Analysing Ontological Structures through Name Pattern Tracking

Ondřej Šváb-Zamazal and Vojtěch Svátek

University of Economics, Prague, Dept. Information and Knowledge Engineering,
Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
ondrej.zamazal@vse.cz, svatek@vse.cz

Abstract. Concept naming over the taxonomic structure is a useful indicator of the quality of design as well as source of information exploitable for various tasks such as ontology refactoring and mapping. We analysed collections of OWL ontologies with the aim of determining the frequency of several combined name&graph patterns potentially indicating underlying semantic structures. Such structures range from simple set-theoretic subsumption to more complex constructions such as parallel taxonomies of different entity types. The final goal is to help refactor legacy ontologies as well as to ease automatic alignment among different models. The results show that in most ontologies there is a significant number of occurrences of such patterns. Moreover, their detection even using very simple methods has precision sufficient for a semi-automated analysis scenario.

1 Introduction

Concept names in semantic web (OWL) ontologies with set-theoretic semantics are sometimes viewed as secondary information. Indeed, for logic-based reasoners, which are assumed to be the main customers exploiting these ontologies, anyhow cryptic URLs can serve well. Experience however shows that even in ontologies primarily intended for machine consumption, the naming policy is not (and should not be) completely arbitrary. It is important for ontology developers, maintainers, adoptors etc. to be able to see the semantic structure of a large part of the ontology at once, and ontology editors normally use base concept names (local URLs) and not additional linguistic labels within their taxonomy view. At the same time, while inspecting possibly complex OWL axioms, self-explaining concept names are extremely helpful.

This leads us to the hypothesis that concept naming in OWL ontologies can (at least in some cases) be a useful means for analysing their conceptual structure, detecting modelling errors and assessing their quality. Obviously, a 'true' evaluation of concept naming in specialised domain ontologies requires deep knowledge of the domain. We however assume that even in specialised ontologies, the 'seed' terms often belong to generic vocabulary and the domain specialisation is frequently achieved via adding syntactic attributes (such as adjectives, nouns in apposition or prepositional phrases), leading to multi-word terms. The occurrence of certain tokens in the names of multiple concepts (as well as other ontology constructs, in particular, properties) is strongly correlated with the graph structure of the ontology. Analysis of the graph structure and concept naming in combination may thus help reveal important semantic structures non-detectable by more formal methods (especially for ontologies less abundant with axioms). Deeper understanding of the structure of an ontology thus acquired can help in e.g. mapping it properly to other ontologies.

The paper is structured as follows. Section 2 explains the very simple lexical (to say, text string) analysis of concept naming used in our approach, namely, tokenisation and head noun detection. Section 3 presents the initial version of descriptive model for combined name&graph patterns in ontologies. Section 4 defines the four patterns we so far concentrated on. Sections 5 and 6, respectively, describe experiments for different patterns, which aimed both to verify their abundance in real-world ontologies and their reliability in detecting some 'semantic finding'. Sections 7 discusses the methodology used and results obtained in broad. Sections 8 is devoted to a survey of related projects. Finally, section 9 summarises the contributions and sets up directions for future work.

2 Lexical Analysis of OWL Concept Names

The vast majority of concept names in OWL can be, after *tokenisation*, interpreted as short noun phrases in singular form.[1] In our analysis, we focused on straightforward heuristic discovery of the unigram term that acts as *head noun* of the whole phrase and on which all other terms (tokens) are dependent. The approach used is extremely shallow from the point of view of NLP, but seems to work in a reasonably high[2] proportion of ontology concepts, given the restricted nature of concept naming and the syntactic regularity (fixed word order) of English.The following short subsections are devoted each to one of these steps: tokenisation and head noun detection.

2.1 Tokenisation

Tokenisation is, for short 'technical' items such as OWL concept names, usually assumed to rely on identification of one of a few delimiters. In our project we focused on the following three: underscore (`Concept_name`), hyphen (`Concept-name`) and change of lowercase letter to uppercase (`ConceptName`), which is most parsimonious and therefore most frequent. Although the semantics of these delimiters could in principle differ (especially the hyphen is likely to be used for more specific purposes than the remaining two, on some occasions), we treat them as equivalent for the sake of simplicity. We also ignore sub-string relationship without explicit token boundary (i.e. between two single-word expressions), assuming that they often deviate from proper subclass relationship (as in 'fly' vs. 'butterfly', or even worse e.g. 'stake' vs. 'mistake').

2.2 Head Noun Detection

The rule we used for detection of head noun can be summarised as follows:

[1] Plural form usually indicates a trivial and relatively harmless naming error.

[2] As we constantly estimated the accuracy of lexical analysis, as perceived while looking at its results, as much higher than that of the subsequent graph structure analysis steps (and not far from 100%), we did not carry out any quantitative evaluation so far. It should nevertheless be done in the future, for a representative sample of concepts from different ontologies.

