Problems impacting the quality of automatically built ontologies

Toader GHERASIM[1] and **Giuseppe BERIO**[2] and **Mounira HARZALLAH**[3] and **Pascale KUNTZ**[4]

Abstract. Building ontologies and debugging them is a time-consuming task. Over the recent years, several approaches and tools for the automatic construction of ontologies from textual resources have been proposed. But, due to the limitations highlighted by experimentations in real-life applications, different researches focused on the identification and classification of the errors that affect the ontology quality. However, these classifications are incomplete and the error description is not yet standardized. In this paper we introduce a new framework providing standardized definitions which leads to a new error classification that removes ambiguities of the previous ones. Then, we focus on the quality of automatically built ontologies and we present experimental results of our analysis on an ontology automatically built by Text2Onto for the domain of composite materials manufacturing.

1 Introduction

Since the pioneering works of Gruber [15], ontologies play a major role in knowledge engineering whose importance is growing with the rise of the semantic Web. Today they are an essential component in numerous applications in various fields: e.g. information retrieval [22, 20], knowledge management [26], analysis of social semantic networks [8] and business intelligence [27]. However, despite the maturity level reached in ontology engineering, important problems remain open and are still widely discussed in the literature. The most challenging issues concern the automation of ontology construction and their evaluation.

The increasing popularity of ontologies and the scaling changes of this last decade have motivated the development of ontology learning techniques. Promising results have been obtained [6, 5]. And, although these techniques have been often experimentally proved to be not sufficient enough for constructing ready-to-use ontology [5], their interest is not questioned in particular in technical domains [17]. Few recent works recommend an integration between ontology learning techniques and manual intervention [27].

Whatever their use, it is essential to assess their quality throughout their development. Several ontology quality criteria and different evaluation methods have been proposed in the literature [19, 4, 11, 21, 1]. However, as mentioned by [28], defining "a good ontology" remains a difficult problem and the different approaches only permit to "recognize problematic parts of an ontology". From an operational point of view, error identification is a very important step for the ontology integration in real-life complex systems. And, different researches recently focused on that issue [13, 2, 24]. However, as far as we know, a generic standardized description of these errors does not still exist. It seems however a preliminary step for the development of assisted construction method.

In this paper, we focus on the most important errors that affect the quality of semi-automatically built ontologies. To get closer the operational concerns we propose a detailed typology of the different types of problems that can be identified when evaluating an ontology. Our typology is inspired from a generic standardized description of the notion of quality in conceptual modeling [18]. And, our analysis is applied on a real-life situation concerning the manufacturing of pieces in composite materials for the aerospace industry.

The rest of this paper is organized as follows. Section 2 is a state-of-the art of the ontology errors. Section 3 describes a framework which provides a standardized description of the errors and draws correspondences between our new classification and the main errors previously identified in the literature. Section 4 presents our experimental results in the domain of composite materials manufacturing. More precisely, we analyze errors affecting an ontology produced by an automatic construction tool (here Text2Onto) from a set of technical textual resources.

2 State-of-the art on ontological errors

In the literature, the notion of "ontological error" is often used in a broad sense covering a wide variety of problems which affect the ontology quality. But, from several studies published this last decade, we have identified four major denominations associated to complementary definitions: (1) "taxonomic errors" [14, 13, 9, 2], (2) "design anomalies" or "deficiencies" [2, 3], (3) "anti-patterns" [7, 25, 23], and (4) "pitfalls" or "worst practices [23, 24].

2.1 Taxonomic errors

From the pioneering works of Gomez-Perrez [14], the denomination "taxonomic error" is used to refer to three types of errors that affect the taxonomic structure of ontologies: inconsistency, incompleteness and redundancy. Recently, extensions have been proposed to non-taxonomic properties [3], but in this synthesis we focus on taxonomic errors.

Inconsistencies in the ontology may be logical or semantic. More precisely, three classes of inconsistencies in the taxonomic structure have been detailed: circularity errors (e.g. a concept that is a specialization or a generalization of itself), partitioning errors which produce logical inconsistencies (e.g. a concept defined as a specialization of two disjoint concepts), and semantic errors (e.g. a taxonomic relationship between two concepts that is not consistent with the semantics of the latter).

[1] LINA, UMR 6241 CNRS, e-mail: toader.gherasim@univ-nantes.fr
[2] LABSTICC, UMR 6285 CNRS, email: giuseppe.berio@univ-ubs.fr
[3] LINA, UMR 6241 CNRS, e-mail: mounira.harzallah@univ-nantes.fr
[4] LINA, UMR 6241 CNRS, e-mail: pascale.kuntz@polytech.univ-nantes.fr

Incompleteness is met when concepts or relations of specialization are missing, or when some distributions of the instances of a concept between its sons are not stated as exhaustive and/or disjoint.

In the opposite way, *redundancy* errors are met when a taxonomic relationship can be directly deduced by logical inference from the other relationships of the ontology, or when concepts with the same father in the taxonomy do not share any common information (no instances, no children, no axioms, etc.) and can be only differentiated by their names.

2.2 Design anomalies

Roughly speaking, design anomalies mainly focus on ontology understanding and maintainability. They are not necessarily errors but undesirable situations. Five classes of design anomalies have been described: (1) "lazy concepts" (leaf concepts in the taxonomy not implied in any axiom and without any instances); (2) "chains of inheritance" (long chains composed of intermediate concepts with a single child); (3) "lonely disjoint" concepts (superfluous disjunction axiom between distant concepts in the taxonomy which may disrupt inference reasoning); (4) "over-specific property range" (too specific property range which should be replaced by a coarser range which fits the considered domain better); (5) "property clumps" (duplication of the same properties for a large set of concepts instead of the inheritance of these properties from a more general concept).

