Ontology Quality Problems - An Experience with Automatically Generated Ontologies

Mounira Harzallah, Giuseppe Berio, Toader Gherasim, Pascale Kuntz

Ontology Quality Problems
An Experience with Automatically Generated Ontologies

Mounira Harzallah[1], Giuseppe Berio[2], Toader Gherasim[1] and Pascale Kuntz[1]
[1]*LINA, University of Nantes, Rue Christian Pauc, Nantes, France*
[2]*IRISA, University of Bretagne Sud, Vannes, France*

Keywords: Ontology, Ontology Quality Problems, Quality Problems Detection, Ontology Quality Evaluation, Automatically Generated Ontology.

Abstract: Ontologies play a major role in the development of personalized and interoperable applications. However, validation of ontologies remains a critical open issue. Validation is fundamentally driven by an "ontology evaluation", often referred to as "quality evaluation", as better explained in the Introduction. This paper reports an experience designed on our previous work on quality evaluation and using ontologies automatically generated from some textual resources. In the previous work, we have proposed a standard typology of problems impacting (negatively) on the quality of one ontology (named *quality problems*). The experience shows how our previous work can be practically deployed. One a posteriori analysis of experience results and lessons learnt presented in the paper make explicit and concrete key contributions to validation. Finally, conclusions highlight both limitations of the experience and research perspectives.

1 INTRODUCTION

Ontologies are becoming widely recognised as components in various types of systems and applications: e.g. knowledge management, social network analysis, business intelligence. In particular, ontologies play a major role in the development of personalized applications, such as learning management applications (Lundqvist *et al.* 2011). Ontologies have been and still are manually designed by involving human experts. However, the ever-increasing access to textual sources (as technical documents, web pages and so on) has motivated the development of tools for automatizing as much as possible the design and implementation process. Partial automation is also quite useful for scaling changes in available textual resources and knowledge exploitation from the web (Sanchez, 2007; Mustapha *et al.* 2009) that require processes where human involvement is minimized. Promising results were reached (Cimiano & Volker, 2005; Cimiano *et al.* 2009). Unfortunately experimental studies underline have then raised limits for real-life applications (Cimiano *et al.* 2009; Hirst, 2009), and recent works recommend a better integration of human involvement (Simperl & Tempich, 2009).

Following this recommendation, we see the process of building an ontology as made of two main subprocesses running in parallel and cooperating:

- One generation process,
- One validation process.

The generation process focuses on the extraction of relevant items (such as terms or relations) and the identification and naming of relevant knowledge (such as concepts). The validation process is performed anytime when needed during the generation process. This is because, according to our experiences, validation should be performed as soon as possible focusing on subparts (such as subset of concepts) of the ontology under construction. Furthermore validation process can be defined as the process guaranteeing the expected quality of the ontology (while the generation process makes the ontology content available). Thus the validation process is a process firstly looking for poor quality, defects or problems in the ontology under construction (this first looking is sometimes referred to as "*quality evaluation* or *ontology evaluation*") and secondly proposing alternatives and applying selected alternatives for increasing quality, leaving out defects and problems in the ontology under construction. In this paper, we are going to focus on the first aspect of the validation process (i.e. quality evaluation), responsible for warning poor quality,

Harzallah M., Berio G., Gherasim T. and Kuntz P..
Ontology Quality Problems - An Experience with Automatically Generated Ontologies.
DOI: 10.5220/0005141301200129
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 120-129
ISBN: 978-989-758-049-9

defects, and problems in the ontology under construction.

Since the pioneering works of Gruber in the 90's (Gruber, 1993), "ontology evaluation" (or quality evaluation) has been discussed in (Gomez-Perez, 1995), and various procedures and features have been proposed (Duque-Ramos *et al.* 2011; Gomez-Perez, 2004; Hartmann *et al.* 2004, Baumeister & Seipel, 2005): e.g. quality evaluation by a set of measures, comparing ontologies to reference ontologies (also called gold standard), performing assessment of formal correctness, quality qualitative evaluation performed by experts, quality evaluation according to the results of a given application using the ontology, using pre-defined anti-patterns corresponding to lower, bad and poor quality (Roussey *et al.* 2009). Roughly speaking, quality evaluation also depends on three major criteria: (1) the dimensions which are evaluated (functional dimension, structural dimension or usability dimension) (Gangemi *et al.* 2006; Duque-Ramos et al., 2011), (2) the evaluation mode (manual vs. automatic) (Vrandecic, 2009) and the user profile if involved (knowledge engineers, business analysts, practitioners, etc.) (Hartmann *et al.* 2004), (3) and the phase in which evaluation is conducted (e.g. during the ontology development, just before ontology publication and so on) (Hartmann *et al.* 2004; Tartir *et al.* 2010).

