Evolving Ontology Evolution

Giorgos Flouris, Dimitris Plexousakis, Grigoris Antoniou

Department of Computer Science, University of Crete
P.O. Box 2208, GR 71409, Heraklion, Greece
Institute of Computer Science, FO.R.T.H.
P.O. Box 1385, GR 71110, Heraklion, Greece
`{fgeo, dp, antoniou}@ics.forth.gr`

Abstract. One of the crucial tasks towards the realization of the Semantic Web vision is the efficient encoding of human knowledge in ontologies. Thus, the proper maintenance of these, usually large, structures and, in particular, their adaptation to new knowledge (ontology evolution) is one of the most challenging problems in the current Semantic Web research. In this paper, we uncover a certain gap in the current research area of ontology evolution and propose a research direction based on belief revision. We present some results in this direction and argue that our approach introduces an interesting new dimension to the problem that is likely to find important applications in the future.

1 Introduction

Originally introduced by Aristotle, *ontologies* are often viewed as the key means through which the vision of the Semantic Web can be realized [4]. The importance of ontologies in current research is emphasized by the interest shown by both the research and the enterprise community to various ontology-related problems [27].

Ontologies are often large and complex structures, whose development and maintenance give rise to certain sturdy and interesting research problems. One of the most important such problems is *ontology evolution*, which is the problem of modifying an ontology in response to a certain change in the domain or its conceptualization.

There are several cases where ontology evolution is applicable. An ontology, just like any structure holding information, may need to change simply because the world has changed [31]; in other cases, we may need to change the perspective under which the domain is viewed [29], or we may discover a problem in the original conceptualization of the domain; we might also wish to incorporate additional functionality, according to a change in users' needs [13]; furthermore, new information, which was previously unknown, classified or otherwise unavailable may become accessible or different features of the domain may become important [18].

In this paper, we argue that the currently used ontology evolution model has several weaknesses. We present an abstract proposition for a future research direction that will hopefully resolve these weaknesses, based on the related field of *belief change* [11]. Finally, we present an application of our research model in which the AGM theory [1] is generalized so as to be applicable to ontology evolution.

2 Ontology Evolution

An ontology can be defined as a specification of a conceptualization of a domain [23]. Thus, ontology evolution may be caused by either a change in the domain, a change in the conceptualization or a change in the specification [23]. Our understanding of the term ontology evolution covers the first two types of change (changes in the domain and changes in the conceptualization). The third type of change (change in the specification) refers to a change in the way the conceptualization is formally recorded, i.e., a change in the representation language; this is dealt with in the field of *ontology translation* [7], [20]. Unlike [30], we don't consider the (important) issue of propagating the changes to dependent elements, as this part of ontology change is handled by the related field of *ontology versioning* [23].

In order to tame the complexity of the problem, six phases of ontology evolution have been identified, occurring in a cyclic loop [30]. Initially, we have the *change capturing* phase, where the changes to be performed are identified; these changes are represented in a suitable format during the *change representation* phase. There are two major types of changes, namely elementary and composite changes [30]. Elementary changes represent simple, fine-grained changes; composite changes represent more coarse-grained changes and can be replaced by a series of elementary changes. However, it is not generally appropriate to use a series of elementary changes to replace a composite change, as this might cause undesirable side-effects [30]. The proper level of granularity should be identified at each case. Examples of elementary changes are the addition and deletion of elements (concepts, properties etc) from the ontology. There is no general consensus on the type and number of composite changes that are necessary. In [30], 12 different composite changes are identified; in [29], 22 such operations are listed; in [32] however, the authors mention that they have identified 120 different interesting composite operations and that the list is still growing! In fact, the number of definable composite operations can only be limited by setting a granularity threshold on the operations considered; if we allow unlimited granularity, we will be able to define more and more operations of coarser and coarser granularity, limited only by our imagination [24].

The third phase of ontology evolution is the *semantics of change* phase, in which possible problems that might be caused in the ontology by the identified changes are determined and resolved; for example, if a concept is removed, we should decide what to do with its instances. The role of the *implementation* phase is to implement the changes identified in the two previous phases, to present the changes to the ontology engineer for final verification and to keep a log of the implemented changes [14]. The *change propagation* phase should ensure that all induced changes will be propagated to the interested parties (agents, ontologies etc). Finally, the *change validation* phase allows the ontology engineer to review the changes and possibly undo them, if desired. This phase may uncover further problems with the ontology, thus initiating new changes that need to be performed to improve the conceptualization; in this case, we need to start over by applying the change capturing phase of a new evolution process, closing the cyclic loop.

3 Discussion on Current Research Directions

Current ontology evolution tools have reached a high level of sophistication; the current state of the art can be found in [14]. While some of these tools are simple ontology editors, others provide more specialized features to the user, like the support for evolution strategies, collaborative edits, change propagation, transactional properties, intuitive graphical interfaces, undo/redo operations etc.