2.3 Anti-patterns

Ontology design patterns (ODP) are formal models of solutions commonly used by domain experts to solve recurrent modeling problems. Anti-patterns are ODP that are *a priori* known to produce inconsistencies or unsuitable behaviors. [23] also called anti-patterns *ad-hoc* solutions specifically designed for a problem even if well-known ODP are available. Three classes of anti-patterns have been described [7, 25, 23]: (1) "logical anti-patterns" that can be detected by logical reasoning; (2) "cognitive anti-patterns" (possible modeling errors due to misunderstanding of the logical consequences of the used expression); (3) "guidelines" (complex expressions valid from a logical and a cognitive point of view but for which simpler or more accurate alternatives exist).

2.4 Pitfalls

Pitfalls are complementary to ODPs. Their broad definition covers problems affecting the ontology quality for which ODPs are not available. Poveda et al. [24] described 24 types of experimentally identified pitfalls as, for instance, forgetting the declaration of an inverse relation when this latter exists or of the attribute range. And they proposed a pitfall classification which follows the three evaluable dimensions of an ontology proposed by Gangemi et al. [11]: (1) structural dimension (aspects related to syntax and logical properties), (2) functional dimension (how well the ontology fits a predefined function), (3) the usability dimension (to which extent the ontology is easy to be understood and used). Four pitfall classes correspond to the structural dimension: "modeling decisions" (*MD*, situations where OWL primitives are not used properly), "wrong inference" (*WI*, e.g. relationships or axioms that allow false reasoning), "no inference" (*NI*, gaps in the ontology which do not allow inferences required to produce new desirable knowledge), "real world modeling" (*RWM*, when commonsense knowledge is missing in the ontology). One class corresponds to the functional dimension: "requirement completeness" (*RC*, when the ontology does not cover its specifications). And, two classes correspond to the usability dimension: "ontology understanding" (*OU*, information that makes understandability more difficult e.g. concept label polysemy or label synonymy for distinct concepts, non explicit declaration of inverse relations or equivalent properties) and "ontology clarity" (*OC*, e.g. variations of writing-rule and typography for the labels).

It is easy to deduce from this classification that some pitfalls should belong to different classes associated to different dimensions (e.g. the fact that two inverse relations are not stated as inverse is both a "no inference" (*NI*) pitfall and an "ontology understanding" (*OU*) pitfall). Another attempt [24] proposed a classification of the 24 identified pitfalls in the three error classes (inconsistency, incompleteness and redundancy) given by Gomez-Perrez et al. [14]. But, these classes are concerned by the ontology structure and content, and consequently four pitfalls associated with the ontology context do not fit with this classification.

In order to highlight the links between the different classifications, Poveda et al. tried to define a mapping between the classification in 7 classes deduced from the dimensions defined by Gangemi et al. [11] and the 3 error classes proposed by Gomez-Perrez et al. [14]. However, this task turned out to be very complex, and only four pitfall classes exactly fit with one of the error classes. For the other, there is overlapping or no possible fitting.

3 The framework

The state of the art briefly presented in the previous section shows that the terminology used for describing the different problems impacting on the quality of ontologies is not yet standardized and that existing classifications do not cover the whole diversity of problems described in the literature.

In this section we present a framework providing standardized definitions for quality problems of ontologies and leading to a new classification of these problems. The framework comprises two distinct and orthogonal dimensions: errors vs. unsuitable situations (first dimension) and logical facet vs. social facet of problems (second dimension).

Unsuitable situations identify problems which do not prevent the usage of an ontology (within specific targeted domain and applications). On the contrary, errors identify problems preventing the usage of an ontology.

It is well known that one ontology has two distinct facets: an ontology can be processed by machines (according to its logical specification) and can be used by humans (including an implicit reference to a social sharing).

The remainder of the section is organized alongside the second dimension (i.e. logic vs. social facet) and within each facet, errors and unsuitable situations are defined. The framework is based on "natural" analogies between respectively social and logical errors and social and logical unsuitable situations.

3.1 Problem classification

3.1.1 *Logical ground problems*

The logical ground problems can be formally defined by considering notions defined by Guarino et al. [16]: e.g. Interpretation (Extensional first order structure), Intended Model, Language, Ontology and the two usual relations $\models$, $\vdash$ provided in any logical language. The relation $\models$ is used to express both that one interpretation I is a model of a logical theory T, written as $I \vDash T$ (i.e. all the formulas in T are true in I, written for each formula $\varphi \in T$, $I \vDash \varphi$), and also for expressing the logical consequence (i.e. that any model of a logical theory T is also a model of a formula, written as $T \vDash \varphi$). 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 T, written as $T \vdash \varphi$.

Examples and formalizations hereinafter are provided by using a typical Description Logics notation (but easily transformable in first order or other logics).

The usual **logical ground errors** are listed below.