1. If the name contains a *preposition* then the head noun is the token before the preposition (e.g. *Head*OfDepartment)
2. Otherwise the head noun is the *last* token in the name.

There were also a few modifying heuristics concerning frequent auxiliary words. Slight improvement could still be achieved by handling further, less frequent, generic structures such as verbs in *passive form* (typically following after the head noun and before a preposition) or *disjunctions* of terms expressed using the 'or' construct.

3 Generic Framework for Concept Name&Graph Structures

In the following we present a simple descriptive framework for representing ontology graph structures in combination with concept names. The formalisation is only preliminary and is likely to acquire more rigour in the future. It will also have to be extended to cover at least the notion of non-taxonomic relationship among concepts (i.e. OWL object property).

We start with notions related to tokens within a single concept name.

Definition 1 (Token Set and Head Noun of a Concept). *Given an ontology concept C:*

- *let $\tau(C)$, the* token set *of C, be the set of all distinct tokens from the name of C;*
- *let $T(C)$, the* head noun *of C, be the token from the name of C on which all other tokens are syntactically dependent in a natural-language interpretation.*

This definition is obviously rather imprecise, as it leaves open the question of tokenisation as well as syntactical analysis of the token sequence. For simplicity, we assume that tokenisation can be reliably done and the head noun identified using simple methods as described in the previous section, which indeed works for a majority of cases.

Furthermore, in order to be able to talk about *taxonomic structures*, let $\subset$ denote the *descendant-of* and $\subset_d$ the *child-of* (i.e. 'direct' descendant) relationship in an ontology, in the intuitive meaning (excluding equality).

Definition 2 (Structural Cluster). *A* structural cluster *$\mathcal{K}$ is a set of concepts from the same ontology O such that for any concept $C_i \in \mathcal{K}$ there exists a concept $C_j \in \mathcal{K}$ such that at least one of the following holds:*

- $C_i \subset_d C_j$
- $C_j \subset_d C_i$
- *there exists a concept P such that $C_i \subset_d P \ \wedge \ C_j \subset_d P$*

This notion of structural cluster is somewhat heuristic, as it postulates that a set of concepts can be viewed as 'cluster' if each concept in such a set is either parent, child or sibling of another concept from this set. One could certainly argue that e.g. the sibling relationship is not guaranteed to 'intuitively' convey the meaning of 'cluster' to all possible observers. There is also no size limit imposed, hence, a whole OWL/RDFS ontology typically satisfies the notion of structural cluster. However, for our name-pattern-oriented study, this working definition looks satisfactory.

Definition 3 (Descendant of a Structural Cluster). *Given a concept C and a structural cluster $\mathcal{K}$, C is* descendant *of $\mathcal{K}$, $C \subset \mathcal{K}$, if and only if there is a concept $C' \in \mathcal{K}$ such that $C \subset C'$.*

Intuitively, a descendant concept is 'under' the structural cluster in the taxonomy.

Now we will eventually combine the token notions with the structural notions.

Definition 4 (Token Set and Head Noun of a Structural Cluster). *Given a structural cluster $\mathcal{K}$, its* token set $\tau(\mathcal{K})$ *corresponds to the union of the token sets of all concepts $C \in \mathcal{K}$, and its* head noun $T(\mathcal{K})$ *is*

- *equal to the head noun t of all $C \in \mathcal{K}$ if and only if for all $C \in \mathcal{K}$ holds $T(C) = t$*
- *undefined otherwise.*

Structural cluster having a head noun will be referred to as named structural cluster.

For example, a cluster containing three concepts named *OneConcept*, *OneMoreConcept* and *StillOneConcept*, will have the head noun Concept and the token set {One, Concept, More, Still}.

Finally, we need the notion of shared as well as distinct token set of a cluster.

Definition 5 (Shared Token Set and Distinct token set of a Structural Cluster). *Given a structural cluster $\mathcal{K}$*

- *its* shared token set $\sigma(\mathcal{K})$ *corresponds to the set of tokens that are part of names of all concepts from $\mathcal{K}$:*
$$\sigma(\mathcal{K}) = \{t \mid \forall_{C \in \mathcal{K}}\ t \in \tau(C)\}$$
- *its* distinct token set $\delta(\mathcal{K})$ *corresponds to set of tokens that are part of names of some but not all concepts from $\mathcal{K}$, i.e. the complement of the shared token set to the union of all tokens appearing in $\mathcal{K}$:*

$$\delta(\mathcal{K}) = (\bigcup_{C \in \mathcal{K}} t \in \tau(C)) - \sigma\mathcal{K}$$

In the previous example, the shared token set would be {One, Concept}, and the distinct token set would be {More, Still}.

4 Patterns Considered in the Study

The model presented in the previous section is all we need in order to formally describe the patterns considered in our study. The four patterns were chosen based on our preliminary manual analysis of numerous ontologies, and thus correspond to generalisations of 'striking' fragments of real ontologies (the inventory of patterns is thus definitely not complete and will be extended by future research). For each of them, we present:

- a formal description of the pattern
- a verbal description of the pattern
- possible interpretations and derivation triggered for the pattern
- one or more examples.