As recognised in "Handbook on ontologies" (2009) by Vrandecic, proposed approaches mainly focus on how to recognise in the whole ontology or in its parts, potential or occurring, problems or defects, which correspond to lower, bad, or poor ontology quality. More precisely, we consider that there are two main *facets* of quality evaluation: scoring "quality" by introducing explicit measurements, and identifying problems (or defects) impacting ontology quality. The two facets are naturally related: for instance, an ontology (the problem being the defect the axioms causing the problem) can be inconsistent because of one axiom, two axioms and so on leading to a differing quality scores. However, measurements are often not directly correlated, neither theoretically nor empirically, to problems. This situation leads to difficulties to use measurements in practice. On the contrary, problems are often closer to defects than measurements; thus, using problems and their dependencies for removing defects seems more effective than using measurements. Therefore, in our previous work (Gherasim *et al.* 2013) we have decided to focus on the problem facet. This previous work has been specifically targeted one critical aspect of the problem facet i.e. the standardisation of *problems definitions*, currently rather variable and specific. Accordingly, we have proposed in (Gherasim *et al.* 2013) a *typology of problems* which: makes a synthesis of the state of the art, is extensible, is easy to understand and founded a well-known quality framework defining quality for conceptual models (ontologies are special cases of conceptual models). However, in the past work, we do not provide details on how, in practice, the proposed typology can be deployed in the context of the validation process. In this paper, we are going to present one experience (based on 2 ontologies automatically built from textual sources) showing how the proposed typology can concretely be deployed and used in the context of the validation process.

The paper is organised as follow. Section 2 provides a short overview on quality and problems in ontologies. Section 3 introduces the proposed typology. Section 4 describes the performed experience and provides feedbacks reporting (discussion and lesson learnt). Finally, perspectives and conclusion end up the paper.

2 STATE OF THE ART INSIGHTS

As reported in the Introduction, ontology quality evaluation concerns two related facets: ontology quality measurements (i.e. specific *techniques* for how to provide quality score associated to ontologies) and quality problems identified by using some *techniques* (i.e. techniques for how to highlight occurring or potential problems that may lead to partial or full employability of ontologies). Implicitly, quality problems should lead to lower quality measurements and lower quality measurements should warn on potential or occurring problems: therefore, a relationship between problems and measures needs to be put in evidence Figure 1 below provides a simple picture (as a UML class diagram), for representing the facets (i.e quality problem and quality measure) and key relationships. For instance, a well-known quality measure is "ontology depth": "ontology depth" can be used to point a (potential or occurring) problem such as "ontology flatness". However, quality measures do not need to be related to problems ("0 cardinality" in the figure) and the same for problems. Finally, even if a problem may not be associated to some techniques for identifying it, this is an uncomfortable situation in practice.

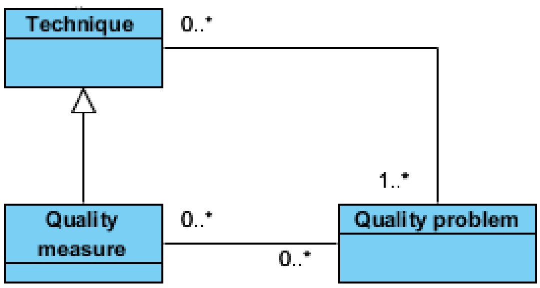

Figure 1: Facets of quality problems and key relationships.

Sections 2.1 and 2.2 below shortly present relevant insights on state of the art approaches.

2.1 Quality Measurements

As reported in the Introduction, existing proposals cover various "quality dimensions". In the context of ontologies, dimensions have not been standardised. For instance, they may be referred to as, syntax, semantics, maintenance and ergonomics or, functional, structural and usability.

One of the most complete proposals associating dimensions and measures is probably oQual (Gangemi *et al.* 2006). In oQual, an ontology is analysed according to three dimensions: i) structural (syntax and formal semantics of ontologies represented as graphs), ii) functional (intended meaning of the ontology and its components) and iii) usability (pragmatics associated with annotations, which contribute to the ontology profiling). A set of measures is associated with each dimension to score the quality. For instance: for structural dimension, depth and breadth of a taxonomy; for functional dimension, precision and recall of the ontology content with respect to its intended meaning; for usability dimension, number of annotations.

Despite the potential interest of proposed measurements, some of them remain quite disconnected to concrete or potential problems and do not account for problems dependencies. For instance, the ratio between number of concepts and number of relations (*N°Concepts/N°Relations*) is a quality measure for evaluating the "cognitive ergonomics", which is closely related to the "easy to use": "easy to use" does not correspond to a concrete problem but a potential "broad quite generic requirement".

Therefore, according to Figure 1 above, having as entry point quality measures (possibly organised alongside several dimensions) does not necessarily make explicit related and dependent problems. Vice-versa, defining problems as entry point provides an evidence of lower, bad and poor quality.