1. *Logical inconsistency* corresponding to ontologies containing logical contradictions for which a model does not exist (because the set of intended models is never empty, an ontology without models does not make sense anyway; formally, given an ontology O and the logical consequence relation $\vDash$ according to the logical language L used for building O, there is no interpretation I of O such that $I \vDash O$). For example, if an ontology contains the following axioms $B \subseteq A$ (B *is_a* A), $A \cap B \subseteq \top$ (A and B are $disjoint$), $c \subseteq B$ (c is $instance\_of\ B$), then $c \subseteq A$ and $c \subseteq A \cap B$, so there is a logical contradiction in the definition of this ontology;
2. *Unadapted*[5] *ontologies* wrt to intended models[6] i.e. an ontology for which something that is false in all (some of) the intended models of L is true in the ontology; formally, there exists a formula φ such that for each (for some) intended model(s) of L, φ is false and $O \vDash \varphi$. For example, if we have in the ontology two concepts A and B that are declared as disjoint ($O \vDash A \cap B \subseteq \perp$) and in each intended model there exists an instance c that is common between A and B (i.e. $c \subseteq A \cap B$), then the ontology is unadapted;
3. *Incomplete ontologies* wrt to intended models i.e. an ontology for which something that is true in all the intended models of L, is not necessarily true in all the models of O; formally, there exists a formula φ such that for each intended model of L, φ is true and $O \nvDash \varphi$. As an example, if in all the intended models $C \cup B = A$, and the ontology O defines $B \subseteq A$ and $C \subseteq A$, it is not possible to prove that $C \cup B = A$;
4. *Incorrect (or unsound) reasoning* wrt the logical consequence i.e. when some specific conclusions are derived by using suitable reasoning systems for targeted ontology applications even if these conclusions are not true in the intended models and must not be derived by any reasoning according to the targeted ontology applications (formally, when a specific formula φ, false in the intended models $O \nvDash \varphi$, can be derived $O \vdash \varphi$ within any of those suitable reasoning systems);
5. *Incomplete reasoning* wrt the logical consequence i.e. when some specific conclusions cannot be derived by using suitable reasoning systems for targeted ontology applications even if these conclusions are true in intended models and must be derived by some reasoning according to the targeted ontology applications (formally, for some specific formula φ, true in the intended models $O \vDash \varphi$, cannot be derived $O \nvdash \varphi$ within those suitable reasoning systems);

The most common **logical ground unsuitable situations** are listed below. These situations impact negatively on the "non functional qualities" of ontologies such as reusability, maintainability, efficiency as defined in the ISO 9126 standard for software quality.

6. *Logical equivalence of distinct artifacts* (concepts / relationships / instances) i.e. whenever two distinct artifacts are proved to be logically equivalent; for example, A and B are two concepts in O and $O \vDash A = B$;
7. *Symmetrically, logically indistinguishable artifacts* i.e. whenever it is not possible to prove that two distinct artifacts are not equivalent from a logical point of view; in other words, if not possible to prove anyone of the following statements: ($O \vDash A = B$), ($O \vDash A \cap B \subseteq \perp$) and ($O \vDash c \subseteq AandO \vDash c \subseteq B$); this case (7) can be partially covered in the case (3) above whenever intended models provide precise information on the equivalence or the difference between A and B;
8. *OR artifacts* i.e. an artifact A equivalent to a disjunction like $C \cup S$, $A \neq C, S$ but for which, if applicable, it does not exist at least a common (non optional) role / property for C and S or because C and S have common instances; in the first case, a simple formalization can be expressed by saying that it does not exist a (non optional) role R such that $O \vDash (C \cup S) \subseteq \exists R.\top$; in the second case, an even simpler formalization is $O \vDash c \subseteq C$ and $O \vDash c \subseteq S$, being c one constant not part of O; the first case targets **potentially heterogeneous artifacts** such as $Car \cup Person$, with probably no counterpart in the intended models, thus possibly leading to unadapted ontologies according to case (2) above; the second case targets **potential ambiguities** as, for instance, one role (property) R logically equivalent to a disjunction ($R_1 \cup R_2$) being ($R_1 \cap R_2$) satisfiable;
9. *AND artifacts* i.e. one artifact A equivalent to a conjunction like $C \cap S$, $A \neq C, S$ but for which, if applicable, it does not exist at least a common (non optional) role / property for C and S; this case is relevant to limit as much as possible some **potentially heterogeneous artifacts** such as $Car \cap Person$, possibly leading to artifact unsatisfiability;
10. While some case of *unsatisfiability* of ontology artifacts (concepts, roles, properties etc.) can be covered by (2) because intended models may not contain void concepts, unsatisfiability tout-court is not necessarily an error but a situation which is not suitable for ontology artifacts (i.e. given an ontology artifact A, $O \vDash A \subseteq \perp$); even if in ontologies it might be possible to define what must not be true (instead of what must be true), this practice is not encouraged;
11. *High complexity of the reasoning task* i.e. whenever something is expressed in a way that complicates the reasoning, while there exist more simple ways to express the same thing;
12. *Ontology not minimal* i.e. whenever the ontology contains unnecessary information:
    - Unnecessary because it can be derived or built[7]. An example of such unsuitable situation is the redundancy of taxonomic relations such as whenever $A \subseteq B$, $B \subseteq C$, and $A \subseteq C$ are all ontology axioms, the last axiom can be derived from the first two ones;

[5] We use the term "unadapted" instead of "incorrect" ontologies because it remains unclear if intended models are defined for building the ontology or may also be defined independently. However, if intended models are defined for building the ontology, the term "incorrect" may be more appropriate.

[6] Intended models should have been defined fully and independently as in the case of models representing abstract structures or concepts such as numbers, processes, events, time and other "upper concepts", often defined according to their own properties. If intended models are not available, some specific entailments can be defined as facts that should necessarily be true in the targeted domain (or for targeted applications); specific counterexamples can also be defined instead of building entire intended models.

[7] Built means that the artifact can be defined by using other artifacts.

- Unnecessary because it is not part of the intended models. For instance, a concept A being part of the ontology (language) but not defined by intended models.

3.1.2 Social ground problems

Social ground problems are related to the **perception (interpretation)** and the **targeted usage** of ontologies by social actors (humans, applications based on social artifacts like WordNet, etc.). Perception (interpretation) and usage may not be formalized at all. In some sense, a further distinction between social facet and logical facet is as the distinction between respectively tacit and explicit knowledge.