4.1 Pattern I: Non-Matching Child

This very simple pattern represents the situation of a child that does not have the same head noun as its parent:

$$C \subset_d P \ \wedge \ T(C) \neq T(P)$$

The ‘non-matching child’ pattern has already been cared of in our previous work [13]. The pattern is connected with the hypothesis that the nature of underlying entity should not change while subclassing, so a change of the head noun would mean ‘some problem’. The derivation associated with the detection of this pattern is thus an ‘alerting’ one. Possible faults of the ontology manifested by the pattern generally fall under two groups:

- Fault in *set-theoretical semantics*: for example, a part-of relationship mistaken for subclass relationship (e.g. ‘Car/Wheel’)
- Improper style of *concept naming*, e.g. omission of the head noun in the child name (e.g. ‘Paper/Accepted’); unlike the previous one, this situation often occurs even in ontologies created by relatively skilled designers.

However, in its raw form, the derivation would have extremely low precision—most alerts would be false ones. Foremost, instead of strict token identity, we should extend the notion of ‘same head noun’ to ‘same semantic term’, including the thesaurus correspondence, namely the situations when the head noun of the child name is a *hyponym or synonym* of the head noun of the parent name.

Thesaurus correspondence would even deserve to be explicitly considered in the framework described in section 3; we only omitted it for the sake of simplicity of the model. In the experiments, we so far only considered thesaurus correspondence for Pattern I; it should later be considered for all four patterns in principle, although this will presumably lead to abrupt increase of computational complexity.

An example of non-matching parent-child pair that probably cannot be healed by thesaurus correspondence is ‘SuffrageLaw/RestrictedSuffrage’,[3] as there is an apparent set-theoretical incoherence between ‘Law’ and ‘Suffrage’ as the head nouns of the two concepts. An example of non-matching child with thesaurus correspondence is ‘JudicialOrganisation/AppealsCourt’ (as ‘Court’ is a hyponym of ‘Organisation’); such a case should not be understood as instance of the pattern proper. More such cases and the corresponding evaluation framework are introduced in section 5.

4.2 Pattern II: Matching Siblings with Non-Matching Parent

The pattern represents the situation that two children do not have the same head noun as their parent but have the same head noun among themselves:

$$C_1 \subset_d P \ \wedge \ C_2 \subset_d P \ \wedge \ T(C_1) = T(C_2) \neq T(P)$$

[3] This as well as the following example comes from the Government ontology stored in the DAML repository (`http://www.daml.org/ontologies/`).

This pattern is obviously a refinement of the previous one. However, it potentially has additional semantics to simple 'trouble-alerting' from the previous case: it might indicate an *overly flat* hierarchy, asking for inclusion of an intermediate concept superordinated to some of the sibling classes only. It can also be produced by a modelling error or by awkward naming. An example[4] is at Fig. 1: the head noun of the parent is Type while the head noun of both its children is Service. This can be seen at least as an improper naming style, if not a sign of wrong conceptualisation.

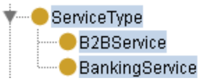

Fig. 1. Example of Pattern II

Again here, the inclusion of *thesaurus correspondence* would be likely to increase *precision*, as different head nouns in parent and children could be identified as synonyms or hypo/hypernyms. In addition, *recall* of the detection could be improved as well via identification of more sibling pairs with the same (thesaurus-mediated) head noun;[5] this would also help distinguish modelling/naming (strict head noun mismatch) errors from situations in which adding an intermediate concept to an overly flat taxonomy could be suggested (thesaurus-based head noun correspondence).

4.3 Pattern III: Matching Outlier

The pattern represents the situation that a concept shares the head noun with a cluster that it is not descendant of:

$$T(C) = T(\mathcal{K}) \ \land \ \neg(C \subset \mathcal{K})$$

Reasons for a concept being considered beyond a cluster of concepts presumably related to the same type of entity can be many.

In the example in Fig. 2, the culprit seems to be the mismatch between the related but non-synonymous notions of Accuracy and Consistency. The core of problem thus possibly has the form of Pattern I (DQ_TemporalAccuracy being parent of DQ_TemporalConsistency).

For larger ontologies, a taxonomic structure genuinely disconnected into multiple (individually valid) parts can also potentially arise by updates made by different designers or over a longer span of time (leading to loss of control over the editing process).

A somewhat different case could be polysemy or homonymy of a term. For example, in an ontology[6] we came across the concept *SugarRefinery* occurring disparately from the cluster of *Refinery* concepts related to chemistry. Such phenomena may not indicate

[4] From http://www.csl.sri.com/users/ton/ontologies/BankServicesPolicy.owl.

[5] However, exhaustive checking of thesaurus relationships for siblings would be computationally costly.