2.2 Quality Problems

Roughly speaking, the generic notion of "ontological error" covers a wide variety of problems. These problems affect different dimensions covered by ontology quality measurements. In the relevant literature, it is possible to find precise and less precise definitions for several recognised problems: (1) "taxonomic errors" (Gomez-Perez *et al.* 2001; Gomez-Perez, 2004; Fahad & Qadir, 2008; Baumeister & Seipel, 2005) or "structural errors" (Buhmann *et al.* 2011), (2) "design anomalies" or "deficiencies" (Baumeister & Seipel, 2005, 2010), (3) "anti-patterns" (Corcho *et al.* 2009; Roussey *et al.* 2009; Poveda *et al.* 2009; Buhmann *et al.* 2011), (4) "pitfalls" or "worst practices (Poveda *et al.* 2009, 2010) and (5) "logical defects" (Buhmann *et al.* 2011). Additional errors could complete this list: e.g. (6) "syntactic errors" (Buhmann *et al.* 2011).

Hereinafter, we shortly present insights on each of the above mentioned problem cases. Syntactic errors are due to violations of conventions of the language in which the ontology is represented. While interesting in practice for building support tools, syntactic errors (6) are conceptually less important than others: therefore they will be no longer considered in the remainder.

Taxonomic errors (1) concern the taxonomic structure and are referred to as: inconsistency, incompleteness and redundancy (Gomez-Perez *et al.* 2001). Three classes of "inconsistency" both logical and semantic have been highlighted: circularity errors (e.g. a concept that is a specialization or a generalization of itself), partitioning errors (e.g. a concept defined as a specialization of two disjoint concepts), and semantic errors (e.g. a taxonomic relationship in contradiction with the user knowledge). Incompleteness occurs when for instance, concepts, relationships or axioms are missing. Finally, redundancy occurs when for instance, a taxonomical relationship can be deduced from the others by logical inference.

Design anomalies (2) concern ontology understanding and maintainability (Baumeister & Seipel, 2005, 2010): lazy concepts (leaf concepts without any instance or not considered in any relation or axiom), chains of inheritance (long chains composed of concepts with a unique child), lonely disjoint concepts (superfluous disjunction axioms between distant concepts, which can disrupt inference reasoning), over-specific property range and property clumps (duplication of the same properties for a large concept set which can be retrieved by inheritance).

Anti-patterns (3) are known or recognised templates potentially leading to identified problems (Roussey *et al.* 2009; Buhmann *et al.* 2011). Some classes of anti-patterns are: logical anti-patterns (producing conflicts that can be detected by logical reasoning), cognitive anti-patterns (caused by a misunderstanding of the logical consequences of the axioms), and guidelines (complex expressions true from a logical and a cognitive point of view but for which simpler or more accurate alternatives exist).

Pitfalls (4) cover problems for which ontology design patterns (ODPs) are not available. An ODP cover ad-hoc solutions for the conception of recurrent particular cases (Corcho *et al.* 2009). (Poveda *et al.* 2010) have proposed 24 kinds of pitfalls grouped on 7 classes, them-self classified under the three ontology dimensions cited above (Gangemi *et al.* 2006). Four pitfalls classes are associated with the structural dimension: modelling decisions (false uses of OWL primitives), wrong inference (false reasoning induced by relations or axioms), no inference (lacks in the ontology which prevent inferences required to produce new desirable knowledge), real-world modelling (common sense knowledge missing). One class is associated with the functional dimension: requirement completeness (e.g. uncovered specifications). And, two classes are associated with the usability dimension: ontology understanding (information that makes understandability more difficult e.g. concept label polysemy or label synonymy for distinct concepts) and ontology clarity (e.g. variations of writing-rule and typography for the labels). Poveda *et al.* (2010) have also tried to classify the 24 pitfalls according to the three previous taxonomic error classes (Gomez-Perez *et al.* 2001); but pitfalls which concern the ontology context do not fit with this classification.

3 PROBLEM STANDARDISATION OVERVIEW

Mentioned in the Introduction and made evident in Section 2.2, heterogeneity in quality problems and their definitions is due to distinct experiences, communities and perception of ontologies. Standardisation enables a much better understanding of what problems are and to what extent these problems are critical before using the ontology. We have therefore proposed a two-level rigorous problem typology summarised in Table 1. Level 1distinguishes logical from social ground problems and level 2 distinguishes errors from unsuitable situations. *Errors* are problems (mostly) preventing the usage of an ontology. We add "mostly" because in the case of "inconsistency error" (Table 1), some researches focus on how to make usable inconsistent ontologies. On the contrary*, unsuitable situations* are problems which do not prevent its usage (within specific targeted domain and applications). Therefore, while errors need to be solved, unsuitable situations may be maintained in the ontology.

At level 1, "*social ground problems*" are related to the interpretation and the targeted usage of the ontologies by social actors (humans or applications based on social artefacts); whereas, *logical errors* and most of *logical unsuitable* cases can be rigorously formalized within a logical framework. For instance, they can be formally defined by considering key notions synthesised by Guarino *et al.* (2009) i.e.: Interpretation (I) (extensional first order structure), Intended Model, Language (L), Ontology (O) and the two usual relations $\models$ and $\vdash$ provided in any logical language. The relation $\models$ is used to express both that one interpretation I is a model of a logical theory L, written as $I \models L$ (i.e. all the formulas in L are true in I: for each formula $\varphi \in L$, $I \models \varphi$), and also for expressing the logical consequence (i.e. that any model of a logical theory L is also a model of a formula: $L \models \varphi$). The relation $\vdash$ is used to express the logical calculus i.e. the set of rules used to prove a theorem (i.e. any formula) φ starting from a theory L: $L \vdash \varphi$). Accordingly, when needed, problems are formalised by using classical description logic syntax that can also be transformed in FOL or other logics.

