Modeling Intelligent Ontology Evolution Using Biological Evolutionary Processes

Philip O'Brien
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia B3H 3V6
Email: pobrien@cs.dal.ca

Syed Sibte Raza Abidi
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia B3H 3V6
Email: sraza@cs.dal.ca

Abstract— **Ontology design and development is rapidly becoming a study in its own rite. It consists of a set of processes that work together to advance ontologies from simple, static documents to complex and dynamic networks of semantic concepts that describe a domain. This paper discusses these processes with a particular focus on ontology evolution. Ontology evolution has arisen from a demand to track and detect changes in ontologies and the environments in which they live. We review this subset of the ontology development domain and propose an alternate evolutionary model for ontology evolution.**

We propose an alternative model for ontology development that borrows from notions found in natural evolution. Pursuing a coarse that is well-suited for ontology evolution will yield benefits by easing and simplifying the process of ontology evolution.

I. Introduction

When sitting on his stool, on an island nearly 500 miles off the coast of mainland South American, one may wonder if Charles Darwin ever considered how his theory of the origin of species [1] and evolution would effect seemingly unrelated scientific fields of study? Its reasonable to think he did not.

But nearly 150 years later the concept of evolution has been widely used in many areas of scientific research to explain new phenomena and to develop new techniques that hone their scientific underpinnings and motivation in the theory of evolution. In computer science it has been used in the design of genetic algorithms, neural networks, and software engineering. Recently, and more closely related to the topic of this paper, it has been used to describe the dynamic changes demanded of domain ontologies and the applications that use them [2].

Ontology evolution, an emerging research theme in the realm of knowledge management, is the gradual adaptation of an ontological representation, conceptualization and specification of the concepts and properties of a domain in which the ontology lives [3]. It is the enterprise of change management witnessed in new versions of an ontology and the relationships between subsequent versions [4]. It is the cummulative sum of the transitions that take an ontology through the sequential revisions of that ontology and arrive at the latest, or most evolved, version. Change management and detection of required changes in an environment is necessary to provide true representation of the domain an ontology is intended to represent. Ontology evolution has arisen from this need to adapt promptly while supporting the applications that have not yet evolved to coincide with the ontology changes.

Ontological evolution has developed as a consequence of other fundamental work in ontology revisioning, ontology versioning, ontology adaptation, and ending, in its present state-of-the-art, with ontology evolution. This development and history is detailed further in this paper. Klein, Fensel, Kirakov and Ognyanov [3] initially proposed the concept of ontology evolution. Although it was not explicitly mentioned in their 2002 paper, they first addressed the issues of what is currently considered evolution of an ontology.

The rate at which knowledge is created and changed is astounding. Knowledge based systems tend to make the changes yet the ontological specification of the knowledge base and the consistencies of the actual knowledge creation process grow more complex. A model with well known parameters and processes that are traceable is required. We argue that these parameters and processes are best modeled using natural evolution. This paper proposes we model ontology evolution processes based on natural evolutionary paradigms and theories. We describe how we can adapt concepts from known processes and ideas in natural (biological) evolution and use these concepts to guide ontology development. Using these well understood processes, we hope to bring light to the complex and daunting task of engineering ever-evolving ontologies.

II. A Brief Overview of Natural Evolution

Natural evolution theory consists of three main parts:

1) An organism's DNA can change, or mutate: A change may occur because of X-rays, cosmic rays, nuclear radiation, or random chemical reactions in a cell. These changes may affect offspring immediately or after several generations.

2) A change (or mutation) is either harmful, beneficial, or neutral: When a change is harmful it is unlikely the organism will be in contention for advancement of a species and will likely not contribute back to the gene pool. If a change is beneficial it is likely the organism will compete and perform better than other organisms and will reproduce more. Through reproduction the mutation spreads until it is dominant and observed in many samples of a species.

3) Over time, cascading changes in an organism's DNA result in new species being formed: This process is termed *speciation*.

Evolution is not: (1) simple changes in DNA in order to adapt to an environment or (2) localized to one organism. Evolution is the result of reproduction and persistence of a mutation throughout a population.

After mutations and reproduction have occurred, natural selection takes over. The theory of "natural selection" states that those organisms with beneficial mutations will survive in adverse environments where others will not. The surviving organisms will be more likely to reproduce and pass on the beneficial genes to its offspring.

Speciation is a consequence of natural selection and creates new species from a single gene pool. Mutations in DNA propagate over time to possibly produce two species, derived from the same gene pool, that cannot interbreed. Consider putting a group of Saint Bernards on an island and a group of terriers on another. Initially the two breeds could reproduce (probably through artificial insemination). Over time, however, the two islands may witness the creation of new species that cannot interbreed. This observation was made by Charles Darwin [1] in the birds of the Galapagos Islands.

III. Ontology Development

In approaching ontology evolution it is useful to understand evolution's role in the ontology life cycle. Following the literature, it is apparent that it falls under ontology development which itself is an umbrella of a number of other activities. These activities are performed nearly sequentially to produce the ontology life line.