[6] http://www.daml.org/experiment/ontology/beta/military-elements-ont

lower quality of the ontology but may in turn bring interesting insights to the domain itself, possibly in view of extending it (here, say, by the relationship of refining some crude substance to a useful product).

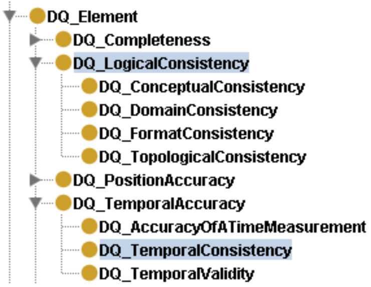

Fig. 2. Example of Pattern III

As token matching is considered positively ($T(C) = T(\mathcal{K})$ in the above formula), reflecting *thesaurus correspondence* would improve the *recall* of the pattern detection.

4.4 Pattern IV: Parallel Structure Candidate

This pattern is semantically rather different from the previous three, as it is not connected to ancestor/descendant relationships, and thus, consequently, does not rely upon the notion of head noun. It simply expresses that two clusters share their *distinct token set*, which might indicate that there are ‘parallel taxonomies’ with possibly different underlying entities:

$$\delta(\mathcal{K}) = \delta(\mathcal{K}')$$

An example of easily-detectable parallel taxonomies[7] is in Fig. 3. More accurate verification if there is indeed a pair of parallel taxonomies (with name-wise correlated taxonomic paths rather than just common distinct token sets) could of course be done but would be computationally costly.

Pattern IV could possibly be used not only within one ontology but also *across* ontologies ($\mathcal{K}$ and $\mathcal{K}'$ being each from a different ontology), i.e. for *ontology matching*.

Again, as the token matching is considered positively, reflecting *thesaurus correspondence* would improve the *recall* of the pattern detection—in particular for the mentioned ontology matching case, where identical naming would be less likely than within a single ontology. In addition, from the practical point of view, it would make sense to *fuzzify* the pattern, as taxonomies could be seen as parallel even if one of them contains some concepts unmatched in the other.

Finally, parallel taxonomies could be matched by name tokens not only at the level of concepts but also at the level of *properties* and even at the level of *instances*, as

[7] From `http://www.daml.org/experiment/ontology/beta/military-elements-ont`.

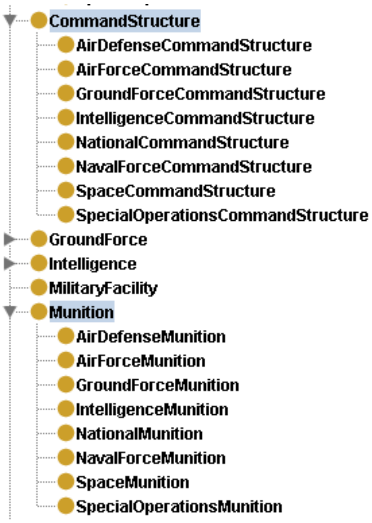

Fig. 3. Example of Pattern IV

subclassing and instantiation can be seen as mutually alternative modelling options in some situations [9]. This will probably lead to substantial refinement and extension of the pattern in the future.

5 Experiments for Pattern I

5.1 Method and Inputs Used

In the first, manual[8] phase of our experiments, focused solely on Pattern I,[9] we restricted the analysis to 3 small- to medium-sized ontologies we picked from public repositories. Their choice was more-or-less 'random', we however avoided ontologies that appear as mere (converted) ad hoc taxonomies without the assumption of set-theoretic semantics, as well as 'toy' models designed for demonstrating DL reasoning (such as 'pizzas' or 'mad cows'), which are actually quite common in such repositories, cf. [15]. In this paper we only show the detailed results for one of them, the ATO Mission Models ontology, as illustration, and also include a summary table. Results for the other two ontologies, Government and EuroCitizen, are in [13].

The *tokenisation* and *head noun detection* was carried out (straightforwardly, by eye) as described in Section 2.1. For *thesaurus correspondence*, we decided to use *WordNet*[10] as means to address synonymy and hyponymy, with the assumption that a general thesaurus is likely to contain the head nouns of multi-word domain terms.

[8] For examining the ontologies, we simply unfolded their taxonomies in Protégé.

[9] The experimental results have already been published in the workshop paper [13]; here we only include a part of them, with a substantially reworked interpretation.

[10] `http://wordnet.princeton.edu/`

However, we separately counted and listed the cases where the pattern compliance was established via WordNet only. We did not use WordNet for *single-token* child terms[11]; we rather excluded them from the analysis.

The results of the analysis amount to the simple statistics of:

1. Parent-child pairs for which Pattern I would be rejected outright due to identical head noun.
2. Parent-child pairs for which Pattern I would be rejected due to the head noun of the child being WordNet hyponym or synonym of the head noun of the parent.
3. Parent-child pairs where the correspondence between the head nouns cannot be established even via WordNet, but a human evaluator assessed the parent-child relationship as probably correct.
4. Parent-child pairs where the correspondence cannot be established even via WordNet, and a human evaluator assessed the parent-child relationship as probably incorrect (at least at the level of class names).