Problems in Table 1 are not necessarily independent (namely *problem dependency*) according to the following definition: existence of one problem in one ontology may reveal existence of another one; for instance, if an ontology is incomplete (L3), existence social incompleteness (S4) may also be revealed. However, absence of one problem may not reveal absence of another one: suppose that absence of social incompleteness (S4) has been assessed by using some techniques, which are by definition specifically focusing on the user-viewpoint; you may not conclude absence of L3 because L3 can be detected by using intended models (which need to be known) and formally checked by using some logical mechanisms. Problems dependencies made the framework even more effective for finding as many as possible defects in the ontology.

Table 1: The typology of quality problems.

Logical ground problems	
Errors	L1. Logical inconsistency: no I of s.t. I $\models$ O
	L2. Unadapted ontologies: there is a formula φ for some intended models of L, φ is false and O $\models$ φ
	L3. Incomplete ontologies: there is a formula φ for each intended models of L, φ true and O φ
	L4; Incorrect (or unsound) reasoning: when a false formula φ in the intended models O $\not\models$ φ, can be derived from a suitable reasoning system (O $\vdash$ φ)
	L5. Incomplete reasoning: when a true formula φ in the intended models O $\models$ φ, cannot be derived from a reasoning system (O $\nvdash$ φ)
Unsuitable cases	L6. Logical equivalence of distinct artefacts: O $\models A_i = A_j$
	L7. Logical indistinguishable artefacts: impossible to prove any of the following statements: (O $\models A_i = A_j$), (O $\models A_i \cap A_j \subseteq \bot$) and (O $\models c \subseteq A_i$ and $c \subseteq A_j$)
	L8. OR artefacts: A_i equivalent to $A_j \cup A_k$, $A_i \neq A_j$, $A_i \neq A_k$, but for which (if applicable) there is neither role R s.t O $\models (A_j \cup A_k) \subseteq \exists R.\top$, nor instance c s.t. O $\models c \subseteq A_j$ and O $\models c \subseteq A_k$
	L9. AND artefacts: A_i equivalent to $A_j \cap A_k$, $A_i \neq A_j$, $A_i \neq A_k$, but for which (if applicable) there is no common (non optional) role/ property for A_j and A_k
	L10. Unsatisfiability: given an artefact A, O $\models A \subseteq \bot$)
	L11. Complex reasoning: unnecessary complex reasoning when a simpler one exists
	L12. Ontology not minimal: unnecessary information
Social ground problems	
Errors	S1. Social contradiction: contradiction between the interpretation and the ontology axioms and consequences
	S2. Perception of design errors: e.g. modelling instances as concepts
	S3. Socially meaningless: impossible interpretation
	S4. Social incompleteness: lack of artefacts
Unsuitable cases	S5. Lack of/poor textual explanations: lack of annotations
	S6. Potentially equivalent artefacts: similar artefacts identified as different
	S7. Socially indistinguishable artefacts: difficult to distinguish different artefacts
	S8. Artefacts with polysemic labels
	S9. Flatness of the ontology: unstructured set of artefacts
	S10. Non-standard formalization of the ontology: unreleased specific logical use
	S11. Lack of adapted and certified version of the ontology in various languages
	S12. Socially useless artefacts

4 EXPERIENCE

This section presents an experience based on two ontologies automatically generated from different corpora by using Text2Onto (Cimiano & Volker, 2005). We have used Text2Onto in one of our past research projects (Harzallah, 2012) and realised a full comparison with similar tools. The comparison results made possible to select Text2Onto as the best choice for realising the work. However, Text2Onto capability for extracting concepts and taxonomical IS-A relationships has been shown to significantly outperform its capability for extracting other artefacts (Volker & Sure, 2006; Gherasim *et al.* 2012). This has also been confirmed by the performed work in the project: indeed, it was be possible to use, at least as a good base, extracted ontologies as components for realising interoperations between enterprise systems.

4.1 Experience Setting

As said above, we have generated two ontologies by using Text2Onto. Generated raw ontology O1 (resp.

O2) contains 441 (resp. 965) concepts and 362 (resp. 408) taxonomic relationships. The first ontology (O1) has been generated starting from a scientific article in the domain of "ontology learning from texts"; the article contains 4500 words. The second ontology (O2) has been generated starting from a technical glossary composed of 376 definitions covering the most important terms used in the domain of composite materials. The glossary contains 9500 words. The glossary has been provided by enterprises (working in the composite material sector) involved in the (omitted) project.

It should be noted that although showing quite different content features, the size of the two selected textual resources has been deliberately limited to enable further detailed analysis of the experience results.

4.2 Typology Deployment for the Experience

Problems can only be detected by applying appropriate techniques. Without aiming to be exhaustive (which is not an objective of this paper), there are several available techniques that either can be newly used for finding some of the listed problems or have been already used (for instance, reasoning techniques are reused for checking inconsistency, incomplete and unadapt ontologies).