Despite these nice features, the field of ontology evolution is characterized by the lack of adequate formalizations for the various processes involved [7]. Most of the available tools attempt to emulate human behavior, using certain heuristics which are heavily based on the expertise of their developers. They are not theoretically founded and their formal properties remain unspecified. Moreover, they require varying levels of human intervention to work, a rather unrealistic assumption ([7], [20]). In short, current work on ontology evolution resorts to ontology editors or other, more specialized tools whose aim is to *help* users perform the change(s) manually rather than *performing* the change(s) automatically.

We believe that this is not a practical approach to be taken. First of all, the human user that intervenes in the process should be an ontology engineer and have certain knowledge on the domain. Very few people can be both domain and ontology experts. But even for these specialized experts, it is very hard to perform ontology evolution manually [13], [30]. So, it is simply not practical to rely on humans in domains where changes occur often, or where it is difficult, impossible or undesirable for ontology engineers to handle the change themselves (autonomous robots or software agents, time-critical applications etc).

Moreover, different ontology engineers may have different views on how a certain change should be implemented [30]. These views are affected by commonsense knowledge, personal preferences or ideas, subjective opinions on the domain etc. This means that there is no single "correct" way of changing an ontology. Computer-based evolution could (at least) guarantee determinism, objectivity and reproducibility of the results, even though some people may disagree on how a change was implemented. But then, is there a consensus on the effects of a given change even among humans?

Another source of problems for manual ontology evolution is the complexity of modern day ontologies. Complex ontologies are usually developed by several engineers. A change in one part of the ontology might have unintended effects in other parts of the ontology [31]. The person who made the change may be unaware of the full extent of the change's effects, as he doesn't know all the parts of the ontology.

These points uncover the need for automatic ontology evolution; computer-based ontology evolution is not only necessary for many applications, it is also desirable in certain contexts. Human supervision by specialized experts should be highly welcome and encouraged whenever possible; however, the system should be able to work even without it. Human intervention should constitute an optional feature guaranteeing the quality of the evolution process, but should not be a necessary one.

Another problem with current research directions is related to the representation of changes. In tools that are simple ontology editors, there is usually little or no support for any kind of composite changes to the ontology [14]. In more specialized tools for ontology evolution, there is a pre-defined set of elementary and/or composite opera-

tions that are supported, providing a greater flexibility to the user. For each such operation, there is an associated procedure that handles the change as well as the effects of the change (semantics of change phase); this procedure can, in some cases, be parameterized to cover different needs. Unfortunately, there is no guarantee that the provided parameterization is enough to cover any possible need of the knowledge engineer. Unforeseeable needs may require unforeseeable reactions to a given change. Furthermore, there is no limit on the number of composite operations that can be considered and, even if we restrict ourselves to the most common types, there is a large number of them [32]; this makes the process non-scalable. A unifying approach is necessary to cover all cases.

The problem becomes even more complicated due to the fact that not all different types of change are readily available at design-time. New needs may require new operations. For operations that are not in the supported list, the ontology engineer should choose a sequence of two or more simpler (more elementary) operations of different granularity. Unfortunately, such a choice will undoubtedly affect the quality of the change, leading to unforeseeable problems [30]. In addition, it cannot be performed without human participation.

In current approaches, a change request is an explicit statement of the modifications to be performed upon the ontology; however, this request must be determined by the knowledge engineer in response to a more abstract need (e.g., an observation). Thus, current systems do not determine the actual changes to be made upon the ontology when faced with a need for a change; the user should determine them and feed them to the system for implementation. This way, whenever the ontology engineer is faced with a new fact (observation), he decides on his alternatives and selects the "best" one for implementation by the system. This decision is based on his expertise on the subject, not on a formal, step-by-step, exhaustive method of evaluation.

However, to develop a fully automatic ontology evolution algorithm, several issues need to be resolved in a definite, formal manner. For example, how could one track down all the alternative ways to address a given change, using a formal and exhaustive process? How can a computer system decide on the "best" of the different alternatives? Most importantly, what is the definition of "best" in this context? Are there any properties that should be satisfied by a "good" ontology evolution algorithm?

Unfortunately, resolving the above issues in a general manner is not easy using the current research direction because each type of change is treated differently, using a stand-alone, specialized process. Unless a more formal path is taken, the ontology evolution research is doomed to never find answers to these questions.

4 Belief Change and Ontology Evolution

4.1 General Idea, Problems and Opportunities

Our key idea towards resolving the aforementioned deficiencies of current research on ontology evolution is to exploit the extensive research that has been performed in the field of belief change. Belief change deals with the adaptation of a Knowledge Base

(KB) to new information [11]; this fact allows us to view ontology evolution as a special case of the more general problem of belief change. Therefore, it makes sense to apply techniques, methods, tools, ideas and intuitions developed by the belief change community to ontology evolution. Recently, the idea of using results from the belief change literature as an inspiration for ontology evolution research has been independently considered in [21], [25], [28], giving interesting preliminary results.