In the table below, the cases 2, 3 and 4 are explicitly listed and commented. Three symbolic labels were added for better overview:

- ⊗ means: head nouns correspond via thesaurus, i.e. Pattern I would not be detected if (WordNet) thesaurus were used.
- ○ means: correct relationship, thus 'unjustified' detection of Pattern I ('false alarm').
- ● means: incorrect relationship, thus 'justified' detection of Pattern I ('true alarm').

The number of cases 3 ('false positives') and 4 ('true positives') can be viewed as evaluation measures for our envisaged method of conceptual error detection. There could potentially be 'false positives' even among the cases 2 (and theoretically even among the cases 1) due to homonymy of tokens; we however did not clearly identify any such case. The *precision* of our approach can thus be simply established as the ratio of the number of cases 4 vs. the number of cases 3+4.

5.2 Results for ATO Mission Models Ontology

This, US-based military (ATO probably stands for 'Air Tasking Order') ontology, which we picked from the DAML repository,[12] is an ideal example of highly specific ontology rich in multi-token names; there are very few single-token ones, and none of these is involved as subclass in one of the subclass relationships. The ontology contains 86 classes (aside classes inherited from imported ontologies), and there are 116 parent-child relationships (including some multiple inheritance). Of them, 95 have an identical head noun, and 21 don't. Table 1 lists and comments the parent-child relationships that would be detected as Pattern I. We assume (see the table) that the majority of Pattern I occurrences (11, i.e. 52%) are modelling errors;[13] some other 'surface' Pattern I occurrences (5, i.e. 24%) should not be treated as such since the relationship between the tokens could be determined using WordNet, and only a few Pattern I occurrences (5, i.e. 24%) seem to be 'false alarms'.

[11] Our main focus are specialised domain ontologies, whose single-token terms are likely to either miss in standard lexical databases or exhibit a meaning shift there.

[12] `http://www.daml.org/ontologies/`

[13] Or, possibly, artifacts of the DAML→OWL conversion.

Superclass	**Subclass/es**	**Comment**
AirspaceControlMeasure	AirCorridor TimingReferencePoint DropZone CompositeAirOperationsRoute	● Subclassing indeed looks misleading. A ‘measure’ can be *setting up* e.g. a corridor, but not the corridor *itself*.
AirStation	AirTankerCellAirspace	● Probably a *part-of* relationship?
ATOMission	AircraftRepositioning	○ ‘Repositioning’ looks like acceptable term, though not hyponym of ‘mission’ in WordNet.
ATOMission	CompositeAirOperations	⊗ ‘Mission’ is direct hyponym of ‘operation’ in WordNet. Note the misuse of plural form.
ATOMissionPlan	IndividualLocationReconnaissanceRequestMission MissileWeaponAttackMission	● The ‘Plan’ token erroneously missing. The remaining 19 sibling subclasses do have it.
CommandAndControlProcess	AirborneElementsTheaterAirControlSystemMission	● Subclass clearly misplaced: ‘mission’ concept non contiguous.
CommandAndControlProcess	ForwardAirControl	● Probably means ForwardAirControlProcess.
CommandAndControlProcess	FlightFollowing	⊗ ‘Following’ could be seen as process (it is hyponym of ‘processing’ in WordNet).
ConstraintChecking	RouteValidation	○ Specialisation to subdomain; ‘validation’ should be related to ‘checking’ but, surprisingly, this is not the case in WordNet.
ControlAgency	ForwardAirControllerAirborne	○ A tricky case: the end token in subclass is actually an attribute of the head noun (‘controller’). Furthermore, although the relationship between ‘agency’ and ‘controller’ is not intuitive, it might be OK in domain context.
ForwardAirControl	AirborneBattleDirection	⊗ ‘Direction’ is direct subclass of ‘control’ in WordNet.
GroundTheaterAirControlSystem	ControlAndReportingCenter ControlAndReportingElement	○ Though the relationship between the end tokens is not intuitive, it looks OK in the domain context.
IntelligenceAcquisition	AirborneEarlyWarning	● Rather looks like two subsequent processes: warning is *preceded* by intelligence acquisition.
ModernMilitaryMissile	ArmyTacticalMissileSystem	● A system (i.e. group) of missiles, possibly including a launcher, is probably not a missile.
PrepositionedMaterielTask	GroundStationTankerMission	⊗ ‘Mission’ is close hyponym of ‘task’ in WordNet.
SupportingTask	GroundStationTankerMission	⊗ As above.

Table 1. Pattern I occurrences in the ATO Mission Models ontology

5.3 Summary of Results for Pattern I

Table 2 shows the overall figures. The results are most promising for the ATO Mission Models ontology, which is most domain-specific of the three. The *precision* of 'inconsistency alarms', if they were properly implemented, could be acceptable for human inspection and evaluation of the ontology. It could possibly be further improved by adding more thesauri in addition to WordNet, which would help eliminate some of the 'false alarms', cf. Table 1. However, perhaps with the exception of ATO Mission Models, the *coverage* of our simple approach is still too small to guarantee substantial 'cleaning' of taxonomic errors.