There is no need to possess a technique for each typology problem to identify problems within an ontology, especially because ontologies range from very simple (or light) to very complex (or heavy), respectively represented with simple graphs to first order and even higher order logics.

In the experience, ontologies generated with Text2Onto are very simple and basically represented as list of concepts related by IS-A relationships (i.e. concepts organised as a taxonomy resulting in what is sometimes referred to a *lightweight ontology*). Therefore, deploying the typology for this concrete experience does not need to associate techniques for identifying logic ground problems L1 to L5. Indeed, L1 cannot occur because L1 may only occur iff the ontology comprises axioms – other than the taxonomy. L2 to L5 are not applicable because they can be applied only iff intended models are known in some way.

Additionally, L10 (unsatisfiablity) is trivially non occurring for the same reason as above i.e. the very simple type of ontology without instances. The contrary happens for some social ground problems – for instance S5 (Lack or poor textual explanation) is trivially occurring because the tool does not provide any annotation.

For remaining logical ground problems, a first technique is required to transform the original Text2Onto raw outcome in OWL. However, the OWL version is not necessarily certified and S11 trivially occurs. Then, using the OWL version and a reasoning service (Pellet), it has been possible to identify:

- L6 (Logical equivalence of distinct artefacts); (e.g. area= domain = issue = end= section=object, path=shape);
- L12 (Ontology non minimal) because some of the IS-A relationships (original), transformed in OWL subsumptions, can be inferred from other ones.

Concerning L8 and L9 problems, the reasoner has not been able to find any concept equivalent to union or intersection of other concepts. So that, L8 and L9 do not occur. Because of the special form of the ontology comprising only concepts and IS-A relationships, checking L7 has been made possible by counting the pairs of non logically equivalent concepts (checked with L6). Section 4.3 provides additional details on L7 and the adopted technique.

Apart for S5 and S11 problems mentioned above, for the other social ground problems, we have been obliged to identify or develop our own techniques. However, because of the work scope, and because of most of these problems can only be identified only if stakeholders are directly involved (such as end-users, experts and so on), employed techniques do not guarantee unbiased results.

Therefore, by *formal inspection*, we have identified S1 (Social contradiction) by especially inspecting IS-A relationships and pointing the ones contradicting our own IS-A relationships. S2 (Perception of design errors) has been checked by focusing on the ambiguity/vagueness of the dichotomy concept vs. instance.

S3 (meaningless artefacts) has been quite evident with concepts labelled with artificial labels (e.g. a label such as “tx12”).

S4 (social incompleteness) has been detected as follow: whenever a concept is connected only to the root (so that it has no other relationship with other concepts because ontology is lightweight), the ontology is considered to be incomplete because probably lacking of additional IS-A relationships.

Useless artefacts (S12) are as such if it is impossible to provide simple and clear raison for including artefacts in the ontology (for instance, ‘train’, ‘cannot’ were trivially out of the ontology domain scope).

S6 (Potentially equivalent artefacts) has been identified as a problem occurring when labels for concepts are synonyms according to our knowledge of the domain (e.g. area=field, human = person, sheet = plane) or according to known domain references.

S7 (socially indistinguishable artefacts) has been highlighted whenever it was impossible for pair of concepts to both provide factual raison to made them equivalent and factual raison to made them distinct.

S8 (polysemy in artefact labels) has been identified by looking to the existence of several definitions, within the given domain, for the single concept label (e.g. labels such as cycle, repair).

Finally, S9 has been simply detected by calculating the average depth of the ontology as the average counting each taxonomy leaf depth, and comparing it to a (domain-or-application specific or generic-domain-independent) expected typical depth.

Table 2 below summarizes the problems detected, by using deployed techniques, in the two generated ontologies. Next section provides a discussion on experience feedback, mostly based on Table 2.

4.3 Discussion

During the experience, we have remarked the interest, when applicable, of keeping in mind "numbers of occurrences" of a given problem (for instance, S1 can be considered occurring several times as many as ontology artefacts suffer of the problem). Indeed, occurrences are a simple way to highlight differences in the two ontologies, then to identify causes of problems (hopefully defects) and potential correlations between problems. However, not all the problems can be counted: for instance, flatness problem (S9) cannot be counted.

Occurrences figures may not substitute problem dependencies (which are technique-independent and focusing only of existence/absence of problems according to their definitions, see Section 3). Let's consider occurrences of redundant taxonomical relationships (O1(L12:32), O2(L12:49)). These occurrences do not precisely represent semantic redundant relations, because it is possible that some of those relationships contradict user expectations (S1) and need to be removed. But L12 is detected independently from S1 as in the following simple case: Text2Onto generated in our experience *"result" is_a "issue", "issue" is_a "relation" and "result" is_a "relation"*. Despite the evident logical redundancy (easily identified by a reasoner), some of those IS_A relationships above have been considered suffering of S1 problem; so that removing some of them results in breaking independent redundancy detection. More generically, while deployed techniques find out problems independently (so that numbers of occurrences is not meaningful), problems themselves may be dependent (in the case above, L12 depends to S1).

