhD thesis, Vrije Universiteit Amsterdam (August 2004)