	ATO Missions	Government	EuroCitizen
Parent-child relationships	116	27	62
with multi-token subclass	116	24	40
Pattern rejected due to identical head noun	95	11	30
Pattern rejected due to WordNet correspondence	5	8	4
Pattern only rejected by human ('false alarm')	5	3	2
Pattern accepted by human ('true alarm')	11	2	4
Precision of 'alarm'	69%	40%	67%

Table 2. Summary of results for Pattern I

6 Experiments for Patterns II, III and IV

6.1 Data Acquisition, Pre-Processing and Mining

In order to acquire a high number of ontologies, we applied the *Watson* tool[14] via its API. The *Pellet reasoner*[15] was then called via the OWL API[16] in order to obtain a complete taxonomy. The *tokenisation* and *head noun detection* was carried out as described in Section 2.1 similarly as for Pattern I, but this time automatically. *Thesaurus correspondence* was not followed (its inclusion is ongoing work).

The actual algorithm for *pattern mining* within an ontology can be briefly described as follows:

1. All concepts are first grouped according to their *head noun* (for Patterns II and III) or to sharing *any token* (for Pattern IV).
2. The grouping is refined using information about the *taxonomic structure*, yielding *structural clusters* with non-empty token set.[17] This is done by following the parent-child links in each group in a top-down, depth-first manner. *Sibling concepts* (for Pattern II) and *descendant concepts* (for Pattern III) are thus straightforwardly discovered.

[14] `http://watson.kmi.open.ac.uk/editor_plugins.html`

[15] `http://pellet.owldl.com/`

[16] `http://owlapi.sourceforge.net/`

[17] For Patterns II and III, *named structural clusters* are thus detected.

3. For Pattern IV, the *distinct token set* is then computed for each cluster, which then straightforwardly serves as basis for pattern discovery across pairs of clusters.

All data were stored in a database, and the patterns were picked using SQL queries.

6.2 Patterns Statistics

Table 3 lists the statistics of pattern detection over the whole collection of 591 ontologies. Occurrences of Patterns II and III seem to be relatively evenly dispersed across the ontology collection. This is not the case of Pattern IV: there were ten ontologies such that each contained over 200 detected parallel taxonomies, while more than two thirds of ontologies did not contain any. We plan to elaborate on statistical distributions of patterns in the future, relating them to the statistics of more basic features typically observed in ontology metrics [16].

For comparison, we should also recall the Pattern I 'statistics' (on three ontologies): the average count is 27/3 = 9, or about 15 if we include the cases with thesaurus correspondence (analogously to the counts for Patterns II, III and IV). Obviously, all these numbers strongly depend on the size of ontologies. However, they might indicate whether applying the described approach on an ontology of interest will on average yield 'something interesting' at all.

Pattern No.	II	III	IV
Pattern Nickname	Matching Siblings with Non-Matching Parent	Matching Outlier	Parallel Structure Candidate
Total # of occurrences detected	2327	1368	7858
Average # of occurrences per ontology	3,94	2,31	13,30
Ontologies with at least one occurrence	58%	28%	32%

Table 3. Detected occurrences for Patterns II, III and IV over 591 ontologies

6.3 Patterns Evaluation

We only carried out preliminary evaluation of precision of pattern retrieval; as the process is rather demanding, only 17 of the ontologies were so far processed in this way. Six ontologies were just randomly picked from the large collection. The remaining eleven were ontologies from the *OntoFarm* collection,[18] most of which have been previously used for experiments with ontology matching tools within the OAEI challenge.[19] The rationale was to allow for subsequent ontology matching experiments (ongoing work), as the *OntoFarm* collection is one of few available collections of ontologies describing all the same domain (namely, conference organisation).

[18] `http://nb.vse.cz/~svatek/ontofarm.html`

[19] `http://nb.vse.cz/~svabo/oaei2007/`; also see [14]

An experienced knowledge engineer[20] inspected the discovered patterns in Protégé and judged their detection as either 'justified' or 'unjustified'. 'Justified' detections for Patterns II and III were conceived analogously to those for Pattern I, i.e. as potential errors (in naming or underlying conceptualisation). 'Justified' detections for Pattern IV simply corresponded to parallel taxonomies really being present (in contrast to mere coincidence in token naming). The evaluation was certainly rather coarse-grained and subjective: it is inherently more fragile to judge someone's conceptualisation based on concept names than to verify the logical consistency of a knowledge base.

The results are in Table 4. In fact, a significant part of the 'unjustified' detections were due to incorrect handling of exceptional cases such as multiple inheritance; the real accuracy for Patterns II and III is thus likely to be increased merely via algorithms debugging. Some mistakes are however due to inherently hard problems such as term synonymy or polysemy; a few also arose due to imperfect discovery of head noun.