We believe that our approach allows us to kill several birds with one stone. The mature field of belief change will provide the necessary formalizations that can be used by the immature ontology evolution field. Belief change has always dealt with the automatic adaptation of a KB to new knowledge, without human participation; the ideas and algorithms developed towards this aim will prove helpful in our effort to loosen up the dependency of the ontology evolution process on the knowledge engineer. Finally, previous work on belief change can protect us from potential pitfalls and prevent reinventing the wheel for problems whose counterparts have already been studied in the rich belief change literature, while belief change intuitions that are not directly applicable to ontology evolution may serve as an inspiration for developing solutions to similar problems faced by ontology evolution researchers.

Unfortunately, a direct application of belief change theories to ontology evolution is generally not possible, because most such approaches focus on classical logic, using assumptions that fail for most ontology representation languages like Description Logics (DLs) [2] and OWL [6]; despite that, the intuitions behind the theories are usually independent of the underlying language. In the sequel, we revisit some of the most important concepts that have been considered in the belief change literature under the prism of ontology evolution in order to demonstrate the main tradeoffs and intuitions involved in their migration to the ontology evolution context.

4.2 Belief Change Issues in the Context of Ontology Evolution

One of the major issues involved in belief change is a fundamental philosophical choice regarding the representation of the knowledge, i.e., whether the explicitly represented knowledge serves as a justification for our beliefs (a *belief base* under the *foundational* semantics) or whether it simply forms a manageable representation of an infinite structure (a *belief set* under the *coherence* semantics) [12]. Under the foundational model, there is a clear distinction between knowledge stored explicitly (which can be changed directly) and implicit knowledge (which cannot be changed, but is indirectly affected by changes in the explicit knowledge). Under the coherence model, both explicit and implicit knowledge may be directly modified by the ontology evolution (or belief change) algorithm in an unambiguous manner.

The choice of the viewpoint to employ is very important, greatly affecting the ontology evolution (and belief change) algorithms considered. This choice depends on philosophical intuition, personal preference and on the intended use (application) of the KB (ontology in our context). Therefore, all the arguments, ideas and results discussed in the belief change literature ([12], [16]) are equally applicable here.

As already mentioned, standard ontology evolution approaches are "modification-centered": the fact (observation, experiment etc) that initiated the change is not im-

portant and is not known by the system; the system is fed with the actual modifications that should be physically performed upon the ontology in response to this fact.

On the other hand, the belief change approaches are "fact-centered": a new fact reflects a certain need for change. This fact is directly fed into the system, which is responsible for identifying the actual modifications to perform upon the KB to address the change (new fact) and for performing these modifications automatically.

We propose the use of the latter model for ontology evolution. Of course, the issue of determining the modifications to perform upon the ontology in the face of some abstract new fact is far from trivial, but there are several belief change techniques that could be of use here. This way, we add an extra layer of abstraction to ontology evolution: the changes to be performed upon the ontology are decided by the system, not by the ontology engineer. This allows the ontology engineer to deal with high-level facts only, leaving the low-level modifications that should be performed upon the ontology in response to these facts to be determined by the system.

There are two general scientific approaches towards the determination of these low-level modifications: *postulation* or *explicit construction* [26]. Under the postulation approach one seeks to formulate a number of formal conditions (postulates) that a belief change (or ontology evolution) algorithm should satisfy in the given context. Under the explicit construction approach, one seeks certain explicit algorithms or constructions leading to algorithms. The two approaches are not rivalrous but complementary [26]. Both methods have been used in belief change with very interesting results. On the other hand, current research on ontology evolution uses only the explicit construction method; one interesting side-effect of our approach is that it provides the necessary formalisms for the development of a postulation method.

Another issue is related to the acceptance of the new information. It is usually assumed that the new information is accepted unconditionally, implying a complete reliance to the incoming data, according to the *Principle of Primacy of New Information* [5]. This principle coincides with common intuition, because the new information generally reflects a newer and more accurate view of the domain. In the ontology evolution context however, the distributed and chaotic nature of the Semantic Web implies that data may be obtained by unreliable or untrustworthy sources; thus, it makes sense to apply techniques from *non-prioritized belief change* [15], where the new data may be partially or totally rejected.

Some researchers argue that semantical (rather than syntactical) considerations should be the driving force behind belief change, so the result of a change should be independent of the syntactical representation of the KB or the change (*Principle of Irrelevance of Syntax* [5]). This principle generally fails for foundational belief bases, because logically equivalent bases may be formed using completely different sets of axioms, implying different justifications [17]. In current works of ontology evolution, this principle is usually ignored, as the explicit part (syntax) of the ontology has a major impact on the result.

Inconsistent KBs (under classical logic) exhibit explosive behavior: anything is implied from them. This is clearly an undesirable behavior, so the result of a change should be a consistent KB, according to the *Principle of Consistency Maintenance* [5]. The only thing that remains to be settled is the exact meaning of the term "consistency"; in the belief change literature, the meaning of the word is clear: a KB is

inconsistent iff it implies a proposition that is tautologically false. For ontology change however, the term "consistency" has been used (others would say abused) to denote several different things.