There are four **social ground errors**:

1. *Social contradiction* i.e. the perception (interpretation) that the social actor gives to the ontology or to the ontology artifacts is in contradiction with the ontology axioms and their consequences; a natural analogy is with *unadapted ontologies*;
2. *Perception of design errors* i.e. the social actor perception accounts for some design errors such as modeling instances as concepts; a natural analogy is with *unadapted ontologies*;
3. *Socially meaningless* i.e. the social actor is unable to give any interpretation to the ontology or to ontology artifacts as in the case of artificial labels such as "XYHG45"; a natural analogy is with *unadapted ontologies*;
4. *Social incompleteness* i.e. the social actor perception is that one or several artifacts (axioms and/or their consequences) are missing in the ontology; a natural analogy is with *incomplete ontologies*;

The **social ground unsuitable situations** are mostly related to the difficulties that a social actor has to overcome for using the ontology especially due to limited understandability, learnability and compliance (as defined in ISO 9126). As for the logical ground unsuitable situations, it is difficult to dress an exhaustive list; the most common and important are listed below.

5. *Lack of or poor textual explanations* i.e. when there are few, no or poor annotations; prevents understanding by social actors; there are no natural analogies;
6. *Potentially equivalent artifacts* i.e. the social actors may identify as equivalent (similar) distinct artifacts as in the case of artifacts with synonymous or exactly the same labels assigned to distinct artifacts; a natural analogy is with *logically equivalent artifacts*;
7. *Socially indistinguishable artifacts* i.e. the social actors would not be able to distinguish two distinct artifacts as, for instance, in the case of artifacts with polysemic labels assigned to distinct artifacts; a natural analogy is with *logically indistinguishable artifacts*;
8. *Artifacts with polysemic labels* may be interpreted as union or intersection of their several rather distinct meanings associated to labels; a natural analogy is therefore with *OR* and *AND artifacts*.
9. *Flatness of the ontology* (or non modularity), i.e. ontology presented as a set of artifacts without any additional structure, especially if coupled with a important number of artifacts; a natural analogy is with *high complexity of the reasoning task* but also preventing effective learning and understanding by social actors;
10. *Non-standard formalization of the ontology*, using a very specific logics or theory, requires a specific effort by social actors for understanding and learning the ontology but also to use the ontology in standard contexts (reduced compliance); there are no natural analogies;
11. *Lack of adapted and certified versions of the ontology in various languages* requires specific efforts by social actors for understanding and learning the ontology but also to use the ontology in specific standard contexts (limited compliance); there are no natural analogies;
12. *Socially useless artifacts* included in the ontology; a natural analogy is with *ontology not minimal*.

3.2 Positioning state of the art relevant problem classes in to the proposed framework

The precise definitions of the proposed framework allow us to classify most of the ontology quality problems described in literature. Table 1 presents our classification of the different problems mentioned in Section 2. Some of the problems described in literature may correspond to more than one class of problems from our framework, as the definitions of these problems are often very large and sometimes ambiguous.

Table 1 reveals, at a first view, that the proposed framework provides additional problems that are not directly pointed out, to our knowledge, in the current literature about ontology quality and evaluation (but may be mentioned elsewhere). These problems are *No adapted and certified ontology version*, *Indistinguishable artifacts*, *Socially meaningless*, *High complexity of the reasoning task* and *Incorrect reasoning*. However, while covered, other problems are, in our opinion, too much narrowly defined in existing literature about ontology quality and evaluation. For instance, *No standard formalization* is specific to very simple situations while we refer to complete non standard theories.

A deeper analysis of Table 1 reveals that the "logical anti-patterns" presented in [7, 25] belong to the logical ground category and are focusing on *unadapted ontologies* error and *unsatisfability* unsuitable situation. The "non-logical anti patterns" presented in [7, 25] partially cover the logical ground unsuitable situations. The "guidelines" presented in [7, 25] span only over unsuitable situations from both logical and social ground category.

What is qualified as "inconsistency" in [14] span over errors and unsuitable situations and also (as in the case of "semantic inconsistency") over the two dimensions (logical and social), making, in our opinion, the terminology a little bit confusing. According to our framework, we perceive "circularity in taxonomies", as defined in [14], as an unsuitable situation (*logical equivalence of distinct artifacts*) because, from a logical point of veiw, this only means that artifacts are equivalent (not requiring a fixpoint semantics). However, "circularity in taxonomies" can be seen also within a *social contradiction* if actors assign distinct meanings to the various involved artifacts. The problems presented as "incompleteness errors" in [13] belong to the *incomplete ontologies* class of logical errors. The "redundancy errors" fits, in our classification, within the *ontology not minimal* class of logical unsuitable situations.

None of the "design anomalies" presented in [2] is perceived as a logical error. Two of them correspond to a logical unsuitable situation (*logically undistinguishable artifacts*), one to a social error (*perception of design errors*) and the last one to a social unsuitable situation (*no standard formalization*).

Concerning "pitfalls" [24], the most remarkable fact concerns what we call *incomplete reasoning*. Indeed, introducing *ad-hoc* relations such as is_a, $instance\_of$, etc., replacing the "standard" relations such as $subsumption$, $member\_of$, etc., should not be considered as a case of *incomplete ontologies* but as a case of *incomplete reasoning*. This is because accepting a specific ontological commit-

Table 1. Positioning state of the art relevant problem classes in to the proposed framework.