	'Justified'	'Unjustified'	Total	Precision
Pattern II	48	11	59	81%
Pattern III	16	8	24	67%
Pattern IV	24	10	34	71%

Table 4. Precision of detection for Patterns II, III and IV over 17 ontologies

7 Discussion of Methodologies and Results

Two different methodologies of pattern discovery/evaluation from Sections 5 and 6 reflect both the evolution of the underlying model while addressing different hypotheses and the computational requirements of the evaluation. The choice of patterns II to IV was actually partly influenced by the ease of their discovery: as all of them include a certain number of concepts with overlapping token sets, they can be identified using simple token indexing methods.

The main differences between the two methodologies were the following:

- The evaluation of Pattern I was motivated by the task of *error detection*. Therefore, the main interest was in the accuracy of the detection itself. In contrast, when addressing the (richer) Patterns II, III and IV, we also started to explore the potential of *automated semantic interpretation* of the patterns structures.
- Consequently, the evaluation of Pattern I required careful analysis of the context of each detected occurrence to see if there is indeed a (probable) error in the ontology or if there seem to be good semantic reasons for the head noun change. In contrast, the evaluation of the remaining three patterns was more shallow, partly because they were dispersed in more ontologies (none of which could be manually examined so thoroughly). Moreover, the 'unjustified' detections were mainly artifacts of technological errors in the automated analysis itself.

[20] As we dealt with ontologies from diverse domains, it was not feasible to employ domain experts for each.

- Technically, only parent-child pairs where the child had a multi-word name were considered as eligible for Pattern I detection. Namely, we assumed that including single-word names would lead to an increase of false alarms, as such names typically correspond to specific technical term unlikely to appear in thesauri. We did not impose this constraint in subsequent experiments, since single-word terms are rare in specialised ontologies in general (there was e.g. none in the ATO Missions ontology) and even less likely to appear in Patterns II and III currently relying on 'positive' head noun identity (e.g. if one sibling is a single-word term, the other is very unlikely to contain this term as head noun) and on distinct token sets.
- As Patterns II, III and IV were searched for in much higher number of ontologies (591 in contrast to only 3 for Pattern I), the overall statistics of pattern occurrence (despite the automated, less reliable method of the detection) in Table 4 is more meaningful. Analysis of Pattern I is more interesting as a kind of casuistics, detailed in Table 1.
- Due to much higher number of ontologies for Patterns II, III and IV, automated tools (Watson search tool and Pellet reasoner) were used even in the phase of data acquisition and pre-processing.

In both cases, the experimental results fulfilled our expectations. Even if the precision and recall of naming-based methods used is certainly not high enough for fully-automated processing, they may at least bring additional contribution on the top of logic-based methods, and provide at least some hint in case the given ontology is poor in axioms. Although our experiments were carried out in isolation, we see the practical use of name pattern analysis in close connection with other ontology evaluation and analysis methods, which would supply complementary information.

8 Related Research

The question whether naming in knowledge representation languages matters or not has been subject of notorious dispute, see e.g. [8]. Given the presumption that ontologies are not machine-*only* models and are indeed frequently inspected by humans in various contexts, the attention paid to the analysis of concept naming within the graph structure has so far been surprisingly low.

Our research is to some degree similar to projects aiming at converting shallow models such as thesauri or directory headings to more structured and conceptually clean ontologies [3–6, 11]. The main difference lays in our assumption that the ontologies in question are already intended to bear set-theoretical semantics, and that the 'inconsistencies' in naming patterns are due to either sloppy naming (possibly just reflecting shortcut terminology used by domain practitioners) or more serious modelling errors, rather than being an inherent feature of (shallow) models.

On the other hand, the research in 'true' OWL ontology evaluation and refactoring has typically been focused on their logical aspects [1, 7, 16]. Our research is, in a way, parallel to theirs. We aim at similar long-term goals, such as detecting potential modelling inconsistencies or making implicit structures explicit. We however focus on a different aspect of ontologies: the naming policy. Due to the subtler nature of consistency or implicit structures in these realms (usually requiring some degree of acquaintance with the domain), the conclusions of name pattern analysis have to be more cautious than those resulting from logic-based analysis.

9 Conclusions and Future Work

Ontology patterns associating concept naming with graph structures is an underrated source of information about pre-existing OWL ontologies to be used in various applications, from ontology quality evaluation through ontology refactoring to ontology mapping. We outlined a descriptive model for representing such patterns, and carried out experiments for four such patterns with the aim of detecting potential errors in set-theoretic interpretation, awkward naming policy, as well as structures helpful for concept matching across ontologies. Two different strategies were used in the experiments. While for the simplest and most abundant 'non-matching child' pattern the analysis was done manually for a small set of ontologies, for the remaining three (which all involve at least one pair of matching tokens) we focused on automation allowing to process a large collection of ontologies.