In [7], several uses of the term were presented and a certain terminology was fixed. More specifically, an ontology was termed *inconsistent* iff it exhibits the explosive behavior of classical logic, implying falsehood; it was termed *incoherent* iff it does not satisfy certain pre-defined conditions related to the quality of the conceptualization. Such conditions include the use of unsatisfiable concepts, properties with no predefined range and/or domain and others. We also argued that ontology evolution needs to be concerned only with consistency (just like belief change); coherency is a very important issue, but is more related to the area of ontology design. We also showed, by means of intuitive examples, that attempting to resolve incoherencies during ontology evolution could lead to unnecessary loss of information (see [7]).

Undoubtedly, the most important issue in belief change is the *Principle of Minimal Change*, which states that the new KB should be as "close" as possible to the original KB, being subject to minimal "loss of information". The terms "closeness" and "loss of information" have no single interpretation in the literature. There have been several proposals on metrics that count information loss in different ways, being used in different algorithms or representation results, as well as postulations that capture this principle in different ways. The formal realization of this principle is in the core of each belief change algorithm, determining its properties to a large extent.

The same considerations are true in the ontology evolution context. In this context, the loss of information could be counted in terms of the number and importance of the modifications that need to be performed upon the ontology during the change. Alternatively, the loss of information could be counted in model-theoretic terms (via some kind of distance metric between the models satisfying the original and the modified ontology), through some specially designed distance metric between ontologies or via certain conditions (postulates) that identify acceptable and non-acceptable transitions. The counterparts of each of these approaches have been considered and evaluated in the belief change literature, greatly simplifying our task.

The above considerations form only a partial list of the issues that have been discussed in the belief change literature. This analysis shows that the determination of the change(s) to be made in response to some new data is a complex and multifaceted issue and that several considerations need to be taken into account before choosing the modifications to be made upon a KB. The same considerations hold for any type of knowledge change, including ontology evolution. Unfortunately, in the ontology evolution literature, most of these issues are dealt with implicitly, if at all, with no formal (or informal) justification of the various choices and without exhaustively considering the different alternatives.

Furthermore, it is interesting to note that, in the belief change literature, there is no human involved in the process of change; all related approaches deal with the problem in a fully automatic manner. In fact, to the authors' knowledge, the option of using a human in the loop of belief change was never even considered as an option, despite the complexity of the problem. This fact forms an additional argument in favor of the use of belief change techniques for automatic ontology evolution.

5 Reformulating the Problem of Ontology Evolution

Notice that the above discussion was made without any explicit mentioning of the underlying knowledge representation formalism; this supports our belief that most of the intuitions involved in belief change are transferable to other contexts as well (including ontology evolution). However, the migration of belief change techniques to ontology evolution will ultimately require some formal setting to be based upon. In this section we will provide some definitions that set the formal foundations upon which future research in this direction could be based.

5.1 Description Logics (DLs) and Web Ontology Language (OWL)

Before going into the details of our formalization, we will make a brief introduction to two important families of logics that will be useful for our purposes, namely DLs and OWL. *Description Logics* [2] form a family of knowledge representation languages, heavily used in the Semantic Web [3]. In DLs, *classes* are used to represent basic concepts, *roles* to represent basic binary relations between objects and *individuals* to represent objects. Those primitive notions can be combined using certain *operators* (such as $\neg$, $\sqcap$, $\exists$ etc) to produce more complex *terms*. Finally, *connectives* are used to represent relationships between terms, such as inclusion ($\sqsubseteq$), disjointness (disj) and others. Each such relationship is called an *axiom*. Axioms dealing with classes and roles form the *Tbox*, while axioms dealing with individuals form the *Abox*. The operators and connectives that a certain DL admits determine the type and complexity of the available axioms, which, in turn, determine the expressive power and the reasoning complexity of the DL. Reasoning in DLs is based on standard model-theoretic semantics. For more details on DLs and their semantics, see [2]. In this paper, the term *DL Knowledge Base* (*DL KB*) will refer to a set of general Tbox and/or Abox axioms representing knowledge regarding a domain of interest.

The *Web Ontology Language* [6], known as OWL, is a knowledge representation language that is expected to play an important role in the future of the Semantic Web, as it has become a W3C Recommendation. OWL comes in three flavors (or species), namely OWL Full, OWL DL and OWL Lite, with varying degree of expressive power and reasoning complexity. In OWL, knowledge is represented using an RDF-like syntax. OWL contains several features allowing the representation of complex relationships between classes, roles and objects in a pattern very similar to the one used in DLs; this close relationship was verified in [19], where OWL DL and OWL Lite (with their secondary annotation features removed) were shown equivalent to the DLs SHOIN$^+$(D) and SHIF$^+$(D) respectively. On the other hand, OWL Full provides a more complete integration with RDF, containing features not normally allowed in DLs; furthermore, its inference problem is undecidable [19]. For more details on OWL and the differences between its flavors, refer to [6].