Framework				State of the art problems
Logical ground	**Errors**	1	Logical inconsistency	➢ inconsistency error: ”partition errors - common instances in disjoint decomposition”
		2	Unadapted ontologies	➢ inconsistency errors: ”partition errors - common classes in disjoint decomposition”, ”semantic inconsistency” ➢ logical anti-patterns: ”OnlynessIsLoneliness”, ”UniversalExistence”, ”AndIsOR”, ”EquivalenceIsDifference” ➢ pitfalls: P5 (wrong inverse relationship, *WI*), P14 (misusing ”allValuesFrom”, *MD*), P15 (misusing ”not some”/”some not”, *WI*), P18 (specifying too much the domain / range, *WI*), P19 (swapping ∩ and ∪, *WI*)
		3	Incomplete ontologies	➢ incompleteness errors: ”incomplete concept classification”, ”disjoint / exhaustive knowledge omission” ➢ pitfalls: P3 (”is a” instead of ”subclass-of”, *MD*), P9 (missing basic information, *RC & RWM*), P10 (missing disjointness, *RWM*), P11 (missing domain / range in prop., *NI & OU*), P12 (missing equiv. prop., *NI & OU*), P13 (missing inv. rel., *NI & OU*), P16 (misusing primitive and defined classes, *NI*)
		4	Incorrect reasoning	
		5	Incomplete reasoning	➢ pitfalls: P3 (using ”is a” instead of ”subclass-of”, *MD*), P24 - using recursive def., *MD*)
	Unsuitable situations	6	Logical equivalence of distinct artifacts	➢ inconsistency error: ”circularity” ➢ pitfall: P6 (cycles in the hierarchy, *WI*) ➢ non logical anti-pattern: ”SynonymeOfEquivalence”
		7	Logically indistinguishable artifacts	➢ pitfall: P4 (unconnected ontology elements, *RC*) ➢ design anomalies: ”lazy concepts” and ”chains of inheritance”
		8	OR artifacts	➢ pitfall: P7 (merging concepts to form a class, *MD & OU*)
		9	AND artifacts	➢ pitfall: P7 (merging concepts to form a class, *MD & OU*)
		10	Unsatisfiability	inconsistency error: ”partition errors - common classes in disjoint decomposition” ➢ logical anti-patterns: ”OnlynessIsLoneliness”, ”UniversalExistence”, ”AndIsOR”, ”EquivalenceIsDifference”
		11	High complexity of the reasoning task	
		12	Ontology not minimal	➢ redundancy error: ”redundancy of taxonomic relations” ➢ pitfalls: P3 (using ”is a” instead of ”subclass-of”, *MD*), P7 (merging concepts to form a class, *MD & OU*), P21 (miscellaneous class, *MD*) ➢ non logical anti-pattern: ”SomeMeansAtLeastOne” ➢ guidelines: ”Domain&CardinalityConstraints”, ”MinIsZero”
Social ground	**Errors**	1	Social contradiction	➢ inconsistency error: ”semantic inconsistency” ➢ logical anti-pattern: ”AndIsOR” ➢ pitfalls: P1 (polysemic elements, *MD*), P5 (wrong inv. rel., *WI*), P14 (misusing ”allValuesFrom”, *MD*), P15 (misusing ”not some”/”some not”, *WI*), P19 (swapping ∩ and ∪, *WI*)
		2	Perception of design errors	➢ pitfalls: P17 (specializing too much the hierarchy, *MD*), P18 (specifying too much the domain / range, *WI*), P23 (using incorrectly ontology elements, *MD*) ➢ non logical anti-pattern: ”SumOfSome” ➢ design anomaly: ”lonely disjoints”
		3	Socially meaningless	
		4	Social incompleteness	➢ pitfalls: P12 (missing equiv. prop., *NI & OU*), P13 (missing inv. rel., *NI & OU*), P16 (misusing primitive and defined classes, *NI*)
	Unsuitable situations	5	Lack/poor textual explanations	➢ pitfalls: P8 (missing annotation, *OC & OU*)
		6	Potentially equiv. artifacts	➢ pitfalls: P2 (synonym as classes, *MD & OU*)
		7	Indistinguishable artifacts	
		8	Polysemic labels	➢ pitfalls: P1 (polysemic elements, *MD & OU*)
		9	Flatness of the ontology	
		10	No standard formalization	➢ pitfalls: P20 (swapping label and comment, *OU*), P22 (using different naming criteria in the ontology, *OC*) ➢ guidelines: ”GroupAxioms”, ”DisjointnessOfComplement” and ”Domain&CardinalityConstraints” ➢ design anomaly: ”property clumps”
		11	No adapted and certified ontology version	
		12	Useless artifacts	➢ pitfall: P21 (using a miscellaneous class, *MD & OU*)

ment for building intended models, *ad-hoc* relations can be defined in the same way as standard relations. However, using standard reasoning it is expected (and even proved once fixing the logics) that reasoning algorithms are incomplete. However, adding artifacts may also solve some incompleteness and may also be useful for speeding up reasoning.

Only one of the seven classes of ”pitfalls” [24] perfectly fits in one class of our typology: the ”real world modeling” pitfalls belong to the *incomplete ontologies* logical errors. All the ”ontology clarity” pitfalls are social unsuitable situations. All the ”requirement completeness” pitfalls are logical problems. The ”no inference” pitfalls are logical or social *incomplete ontologies* errors. Most (6/9 and 4/5) of the ”modeling decisions” and ”wrong inference” pitfalls are considered as errors. The class of ”ontology understanding” pitfalls spans over 10 classes of problems, covering logical and social errors and unsuitable situations.

Most (16/20) of the pitfalls concerning the ”structural dimension” of the ontology [11] are perceived as errors. All (2/2) the pitfalls concerning the ”functional dimension” of the ontology are logical problems.