The six most occurring problems are the same for O1 and O2, three are social and three are logical problems: S1, S4, S12, L6, L7 and L12. These problems have been checked involving both concepts and relationships. Occurrences of these problems are quite different in O1 and O2: S1 (O1: 130, O2: 45), S12 (O1: 121, O2: 31), L6(O1: 300, O2: 65), These differences may be quite surprising because numbers of concepts and relations in O1 are lower than in O2. We have therefore tried to provide alternatives non-exclusive explanations. Two explanations have been pointed out.

One alternative explanation concerns the nature of the content of the incoming textual resource. A technical glossary (starting point for O2) naturally providing definitions of terms, is more self-contained and more focused than a scientific paper (starting point for O1). Indeed, few concept labels in O2 can be considered very generic thus loosely related to the domain while this is not the case for O1. This seems to be confirmed by the fact that S12 (useless artefacts) occurs very often in O1 if compared to O2.

A second non-exclusive alternative explanation is traced back to the usage of Wordnet made by Text2Onto. Indeed, generic and rather useless concepts belonging to O1 enable Text2Onto to also introduce IS-A relationships belonging to Wordnet; these IS-A relationships are due to the several meanings associated by Wordnet to terms (for instance, for term "type" in case of O1, Text2Onto extracted: "type" is_a "case"; "type" is_a "group" and "type" is_a "kind"; each IS_A relation concerns one quite specific and distinct meaning of the term "type"). This is confirmed by much higher occurrences of S1 in O1 than in O2 (remember that S1 has been detected by focusing on IS-A relationships only, see Section 4.2).

Occurrences can also be fruitful for establishing potential correlations between problems. We have put in evidence three potential correlations.

Correlation 1: S12 is correlated with S1 (confirmed by results reported in Table 2: O1(S12: 121, S1: 130) and O2 (S12: 31, S1:45)). This is because, as also explained above, in our experience useless concepts are often source of incorrect taxonomic relationships.

Correlation 2: S9 (ontology flatness) shows similar values for the two ontologies (O1: 2:02 and O2: 1:99). This problem seems to be correlated to S4: if S4 occurs often (when compared to number of concepts), S9 is likely occurring too. This is because S4 is mostly checked by counting the concepts only related to the root. If S4 occurs often, in any case, the average depth tends to depend on number of concepts only related to the root.

Correlation 3: S12 is correlated with L6 is confirmed of results reported in Table 2: O1(S12: 121, L6: 276) and O2 (S12: 31, L6: 57)) because useless artefacts may generate additional logical equivalences (as in this experience).

4.4 Lessons Learnt

From the discussion above, two main lessons can be reported:

- Explanations for quality problems can be traced back to the content features of the incoming textual resources;
- Correlations and dependencies between problems suggest introducing an order for performing more efficiently the overall problem

Table 2: Identified quality problems in O1 and O2 with detailed occurrences figures and computed values.

Types of problems	Detected problems	
	Ontology O1 (441 concepts and 362 is-a relationships)	Ontology O2 (965 concepts and 408 relationships)
L1	Trivially non occurring	Trivially non occurring
L2	NA	NA
L3	NA	NA
L4	NA	NA
L5	NA	NA
L6	276 (=24*23/2, because we found 24 equivalent concepts) pairs of equivalent concepts (detected on the OWL version)	57 pairs of equivalent concepts (detected on the OWL version)
L7	Trivially occurring ;all pairs of concepts that are not equivalents are indistinguishable ((441*440/2)-276 indistinguishable pairs)	Trivially occurring; all pairs of concepts that are not equivalents are indistinguishable ((965 *964/2)-57 indistinguishable pairs)
L8	No "OR artefact"	No "OR artefact"
L9	No "AND artefact”	" No "AND artefact"
L10	Trivially non occurring	Trivially non occurring
L11	The ontology does not contain any situation that can make inferences more complicated	The ontology does not contain any situation that can make inferences more complicated
L12	32 redundant taxonomic relations	49 redundant taxonomic relation
S1	130 taxonomic relations contradict the evaluator’s knowledge	60 taxonomic relations contradict the evaluator’s knowledge
S2	2 instances were identified as concepts according to evaluator’s knowledge	5 instances were identified as concepts according to evaluator’s knowledge
S3	13 concepts with meaningless labels according to evaluator’s knowledge	21 concepts with meaningless labels according to evaluator’s knowledge
S4	168 concepts only connected to root	360 concepts only connected to root
S5	Trivially occurring (not counted)	Trivially occurring (not counted)
S6	9 pairs of concepts with synonymous labels	3 pairs of concepts with synonymous labels
S7	No couple of socially indistinguishable artefacts	No couple of socially indistinguishable artefacts
S8	7 concepts with polysemic labels	9 concepts with polysemic labels
S9	Flat ontology, affected by a lack of structuration (average depth of leaves = 2.02) Expected depth = at least 5	Flat ontology, affected by a lack of structuration (average depth of leaves = 1.99) Expected depth = at least 7
S10	No: a OWL version is available	No: a OWL version is available
S11	The ontology is not certified	The ontology is not certified
S12	121 useless concepts according to evaluator’s knowledge	31 useless concepts according to evaluator’s knowledge

identification task.