5.2 Representation of Ontologies

Most current ontology evolution algorithms use a graphical representation to visualize the knowledge that is stored in an ontology. This graph-based representation is pervasive in such algorithms, as it affects the decisions on how each change should be implemented. Graphical representations are extremely useful for visualizing the way that the domain conceptualization was implemented in an ontology. They also help novice users and domain experts get acquainted with the field and understand the conceptualization, by hiding much of the semantic and syntactic complexity of the ontology behind intuitive interfaces and simple visual metaphors.

However, such representations are often not expressive enough for certain applications, because some complex facts expressible using DL axioms cannot be easily expressed using a graph [7]. More importantly, they have led ontology evolution research towards a more informal direction, by shifting the relevant research to concepts, roles, individuals and how they are structured in the ontology graph. As a result, most existing work on ontology evolution builds on frame-like or object models [13] and arbitrary axioms are often not considered part of an ontology (little or no attention is paid to them), leading to unnecessary loss of expressive power.

For knowledge engineers and ontology experts, an algebraic representation provides a more concise and formal representation of the conceptualization, has a cleaner semantics and allows easier formal manipulation than the graph-based approach. In fact, a combination of the two approaches usually works best, as it allows us to use the best of both worlds.

Under the algebraic approach, the knowledge of the ontology is stored as a pair <S,A>, where S is the *vocabulary* (or *signature*) containing information on the elements appearing in the ontology (concepts, roles, individuals) and A is a *set of ontological axioms* [20]. The vocabulary may be a single unstructured set containing all the concepts, roles and individuals relevant to the ontology, or it may have some structure denoting, for example, the concept hierarchy; the set of ontological axioms contains an arbitrary number of axioms representing certain facts on these elements.

In this work, we will use a simplification of the algebraic approach, by dropping the signature structure and *representing an ontology as a set of DL axioms* (i.e., a DL KB), under a given, predefined DL. This way, our approach focuses on axioms, following the axiom-centered ontology model [13], ignoring the signature of the ontology. The graphical structure of the ontology can be completely determined by a set of axioms, so our approach provides a more general representation method.

This viewpoint of ontologies facilitates the definition of a common formalism in terms of which both classical logic and ontologies can be described, thus expediting the task of migrating belief change methods (mostly based on classical logic) to ontologies. It is also simpler and more straightforward than the algebraic approach.

The main disadvantage of this model is that we lose the information normally stored in the signature of the ontology. This is not as major a problem as it seems, because most of the information in S can be represented using axioms as well. For example, if S is a poset representing a certain hierarchy between concepts, then the hierarchy information can be expressed in the form of axioms using the subconcept connective of DLs ($\sqsubseteq$). Things become more complicated when one tries to describe

the elements relevant to the ontology, because, in current DLs, there is no way to express the information that a certain element (concept, role or individual) is relevant to the ontology (i.e., it exists in the signature of the ontology). To deal with this problem, we have introduced the *Existence Assertion Operator*, the *Closed Vocabulary Assumption* (CVA) and the *Open Vocabulary Assumption* (OVA) [7], [8].

The existence assertion operator enhances the expressiveness of any given DL by allowing the formation of axioms that express the fact that a certain element is relevant to the conceptualization of the ontology; the formal semantics of this operator are described in detail in [8]. CVA asserts that no element is relevant to the ontology unless its relevance can be deduced by the ontology through the semantics of the existence assertion operator; CVA and the existence assertion operator can be used to express the knowledge originally in the signature structure using axioms. On the other hand, under OVA, all elements are assumed relevant to the ontology and the existence assertion operator is not used. For a detailed account on the existence assertion operator and the differences between CVA and OVA see [7], [8].

5.3 Tarski's Logical Model: The Common Ground

The proposed representation for ontologies was chosen because it allows them to be placed under a very general logical framework in a clean and smooth manner; this framework was introduced by Tarski and defines a logic as a pair $<L,Cn>$, where L is *a set of propositions* of the underlying language and Cn is a function mapping sets of propositions to sets of propositions (*consequence operation*). The intuitive meaning of Cn is that a set $X \subseteq L$ implies exactly the propositions contained in Cn(X). It is assumed that Cn satisfies three intuitive properties (iteration, inclusion, monotony) that allow it to behave in a rational manner. For details see [7], [26].

It can be easily shown that the above framework engulfs most logics used for knowledge representation. In particular, all monotonic DLs and all the formalisms that have been used for belief change are expressible through some $<L,Cn>$ pair. This way, a KB (ontology) is a set $K \subseteq L$ of an underlying logic (DL) $<L,Cn>$. This viewpoint provides the necessary connection between ontologies and belief change.