4 Problems that affect the quality of automatically built ontologies

Although the proposed framework is general, we are especially concerned by ontologies automatically built from textual resources. We therefore aim at pointing the problems that are expected in automatically constructed ontologies (i.e. there is evidence of their presence or they will appear in future enrichments[8] of the ontology). We are also interested by the opposite case, i.e. if there are unexpected problems in automatically constructed ontologies: it should be noted that unexpected problems are problems that even if the ontology may suffer of them, there is no evidence of their presence/absence for the ontology as it is (however, these problems may appear in future enrichments of the ontology). Our analysis is performed in two steps. In the first step (Section 4.1), we point out expected/unexpected problems due to inherent limitations of the tools for automatic ontology construction. In the second step (Section 4.2), we assess the results obtained in the first step by discussing our experience with the tool Text2Onto.

4.1 Expected and unexpected problems in an automatically built ontology

In a previous work [12] we have deeply studied four approaches (and associated tools) for the automatic construction of ontologies form texts and we compared them with a classical methodology for manual ontology construction (Methontology). This analysis highlighted that none of the automated approaches (and associated tools) covers all the tasks and subtasks associated to each step of the classical manual method. The ignored tasks/subtasks are:

1. The explicit formation of artifacts (concepts, instances and relationships) from terms[9]; usually, the automatic tools consider that each term represents a distinct artifact: they do not group synonymous terms and do not choose a single sense for polysemic terms
2. The identification of axioms (e.g. the disjunction axioms)
3. The identification of attributes for concepts
4. The identification of natural language definitions for concepts

[8] Enrichment should be understood as adding artifacts to the existing ones.
[9] A term corresponds to one or several words found in one text.

Table 2. What problems are expected in automatically built ontologies.

Types of problems	Expected (Yes/No) and Why
1. Logical inconsistency	N (no axiom is defined ⇒ contradictions are unexpected; but they remain possible in the case of future enrichments)
2. Unadapted ontologies	Y (taxonomic relationships extraction algorithms are syntax based ≠ from the intended models)
3. Incomplete ontologies	Y (automatically extracted knowledge is limited to concepts and taxonomies ≠ from the intended models)
4. Incorrect reasoning	N (they might appear for complete formalization of concepts and relationships)
5. Incomplete reasoning	
6. Logical equivalence of distinct artifacts	Y (automatic tools consider that each term defines a different artifact ⇒ the ontology may contain logically equivalent & logically indistinguishable artifacts)
7. Logically indistinguishable artifacts	
8. OR artifacts	Y (polysemy of terms directly affects concepts / relationships: OR / AND concepts / relationships may appear)
9. AND artifacts	
10. Unsatisfiability	Y (polysemy of terms directly affects concepts / relationships: these latter may become unsatisfiable if their polysemic senses are combined)
11. High complexity of the reasoning task	N (few or no axioms are defined ⇒ reasoning remains very basic; but, it can be more complex if the ontology is further enriched)
12. Ontology not minimal	Y (automatic tools introduce redundancies in taxonomies)
1. Social contradiction	Y (ontologies are built from limited textual resources which may introduce contradiction in taxonomies)
2. Perception of design errors	Y (the built ontology may contain concepts that are considered more close to instances by the social actor.)
3. Social meaningless	Y (several meaningless concepts with obscure labels are often introduced)
4. Social incompleteness	Y (probably due to limited textual corpus)
5. Lack of or poor textual explanations	Y (usually automatic tools do not provide textual explanations)
6. Potentially equivalent artifacts	Y (automatic tools consider that each term defines a different artifact ⇒ distinct concepts can have synonymous labels ⇒ these latter are perceived as potentially equivalent)
7. Indistinguishable artifacts	Y (the ontology is incomplete ⇒ it contains concepts that can be distinguished only by their labels; if such concepts have synonymous labels, they are indistinguishable)
8. Artifacts with polysemic labels	Y (automatic tools consider that each term defines a different artifact ⇒ it is possible to have concepts with polysemic labels)
9. Flatness of the ontology	Y (the ontology is poorly structured and has no design constraints - e.g. no disjunction axiom, lazy concepts)
10. No standard formalization	N (automatic tools usually can export their results in different formalization)
11. No adapted and certified ontology version	Y (automatically obtained results closely depend on the input texts language (often English) and certifying them is difficult)
12. Useless artifacts	Y (automatic tools often generate useless artifacts from additional external resources)

Table 2 provides a complete view of expected and unexpected problems according to our experience and suggest why each problem is expected or not.

4.2 Experience with Text2Onto

4.2.1 The experimental setup

During the last two years we were implied in a project called ISTA3 that proposed an ontology based solution for problems related to the integration of heterogeneous sources of information. The application domain was the management of the production of composite components for the aerospace industry. In this context, we tried to simplify the process of deploying the interoperability solution in new domains by using automatic solution for constructing the required ontologies.

The analysis presented in [12] conducted us to choose Text2Onto [6] for the automatic construction of our ontologies. Text2Onto takes as input textual resources from which it extracts different ontological artifacts (concepts, instances, taxonomic relationships, etc.) that are structured together to construct an ontology. Text2Onto performances for extracting concepts and taxonomical relationships are better than its performances for extracting other types of ontological artifacts; consequently, in our tests we used Text2Onto for constructing ontologies containing concepts and taxonomical relationships only.

The textual resource used in the experiment presented in this paper is a technical glossary composed of 376 definitions of the most important terms of the domain of composite materials and how are they used for manufacturing pieces. The glossary contains 9500 words. For constructing the ontology we resort to the standard configuration for the different parameters of Text2Onto: all the proposed algorithms for concepts (and respectively for taxonomic relations) extractions have been used and their results have been combined with the default strategy.