There are many ways in which the current work is to be extended. The most imminent one is more extensive involvement of *prior lexical resources* such as thesauri, which we so far only used in the manual analysis for Pattern I. The accuracy of detection of the remaining four patterns could presumably be improved in this way; synonym/hyponym recognition could result in better precision and recall for Pattern II, and in better recall for Pattern III and IV.

So far, the focus of the research was in the combination of name pattern analysis with graph analysis. There is definitely much room left for improving the name pattern analysis itself, in the first phase just using *term extension* relationship (via adding adjectives or nouns in apposition to the given term) rather than just head noun matching. More advanced techniques of terminology analysis[21] could also be adopted. Such techniques are typically (at least partially) biased by a certain, though possibly broad, discipline; in order to make the proof of concept for our approach in its generic form, we so far avoided delving to this kind of techniques, but they would clearly be a further way for performance improvement.

Furthermore, concept names used as identifiers are obviously not the only lexical items available in ontologies. Future (especially, more automated) analysis should pay similar attention to additional, potentially even multi-lingual *lexical labels* (based on `rdf:label`) and *comments*, which may help reveal if the identifier name is just a shortcut of the 'real' underlying concept name. In addition to class names, *property* naming (in connection with their domain and range) should also be followed, e.g. as drafted in [12]; this will of course require extension of the current formal *descriptive model*. We already carried out initial experiments with the detection of *reified n-ary relations*, with promising results. Detection of concepts corresponding to n-ary relations would allow to associate concepts to relations in ontology mapping (the so-called heterogeneous mappings, see [2]).

[21] The state of the art is being presented, among other, at the TIA conferences (`http://www-sop.inria.fr/acacia/tia2007`)

Finally, one of our main objectives is to ease ontology *mapping*. We therefore plan to exploit the discovered patterns (especially for the OntoFarm collection of ontologies) in the matching process. One experimental direction is to measure the effect of name-pattern-based *ontology refactoring* on the *logical properties* of those ontologies after they have been mapped to each other, in a similar style as [7] measures the effect of added disjointness axioms. Another possibility is to directly consider the intra-ontology patterns discovered through naming analysis as building blocks (inputs/outputs) to inter-ontology *correspondence patterns* [10].

The research was partially supported by the IGA VSE grants no.20/07 "Combination and comparison of ontology mapping methods and systems" and no.20/08 "Pattern-based ontology evaluation and matching".

References

1. Baumeister J., Seipel D.: Smelly Owls – Design Anomalies in Ontologies. In: Proc. FLAIRS 2005, 215–220. Again, as the token matching is considered positively, reflecting thesaurus correspondence would improve the *recall* of the pattern detection.
2. Ghidini, C., Serafini, L.: Reconciling concepts and relations in heterogeneous ontologies. In: Proc. ESWC 2006, Budva, Montenegro, 2006.
3. Giunchiglia F., Marchese M., Zaihrayeu I.: Encoding Classifications into Lightweight Ontologies. In: Proc. ESWC 2006.
4. Hepp M., de Bruijn J.: GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In: Proc. ESWC 2007.
5. Kavalec M., Svátek V.: Information Extraction and Ontology Learning Guided by Web Directory. In: ECAI Workshop on NLP and ML for ontology engineering. Lyon 2002.
6. Magnini B., Serafini L., Speranza M.: Making Explicit the Hidden Semantics of Hierarchical Classifications. In: Proc. AI*IA 2003.
7. Meilicke, C., Völker, J., Stuckenschmidt, H.: Learning Disjointness for Debugging Mappings between Lightweight Ontologies. In this volume.
8. Nirenburg S., Wilks Y.: Whats in a symbol: Ontology and the surface of language. *Journal of Experimental and Theoretical AI*, 13:9-23.
9. Rector, A. (ed.): Representing Specified Values in OWL: "value partitions" and "value sets". W3C Working Group Note, 17 May 2005, online at `http://www.w3.org/TR/swbp-specified-values/`.
10. Scharffe, F., Euzenat, J., Ding, Y., Fensel, D.: Correspondence Patterns for Ontology Mediation. In: Workshop on Ontology Matching collocated with ISWC, Busan, Korea, 2007.
11. Serafini L., Zanobini S., Sceffer S., Bouquet P.: Matching Hierarchical Classifications with Attributes. In: Proc. ESWC 2006.
12. Svátek, V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: 7^{th} Conf. on Business Information Systems (BIS-04), Poznan, April 2004.
13. Svátek V., Šváb O.: Tracking Name Patterns in OWL Ontologies. In: EON-2007 at ISWC-2007, Busan, Korea.
14. Šváb O., Svátek V., Stuckenschmidt H. Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In: ESWC-2007. Innsbruck, Austria.
15. Tempich C., Volz R.: Towards a benchmark for Semantic Web reasoners - an analysis of the DAML ontology library. In: EON Workshop at ISWC 2003.
16. Vrandecic D., Sure Y.: How to Design Better Ontology Metrics. In: Proc. ESWC 2007.