5.4 Ontology Evolution Operations

As already mentioned, our approach is "fact-centered": each new fact leads to one operation upon the ontology, which is implemented through an automatically generated sequence of modifications. However, not all types of facts have the same semantics. In this respect, the belief change literature shows the way once again: four types of changes have been identified, each having different semantics and being addressed by a different operation (see [7], [22]). More specifically, a fact may denote that something should be added or retracted from the KB; it may also enhance our knowledge regarding a static world, or denote the way in which the real world has changed (dynamic world). Each of these combinations is handled by a different operation, namely *revision*, *contraction*, *update* and *erasure*, as table 1 shows:

Table 1. Operations for Belief Change and Ontology Evolution

Operation	Type of Change (Addition/Retraction)	State of the World (Static/Dynamic)
Revision	Addition	Static
Contraction	Retraction	Static
Update	Addition	Dynamic
Erasure	Retraction	Dynamic

In the same sense, we could define four different operations for ontology evolution. This approach has the advantage of dealing with four operations only (covering all types of changes), thus resolving the scalability problems discussed in section 3. Under this viewpoint, a change involves the identification of the operation (out of the four operations above) as well as the operand of the change (new fact), which are then fed into the system for implementation. But what should be the operand of such an operation? In other words, what constitutes a "change request" in our framework?

In belief change, the change is usually represented using a single proposition; we will slightly generalize this viewpoint by assuming that a change request can be any *set of propositions* (i.e., axioms) of the underlying DL. Our approach is, of course, more general than the standard belief change option; the question is, is this generalization appropriate or necessary for ontology evolution? We argue that the properties of the representation languages commonly used in ontologies (such as DLs) make such an option necessary.

Most belief change approaches assume that the underlying logic contains the usual operators of classical logic (like ∧, ∨ etc) and includes classical tautological implication. Moreover, sets of expressions have conjunctive semantics, so any finite set can be equivalently represented as the conjunction of the set's propositions (i.e., a singular set). The above assumptions fail for DLs and OWL [9], because, in many DLs, the conjunction of axioms is not possible (among other things); thus, in such DLs, there are facts which are expressible by a set of axioms, yet non-expressible by any single axiom. For this reason, we believe it would be unnecessarily restrictive to constraint the change to be a single axiom only, as this does not take full advantage of the expressive power of the underlying DL.

6 An Application: The AGM Theory in Ontologies

The AGM theory of contraction [1] is undoubtedly the most influential work in belief change. For this reason, we chose to apply our ideas to this theory first, and determine whether this particular theory can be applied to ontology evolution. In this section, we provide a short introduction on the generalization of the AGM theory, as well as the main results regarding its applicability in the ontological context. For more details on this work, refer to [8], [9], [10].

6.1 The AGM Theory and its Generalization

Alchourron, Gärdenfors and Makinson (AGM for short), in their seminal paper [1], dealt with revision and contraction, as well as with a trivial operation, *expansion*. The main contribution of their work was the introduction of a set of rationality postulates that should apply to each of revision and contraction. These postulates provided a solid theoretical foundation upon which most subsequent research on the subject was based. Our work has focused on the operation of contraction which, according to AGM, is the most fundamental among the operators considered [1], [11].

AGM used certain assumptions when formulating their theory. One such assumption is that the underlying logic follows Tarski's model; this was the only assumption that was kept during our generalization of the AGM theory [8], [9]. AGM additionally assumed that the logic is closed under the usual operators ($\neg$, $\wedge$ etc) and that the consequence operator includes classical tautological implication, is compact and satisfies the rule of introduction of disjunctions in the premises. Unfortunately, these additional assumptions fail for DLs and OWL [8]. On the other hand, Tarski's more general framework engulfs DLs, as explained above.

Regarding the operation of contraction, AGM assumed that a KB is a set of propositions of the underlying logic (say $K \subseteq L$) which is closed under logical consequence (i.e., $K = Cn(K)$), also called a *theory*. Any single expression $x \in L$ of the logic can be contracted from the KB. The operation of contraction can be formalized as a function mapping the pair (K, x) to a new KB K' (denoted by $K' = K - x$).

As explained above, these restrictions may cause problems in the ontological context; for this reason, we generalized the AGM model by including cases where both operands are sets of expressions of the underlying logic (i.e., $K' = K - X$, for K, $X \subseteq L$). This is in accordance to the framework we described in section 5.

The above assumptions allow any binary operator to be a "contraction" operator, which, of course, should not be the case; for this reason, AGM introduced several restrictions on the result of a contraction operation. First, the result should be a theory itself. As already stated, contraction is an operation that is used to remove knowledge from the KB; thus the result should not contain any previously unknown information. Moreover, contraction is supposed to return a KB such that the contracted expression is no longer believed or implied. Finally, the result should be syntax-independent and should remove as little information from the KB as possible. The above intuitions were formalized in a set of six postulates, the *basic AGM postulates for contraction*; these are omitted due to lack of space, but can be found in [1].

As shown by the above analysis, the intuitions that led to the development of the AGM postulates are independent of the underlying knowledge representation language. On the other hand, the formulation of the AGM postulates themselves depends on the AGM assumptions (see [1]). This problem is typical of the problems encountered during the migration of belief change techniques to the ontology evolution context: the differences on the underlying intuitions are minimal, but the representation languages and formalisms used are quite different. In such cases, it makes sense to recast the theory under question (in this case the AGM theory) in a setting general enough to contain ontology representation languages (like DLs and OWL).