The first lesson highlights that before starting ontology building (or automated extraction), contents of the incoming textual resources should be evaluated and possibly improved. According to the second lesson, a potential order for running problem identification techniques (during the validation process) can be as follow:

1) Redundant taxonomies (L12) should be inspected to verify if they suffer of S1 (*Dependency suggestion*);

2) Useless artefacts (S12) should be identified (impacting on S1) *(Correlation suggestion)*.

3) Equivalent concepts (L6) should be identified and inspected to verify if they suffer of false taxonomical relations (S1) *(Correlation suggestion)*.

To establish the order, we have considered that dependencies are stronger than correlations: therefore, dependencies are used earlier in the identification. Identification should also run tightly integrated with the rest of the validation process. Specifically, at the end of each of steps (1, 2, 3), appropriate solutions (modification and deletion of involved artefacts, or additional artefacts) to identified problems should be selected and applied.

5 CONCLUSIONS

Through the paper, we have reported a typology of problems impacting the quality of an ontology and we have presented how in practice the typology can be used. Discussion (4.3) and Lessons Learnt (4.4) provide respectively emerging aspects of the proposed typology and suggestions for integrating deployed typology within a validation process.

Of course, performed experience does not cover various important points listed below:

- Because ontologies used in the experience are lightweight, typology deployment has only concerned a subset of problems; important problems, especially logical ground errors, are not covered by the deployment; however, specific techniques have been developed for trying to detect most of the logical ground errors; these techniques focus on algorithms for efficient reasoning and supporting users for expressing expected facts and transforming them in logical formula, justification and revision mechanisms can also be mentioned; however, some works (through SPARQL queries (Baumeister & Seipel, 2005), anti-patterns (Roussey *et al.* 2010; Corcho *et al.* 2009), heuristics (Pammer, 2010), tools OOPS (Poveda *et al.* 2012), MoKi (Pammer, 2010) have undertaken more empirical ways for detecting problems (therefore more focusing on potential problems than on occurring problems);
- Deployed techniques for social ground problems are quite simple; several works have investigated techniques that can be associated to social ground problems; for instance, reusable patterns and heuristics can be found in (Baumeister & Seipel, 2005; Poveda *et al.* 2012; Roussey *et al.* 2009; Burton-Jones *et al.* 2005); however, some techniques for social ground problems are not clearly confined because problems themselves while well-defined cover a quite large spectre of situations; other social ground problems are even recognised as open issues (such as S10 (Kalfoglou, 2010)).
- Lessons learnt are quite interesting and open three main research lines; i) investigating on text improvement for ontology building, ii) investigating on correlations and dependencies in more systematic way and, finally iii) investigating on integrating problem identification sub-process in the overall validation process.

REFERENCES

Baumeister, J., Seipel, D., 2005. Smelly owls-design anomalies in ontologies. Pages 215–220 of: *Proc. of the 18th int. florida artificial intelligence research society conf.*

Baumeister, J., Seipel, D., 2010. Anomalies in ontologies with rules. Web semantics: Science, services and agents on the world wide web, 8(1), 55–68.

Buhmann, L., Danielczyk, S., Lehmann, J., 2011. D3.4.1 report on relevant automatically detectable modelling errors and problems. Tech. rept. LOD2 - Creating Knowledge out of Interlinked Data.

Burton-Jones, A., Storey, V., Sugumaran, V., 2005. A semiotic metrics suite for assessing the quality of ontologies. Data and knowledge engineering, 55(1),84–102.

Cimiano, P., Volker, J., 2005. Text2onto – a framework for ontology learning and data-driven change discovery. Pages 227–238 of: Montoyo, A., Munoz, R., & Metais, E. (eds), *Proc. of the 2nd eur. semantic web conf.*, vol. 3513.

Cimiano, P., Maedche, A., Staab, S., Volker, J., 2009. Ontology learning. Pages 245–267 of: Studer, R., & Staab, S. (eds), Handbook on ontologies,. International Handbooks on Information Systems. Springer.

Corcho, O., Roussey, C., Blazquez, L.M. Vilches., 2009. Catalogue of anti-patterns for formal ontology debugging. Pages 2–12 of: Atelier construction d'ontologies: vers un guide des bonnes pratiques, afia.

Duque-Ramos, A., Fernandez-Breis, J.T., Aussenac-Gilles, N., Stevens, R., 2011. Oquare: Asquare based approach for evaluating the quality of ontologies. *Journal of research and practice in information technology*, 43, 159–173.

Fahad, M., Qadir, M.A., 2008. A framework for ontology evaluation. Pages 149–158 of: *Proc. Of the 16th int. conf. on conceptual structures (iccs'2008)*, vol. 354.

Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J., 2006. Modelling ontology evaluation and validation. Pages 140–154 of: *Proc. of the 3rd. semantic web conf. (eswc2006)*. LNCS, no. 4011.

Gherasim,T., Harzallah, M., Berio, G. Kuntz, P., 2013. Methods and tools for automatic construction of ontologies from textual resources: A framework for comparison and its application. Pages 177–201 of: Advances in knowledge discovery and management, vol. 471. Springer.

Gherasim,T., Harzallah, M., Berio, G. Kuntz, P., 2012. Problems impacting the quality of automatically built ontologies. *In Proceedings of the 8th Workshop on Knowledge Engineering and Software Engineering (KESE-2012)*, held in conjunction with ECAI 2012, p. 25–32, 2012.

Gomez-Perez, A., 1995. Some ideas and examples to evaluate ontologies. Pages 299–305 of: *Proc. of the 11th conf. on artificial intelligence for applications*.

Gomez-Perez, A., 2004. Ontology evaluation. Pages 251–274 of: Staab, S., & Studer, R. (eds), Handbook on ontologies, 1 edn. Int. Handbooks on Inf. Syst. Springer.

Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O., 2001. Ontological engineering: With examples from the areas of knowledge management, e-commerce and the semantic web. Adv. Inf. And Know. Processing. Springer.

Gruber, T.R., 1993. Atranslation approach to portable ontology specifications. Knowledge acquisition, 5(2), 199–220.

Guarino, N., Oberle, D., & Staab, S., 2009. What is an ontology? Pages 1–17 of: Studer, R., & Staab, S. (eds), Handbook on ontologies, *2 edn. International Handbooks on Information Systems*. Springer.

Hartmann, J., Spyns, P., Giboin, A., Maynard, D., Cuel, R., Suarez-Figueroa, M. C., Sure, Y., 2004. Methods for ontology evaluation. Tech. rept. Knowledge Web Deliverable D1.2.3.

Harzallah, M., 2012. Développement des ontologies pour l'interopérabilité des systèmes hétérogènes, applications aux cas industriels du projet ISTA3. Livrable final de la tâche 2.4 du projet ISTA3. http://interop-vlab.eu/workspaces/ISTA 3/.

Hirst, G., 2009. Ontology and the lexicon. Pages 269–292 of: Studer, R., & Staab, S. (eds), Handbook on ontologies, *2 edn. International Handbooks on Information Systems*. Springer.

Kalfoglou, Y., 2010. Cases on semantic interoperability for information systems integration: Practices and applications. IGI Global.

Lundqvist, K. O., Baker, K., Williams, S., 2011. Ontology supported competency system. Int. j. of knowledge and learning, vol.7, no.3/4, 197 – 219.. Pages 217–231 of: *Proc. of 2nd int. conf. on web and information technologies (icwit2009)*.

Pammer, V. 2010. Automatic support for ontology evaluation - review of entailed statements and assertional effects for owl ontologies. Ph.D. thesis, Graz University of Technology.

Poveda, M., Suarez-Figueroa, M.C., Gomez-Perez, A., 2009. Common pitfalls in ontologydevelopment. Pages 91–100 of: *Proc. of the current topics in artificial intelligence (caepia'09)*, and 13th conf. on spanish association for artificial intelligence.

Poveda, M., Suarez-Figueroa, M.C., Gomez-Perez, A., 2010. A double classification of commonpitfalls in ontologies. Pages 1–12 of: *Proc. of the workshop on ontology quality at ekaw 2010*.

Poveda, M., Suarez-Figueroa, M.C., Gomez-Perez, A., 2012. Validating ontologies with oops! Pages 267–281 of: Knowledge engineering and knowledge management. LNCS. Springer.

Roussey, C., Corcho, O., Blazquez, L.M. Vilches., 2009. A catalogue of owl ontology antipatterns. Pages 205–206 of: *Proc. of the 5th int. conf. on knowledge capture (k-cap 2009)*.

Roussey, C., Scharffe, F., Corcho, O., Zamazal, O., 2010. Une méthode de débogage d'ontologiesowl basées sur la détection d'anti-patrons. Pages 43–54 of: Actes de la *21e conférence en ingénierie des connaissances (IC2010)*.

Simperl, E., & Tempich, C., 2009. Exploring the economical aspects of ontology engineering. Pages 445–462 of: Studer, R., & Staab, S. (eds), Handbook on ontologies, *2 edn. International Handbooks on Information Systems*. Springer.

Tartir, S., Arpinar, I.B., Sheth, A.P., 2010. Ontological evaluation and validation. Pages 115–130 of: Poli, R., Healy, M., & Kameas, A. (eds), Theory and applications of ontology: Computer applications. Springer.

Volker, J., Sure, Y., 2006. Data-driven change discovery-evaluation. Tech. rept. Deliverable D3.3.2for SEKT Project, Institute AIFB, University of Karlsruhe.

Vrandecic, D., 2009. Ontology evaluation. Pages 293–314 of: Studer, R., Staab, S. (eds), Handbook on ontologies, *2 edn. Int. Handbooks on Inf. Syst*. Springer.