The constructed ontology is an automatically built domain ontology that contains 965 concepts and 408 taxonomic relationships. Some of the central concepts of this ontology are: ”technique”, ”step”, ”compound”, ”fiber”, ”resin”, ”polymerization”, ”laminate”, ”substance”, ”form”.

4.2.2 Identified problems

Table 3 summarizes which types of problems have been identified in the automatically constructed ontology in our experience with Text2Onto. It also indicates, when possible, how many problems have been identified. Most of problems are relatively easy to identify and to quantify (e.g. the number of cycles in the taxonomical structure), but there are exceptions (e.g. the number of concepts or taxonomic relationships that are missing from the ontology).

4.2.3 Discussion

No intended model or use case scenario was available when the expert analyzed the automatically constructed ontology. Consequently, it was able only to make a supposition concerning the logical completeness of the ontology and no logical error (*unadapted ontology*, *incomplete* or *incorrect reasoning*) was identified.

Few logical unsuitable situations are identified, but it is remarkable that they were identified automatically.

Unsurprisingly, most of the identified problems are social problems.

Table 3. Types of problems identified in the automatically constructed ontology.

Types of problems	**Identyfied (Yes/No) and How**
1. Logical inconsistency	No
2. Unadapted ontologies	No
3. Incomplete ontologies	Yes: Some relationships are missing to connect the 389 lazy concepts; some of them are explicitly indicated in the textual corpus
4. Incorrect reasoning	No
5. Incomplete reasoning	No
6. Logical equivalence of distinct artifacts	Yes: 3 cycles in the hierarchy; (automatically detected by reasoners)
7. Logically indistinguishable artifacts	Yes: * 389 lazy concepts (automatically identified by an ad-hoc algorithm) *73 groups of ”leaf” concepts; each group is composed of concepts that are indistinguishable; (automatically identified by an ad-hoc algorithm)
8. OR artifacts	No
9. AND artifacts	No
10. Unsatisfiability	No
11. High complexity of the reasoning task	No
12. Ontology not minimal	Yes: one taxonomical relationship can be deduced from two taxonomical relationships already present in the ontology (automatically identified by an ad-hoc algorithm)

1. Social contradiction	Yes: 15 taxonomic relationships are jugged semantically inconsistent by the expert
2. Perception of design errors	Yes: 5 concepts that are interpreted as instances by the expert (units of measure and proper names)
3. Social meaningless	Yes: 21 concepts that have meaningless labels, for the expert
4. Social incompleteness	Yes
5. Lack of or poor textual explanations	Yes: no annotation associated to the ontology or to its artifacts
6. Potentially equivalent artifacts	Yes: 6 pairs of concepts have synonym labels, for the expert
7. Indistinguishable artifacts	No
8. Artifacts with polysemic labels	Yes: 69 concepts with polysemic labels, for the expert
9. Flatness of the ontology	Yes: 389 lazy concepts lead to a poorly structured ontology
10. No standard formalization	No
11. No adapted and certified ontology version	No
12. Useless artifacts	Yes: 28 concepts are not necessary (3 are too generic, 25 are out of the domain)

The analysis in Section 4.1 suggest that most of the problems that are expected in the automatically constructed ontologies are due to the fact that the automatic tool do not take into account the synonymy and the polysemy of terms when constructing concepts. However, even if Text2Onto, as configured for our test, do not group synonym terms when forming concepts, and allows polysemic terms to be labels for concepts, our test-case reveals that only two types of problems (*socially indistinguishable artifacts* and artifacts with *polysemic labels*) may be imputed to this limitation.

Most of the identified problems are related to the fact that the automatically constructed ontology seems to be incomplete.

5 Conclusion

In this paper, we have introduced a framework providing standardized definitions for different errors that have some impact on the quality of the ontologies. This framework aims at both unifying various error descriptions presented in the recent literature and completing them. It also leads to a new error classification that removes ambiguities of the previous ones. During ontology evaluation this framework may be used as a support for verifying in a systematic way if the ontology contains errors or unsuitable situations.

In the second part of the paper we focused on the quality of automatically built ontologies and we present experimental results of our analysis on an ontology automatically built by Text2Onto. The results show that a large part of the identified errors are linked to the ontology incompleteness. Moreover, it confirms that the identification of logical errors other than inconsistency requires intended models (or at least a set of positive and negative examples) and use case scenarii.

Due to the increasing complexity of the software, the identification of the origin of each error in the ontology building process remains an open question. And a further works consists in associating the identified errors with the different tasks of an ontology construction (e.g. the Methontology tasks [10]). This work could help to improve the quality results of the software by a retro-engineering process and/or to design assistant to detect and to solve major errors.

REFERENCES

[1] M. Almeida, 'A proposal to evaluate ontology content', *Journal of Applied Ontology*, **4(3-4)**, 245–265, (2009).

[2] J. Baumeister and D. Seipel, 'Smelly owls design anomalies in ontologies', in *Proc. of 18th Int. Florida Artificial Intelligence Research Society Conf. (FLAIRS)*, pp. 215–220, (2005).

[3] J. Baumeister and D. Seipel, 'Anomalies in ontologies with rules', *Web Semantics: Science, Services and Agents on the World Wide Web*, **8**(1), 55–68, (2010).

[4] A. Burton-Jones, V. Storey, and V. Sugumaran, 'A semiotic metrics suite for assessing the quality of ontologies', *Data Knowl. Eng.*, **55(1)**, 84–102, (2005).

[5] P. Cimiano, A. Madche, S. Staab, and J. Volker, 'Ontology learning', in *Handbook on Ontologies*, eds., R. Studer and S. Staab, Int. Handbook on Inf. Syst., 245–267, Springer, 2 edn., (2009).