Towards this aim, each AGM postulate was reformulated so as to be applicable to all logics under our framework, while preserving the intuition that led to its definition. The resulting postulates can be found below, where the naming and numbering of each postulate corresponds to the original AGM naming and numbering [9]:

(K−1) Closure: Cn(K−X)=K−X
(K−2) Inclusion: K−X⊆Cn(K)
(K−3) Vacuity: If X⊈Cn(K), then K−X=Cn(K)
(K−4) Success: If X⊈Cn(∅), then X⊈Cn(K−X)
(K−5) Preservation: If Cn(X)=Cn(Y), then K−X=K−Y
(K−6) Recovery: K⊆Cn((K−X)∪X)

It can be easily shown that these postulates are equivalent to the original ones in the presence of the AGM assumptions. Unfortunately, it soon became clear that not all logics in our wide framework can admit a contraction operator that satisfies the (generalized) AGM postulates, unlike the logics satisfying the AGM assumptions.

6.2 AGM-Compliance and Related Results

Following this observation, we defined a logic to be *AGM-compliant* iff a contraction operator that satisfies the generalized AGM postulates can be defined in the given logic. This class of logics was characterized using three different necessary and sufficient conditions based on the notions of *decomposability*, *cuts* and *max-cuts* [9]. These results allow one to determine whether any given logic (in the wide sense of Tarski's model) is AGM-compliant or not; notice that this is true even for logics that are not interesting for the purposes of the Semantic Web or ontology representation.

The above research had several interesting side-effects. Firstly, a certain connection between the AGM theory and the foundational model was uncovered. The AGM theory follows the coherence model and there is a known result from the literature stating that the AGM theory is not suitable for a foundational KB [17]; our results verified that this holds even for the generalized AGM theory. More specifically, the dual notions of *base decomposability* and *base cuts* for the foundational case under a belief base were defined; it was shown that these notions form the basis for two necessary and sufficient conditions under which the AGM theory can be applied in the foundational model. Unfortunately, these conditions are very powerful, being satisfied by only few uninteresting logics (which don't satisfy the AGM assumptions) [9].

Another result follows from the definition of a certain equivalence relation which was shown to preserve AGM-compliance [8]. This relation uncovered a certain connection between the AGM theory and the lattice theory: the class of logics modulo this equivalence relation is isomorphic to the class of complete lattices modulo the standard equivalence relation of lattices. These results combined show that AGM-compliance is a feature that can be solely determined by the structure of the complete lattice that is used to represent the logic under question, allowing us to use the richness of results related to lattice theory in the context of AGM-compliance [8].

Given the theoretical foundations set by this work, we were able to determine the AGM-compliance of many DLs (and OWL), as well as to provide specialized condi-

tions and heuristics allowing one to determine the AGM-compliance of any given DL, including those not covered by our work [10]. These results determine, to a large extent, the applicability of the AGM theory to languages used for ontology representation. In addition, a preliminary study on revision was performed [10].

6.3 Evaluation of AGM-Compliance

There is still a long way to go before fully determining the connection between the AGM theory and ontologies. It should be emphasized that AGM-compliance is a property that simply guarantees the existence of a contraction operator satisfying the (generalized) AGM postulates in the DL under question. The extent to which the richness of results related to the AGM theory [11] can be applied to AGM-compliant DLs still remains undetermined. Also, the connection of AGM-compliance with the operation of revision and the related representation results [11] is still unexplored.

On the other hand, this work indicates that important theories from the belief change literature can be migrated, at least partially, to the world of ontologies. Thus, not only the intuitions of the belief change research can be used in our quest for ontology evolution algorithms; certain theories themselves could also prove helpful. Moreover, the application of the AGM theory in this context showed that our ontological framework is suitable not only for capturing the peculiarities of the ontology representation languages and the needs of the related applications, but also for allowing the application of belief change theories to the problem of ontology evolution.

7 Conclusion and Future Work

This paper introduced a formal, logic-based approach to ontology evolution, which will hopefully provide the necessary formalization to this yet immature [29] field. This approach was based on a reformulation of the problem which allows us to view it as a special case of the more general, and extensively studied, problem of belief change. This way, most of the techniques, ideas, algorithms and intuitions expressed in the belief change field can be migrated to the ontology evolution context.

We argued that our approach will lead to several formal results related to ontology evolution and resolve several weaknesses of the currently used model. Our study did not provide any concrete solutions to the problem; our goal was to provide solid theoretical foundations upon which deeper results can be based, thus paving the road for the development of effective solutions to the problem of ontology evolution.

As an application of the proposed research direction, we evaluated the feasibility of applying the AGM theory of contraction [1], one of the most important belief change theories, to the ontological context. The difficulties encountered during this migration attempt are probably typical of the difficulties that will be encountered during the application of other belief change ideas to ontology evolution.

Our approach uncovered a different viewpoint on the problem of ontology evolution. We have scratched the surface of the relation between this problem and belief change; much more work needs to be done on this issue, both in theoretical and in

practical grounds. The application of specific belief change algorithms or postulations in the context of ontology evolution could prove interesting and uncover useful approaches to this problem. The proposed migration of the AGM theory to the ontology evolution context is not complete either, as only the contraction operator was considered; future work should address the problem of revision as well.

References

1. Alchourron, C., Gärdenfors, P., Makinson, D.: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. Journal of Symbolic Logic 50 (1985) 510-530
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2002)
3. Baader, F., Horrocks, I., Sattler, U.: Description Logics as Ontology Languages for the Semantic Web. In: Hutter, D., Stephan, W. (eds): Festschrift in honor of Jörg Siekmann. Lecture Notes in Artificial Intelligence, Springer-Verlag, (2003)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5) (2001) 34-43
5. Dalal, M.: Investigations Into a Theory of Knowledge Base Revision: Preliminary Report. In Proceedings of the 7th National Conference on Artificial Intelligence (1988) 475-479
6. Dean, D., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuiness, D., Patel-Schneider, P., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (2005) web page: http://www.w3.org/TR/owl-ref/
7. Flouris, G., Plexousakis, D.: Handling Ontology Change: Survey and Proposal for a Future Research Direction. Technical Report FORTH-ICS/TR-362 (2005)
8. Flouris, G., Plexousakis, D., Antoniou, G.: AGM Postulates in Arbitrary Logics: Initial Results and Applications. Technical Report FORTH-ICS/TR-336 (2004)
9. Flouris, G., Plexousakis, D., Antoniou, G.: Generalizing the AGM Postulates: Preliminary Results and Applications. In Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (2004)
10. Flouris, G., Plexousakis, D., Antoniou, G.: On Applying the AGM Theory to DLs and OWL. In Proceedings of the 4th International Semantic Web Conference (2005)
11. Gärdenfors, P.: Belief Revision: An Introduction. In: Gärdenfors, P. (ed): Belief Revision. Cambridge University Press (1992) 1-20
12. Gärdenfors, P.: The Dynamics of Belief Systems: Foundations Versus Coherence Theories. Revue Internationale de Philosophie 44 (1992) 24-46
13. Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. In Proceedings of the 2nd European Semantic Web Conference (2005)
14. Haase, P., Sure, Y.: D3.1.1.b State of the Art on Ontology Evolution (2004) web page: http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/SEKT-D3.1.1.b.pdf
15. Hansson, S.O.: A Survey of Non-prioritized Belief Revision. Erkenntnis 50 (1999)
16. Hansson, S.O.: In Defense of Base Contraction. Synthese 91 (1992) 239-245
17. Hansson, S.O.: Knowledge-level Analysis of Belief Base Operations. Artificial Intelligence 82 (1996) 215-235
18. Heflin, J., Hendler, J., Luke, S.: Coping with Changing Ontologies in a Distributed Environment. In Proceedings of the Workshop on Ontology Management of the 16th National Conference on Artificial Intelligence (1999) 74-79
19. Horrocks, I., Patel-Schneider, P.: Reducing OWL Entailment to Description Logic Satisfiability. Journal of Web Semantics 1(4) (2004) 345-357

20. Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: the State of the Art. Knowledge Engineering Review 18(1) (2003) 1-31
21. Kang, S.H., Lau, S.K.: Ontology Revision Using the Concept of Belief Revision. In Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, part III (2004) 8-15
22. Katsuno, H., Mendelzon, A.O.: On the Difference Between Updating a Knowledge Base and Revising It. Technical Report on Knowledge Representation and Reasoning, University of Toronto, Canada, KRR-TR-90-6 (1990)
23. Klein, M., Fensel, D.: Ontology Versioning on the Semantic Web. In Proceedings of the International Semantic Web Working Symposium (2001) 75-91
24. Klein, M., Noy, N.F.: A Component-Based Framework for Ontology Evolution. In Proceedings of the IJCAI-03 Workshop on Ontologies and Distributed Systems, CEUR-WS, vol. 71 (2003)
25. Lee, K., Meyer, T.: A Classification of Ontology Modification. In Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (2004) 248-258
26. Makinson, D.: How to Give It Up: A Survey of Some Formal Aspects of the Logic of Theory Change. Synthese 62 (1985) 347-363
27. McGuiness, D., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (2000)
28. Meyer, T., Lee, K., Booth, R.: Knowledge Integration for Description Logics. In Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning (2005)
29. Noy, N.F., Klein, M.: Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information Systems 6(4) (2004) 428-440
30. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-driven Ontology Evolution Management. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (2002)
31. Stojanovic, L., Maedche, A., Stojanovic, N., Studer, R.: Ontology Evolution as Reconfiguration-Design Problem Solving. In Proceedings of the 2nd International Conference on Knowledge Capture (2003) 162-171
32. Stuckenschmidt, H., Klein, M.: Integrity and Change in Modular Ontologies. In Proceedings of the 18th International Joint Conference on Artificial Intelligence (2003)