[6] P. Cimiano and J. Volker, 'Text2onto - a framework for ontology learning and data-driven change discovery', in *2nd Eur. Semantic Web Conf.*, eds., A. Montoyo, R. Munoz, and E. Metais, volume 3513, pp. 227–238, (2005).

[7] O. Corcho, C. Roussey, and L. M. V. Blazquez, 'Catalogue of antipatterns for formal ontology debugging', in *Atelier Construction d'ontologies: vers un guide des bonnes pratiques, AFIA 2009*, (2009).

[8] G. Ereteo, M. Buffa, O. Corby, and F. Gandon, 'Semantic social network analysis: A concrete case', in *Handbook of Research on Methods and Techniques for Studying Virtual Communities: Paradigms and Phenomena*, 122–156, IGI Global, (2010).

[9] M. Fahad and M. Qadir, 'A framework for ontology evaluation', in *Proc. of the 16th Int. Conf. on Conceptual Struct. (ICCS2008)*, volume 354, pp. 149–158, (2008).

[10] M. Fernandez, A. Gomez-Prez, and N. Juristo, 'Methontology: From ontological art towards ontological engineering', in *Proc. of the AAAI97 Spring Symposium Series on Ontological Engineering*, pp. 33–40, (1997).

[11] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann, 'Modelling ontology evaluation and validation', in *Proc. of Eur. Sem. Web Conf. (ESWC2006)*, number 4011 in LNCS, (2006).

[12] T. Gherasim, M. Harzallah, G. Berio, and P. Kuntz, 'Analyse comparative de methodologies et d'outils de construction automatique d'ontologies a partir de ressources textuelles', in *Proc. of EGC'2011*, pp. 377–388, (2011).

[13] A. Gomez-Perez, 'Ontology evaluation', in *Handbook on Ontologies*, eds., S. Staab and R. Studer, Int. Handbook on Inf. Syst., pp. 251–274, Springer, 1 edn., (2004).

[14] A. Gomez-Perez, M.F. Lopez, and O.C. Garcia, *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web*, chapter Chap 3.8.2 Taxonomy evaluation, 180–184, Advanced Information and Knowledge Processing, Springer, 2001.

[15] T. R. Gruber, 'A translation approach to portable ontology specifications', *Knowl. Acquisition*, **5(2)**, 199–220, (1993).

[16] N. Guarino, D. Oberle, and S. Staab, 'What is an ontology?', in *Handbook on Ontologies*, 1–17, Springer, 2 edn., (2009).

[17] G. Hirst, 'Ontology and the lexicon', in *Handbook on Ontologies*, eds., R. Studer and S. Staab, Int. Handbook on Inf. Syst., 269–292, Springer, 2 edn., (2009).

[18] J. Krogstie, O.I. Lindland, and G. Sindre, 'Defining quality aspects for conceptual models', in *Proc. of the IFIP8.1 Working Conference on Information Systems Concepts: Towards a Consolidation of Views (ISCO3)*, (1995).

[19] A. Lozano-Tello and A. Gomez-Perez, 'Ontometric: A method to choose the appropriate ontology', *Journal of Database Management*, **15(2)**, 1–18, (2004).

[20] N. Ben Mustapha, H. Baazaoui Zghal, M.A. Aufaure, and H. Ben Ghezala, 'Enhancing semantic search using case-based modular ontology', in *Proc. of the 2010 ACM Symposium on Applied Computing*, pp. 1438–1439, (2010).

[21] L. Obrst, B. Ashpole, W. Ceusters, I. Mani, S. Ray, and B. Smith, 'The evaluation of ontologies: toward improved semantic interoperability', in *SemanticWeb: Revolutionizing Knowledge Discovery in the Life Sciences*, ed., K.-H. Cheung C. J. O. Baker, 139–158, Springer, (2007).

[22] J.D. Osborne, J. Flatow, M. Holko, S.M. Lin, W.A. Kibbe, L. Zhu, M.I. Danila, G. Feng, and R. L. Chisholm, 'Annotating the human genome with disease ontology', *BMC Genomics*, **10**, 63–68, (2009).

[23] M. Poveda, M. C. Suarez-Figueroa, and A. Gomez-Perez, 'Common pitfalls in ontology development', in *Proc. of the Current topics in artificial intelligence (CAEPIA09), and 13th conference on Spanish association for artificial intelligence*, (2009).

[24] M. Poveda, M. C. Suarez-Figueroa, and A. Gomez-Perez, 'A double classification of common pitfalls in ontologies', in *Proc. of Workshop on Ontology Quality (OntoQual 2010), Co-located with EKAW 2010*, (2010).

[25] C. Roussey, O. Corcho, and L. M. V. Blzquez, 'A catalogue of owl ontology antipatterns', in *Proc. of the Fifth Int. Conf. on Know. Capture KCAP*, pp. 205–206, (2009).

[26] N. H. Shah and M. A. Musen, 'Ontologies for formal representation of biological systems', in *Handbook on Ontologies*, eds., R. Studer and S. Staab, Int. Handbook on Inf. Syst., 445–462, Springer, 2 edn., (2009).

[27] E. Simperl and Tempich C., 'Exploring the economical aspects of ontology engineering', in *Handbook on Ontologies*, eds., R. Studer and S. Staab, Int. Handbook on Inf. Syst., 445–462, Springer, 2 edn., (2009).

[28] D. Vrandecic, 'Ontology evaluation', in *Handbook on Ontologies*, eds., R. Studer and S. Staab, Int. Handbook on Inf. Syst., 293–314, Springer, 2 edn., (2009).