Ontologies are engineered to meet specified requirements. An engineered ontology is designed to meet the needs of a domain and a set of users and designers at the time of creation. This human-driven, error-prone procedure is almost never completely accurate, complete, or capable of sustaining usefulness over the lifetime of the ontology, thus *revisions to that ontology* are necessary to cope with frequent changes in the application environment. Revisions consist of operations that change the state of an ontology; deleting concepts, refactoring properties, resolving logical conflicts. These changes produce a new version of the ontology that may be (partially) inoperable with the previous version. As a result, instances of the ontology, applications that use those instances, and dependant ontologies will have difficulty interfacing with the new version. *Ontology versioning* addresses this problem by maintaining multiple versions of the same ontology. Tools that use the instances or schema are required to produce multiple adapters for versions of interest. Automating this process or providing facilities to keep pace with frequent changes in the way concepts are represented in the domain constitutes *ontology adaptation*. These changes are witnessed immediately and mostly locally by the versions subject to the change. They may or may not persist and be represented in future versions of the ontology. Such changes are instantiated and guided, in part, by human observers using ontology editors. When changes are made to adapt to new requirements that persist over time, producing new versions, the multiple versions of the ontology must be tracked and the changes logged in some way as to produce a mapping between subsequent versions in the evolution chain of the ontology. This practice of maintaining and managing the transformations is termed *ontology evolution*.

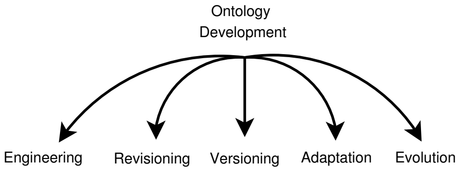

Fig. 1. Ontology development processes. Temporal sequencing flows from left-to-right.

The five processes fall under the same overlying heading of ontology development, as shown in Figure 1. Ontology engineering has been covered in great detail and will be omitted here. The remainder of this section will discuss the remaining four in some detail with an extended look at ontology evolution.

A. Ontology Revisioning

A definition of *ontology* finds its origins with Gruber [5] and deals primarily with the content, formalization, and organization of concepts in a codified ontology. Ontology use has gained wide acceptance in knowledge management systems and hence ontology development has become a research area of its own. It was quickly realized that concrete representations of ontologies were not flexible enough to deal with the many changes observed in the natural world. Thus revisions to ontologies were required. The first work proposing a logical manipulation of ontologies to reflect changes in an environment came from Foo [6]. In his work, Foo proposed that ontology revisioning is the expansion of an ontology through refinement, abstraction, or the addition of concepts, relations, or predicates, among others. When current ontological models are discarded or deemed useless, a revision is made.

Founded on the composition logic of Wiederhold [7], Foo explores the rationalization of ontology revisioning less the predication of its operation. A detailed discussion of operations to revise ontologies, consistency checking and concept validation followed [8]. Farquhar *et al.* presented Ontolingua as a tool to manifest ontology changes and resolve effects of revisioning. This functionality is realized by a set of reusable ontology authoring rules, or operations [8]. Being able to revise ontologies effectively, ontological versioning soon followed.

B. Ontology Versioning

By 2000, revising ontologies was an automated, but often guided, task and the work led to designating versions to growing, expanding ontologies [9].

When a collection of changes are made to an ontology, the result may be incompatible with the original, or logically dissimilar. A series of changes then constitutes designating the new copy of the ontology as a separate version, independent of the previous versions. Versions of an ontology for the same domain may also result when ontologies are independently developed [2] thus requiring a means to identify the different ontologies. Versioning requires some way of differentiating the variants of an ontology. Not only must versions be unique among the many other versions of the same ontology in a single derivation tree, they must be mutually unique in a greater scope; in the past decade the setting for this greater scope has become the World-wide Web as it hosts vast and heterogeneous collections of information.

The difficulty arises if we consider that an ontology is a specification of a conceptualization and that every modification to that specification results in a new conceptualization of the domain. Klein and Fensel take the position that any change to the character representation of an ontology constitutes a revision [2]. If the logical representation of the domain being modeled is changed, the identity of the ontology changes subsequent to that revision and forms a new conceptualization. They leave it to the ontology engineer to decide if a new version is warranted.

Versioning necessitates unique identification of ontologies. An identification scheme that uniquely indicates the identity of an ontology irrespective of its location, use, or of other versions of the ontology is required [2], [3]. Some methods that address this issue leverage Document Object Identifiers (DOI) or Uniform Resource Identifiers (URI) for document identification on the web. The current means of identifying ontologies globally is a comglomerate of hierarchical identification based on a file/ontology/line class system, combined with URIs as well as multi-part version numbers. A complete discussion is given by Klein and Fensel [2].

The need for delimiting the boundaries between conceptualizations of a domain as ontologies change is apparent. Many problems arise when revisions are made in absence of tracking those changes.

Versioning solves many of these issues:

- dependant ontologies remain consistent with their parent ontologies by specifying the version identifier. Other approaches have been pursued to have dependant ontologies dynamically change when the parent undergoes revision. This approach, however, cause conflicts and logical inconsistencies if the revision introduces modifications that do not satisfy some of the semantics of the dependant ontology.
- applications that interface with an ontology version are not required to immediately upgrade when new versions are introduced. Without versioning, it was necessary to make hard-coded changes in the application. Versioning alleviates this obligating the application only to use adapters that interface with interesting versions.
- instances based on an ontology remain valid
- ontology divergence is eased for the reasons given above. This divergence may occur due to irreconcilable differences in terminology, scope, diverging meanings for like terms, and encoding differences [7], [10].

Versioning also provides facilities that allow ontologies to adapt rapidly, if necessary, making ontology adaptation possible [11].

C. *Ontology Adaptation*

Often changes are required of a domain ontology that reflect only the needs of a single application, or in a subset of the instances. Providing for customizable ontologies for local environments without conflicting with the version from which it was adapted is called *ontology adaptation*.

This adaptation is the localized and specific change of a version of an ontology to suit the immediate needs of an actor working with the ontology, or the frequently changing requirements local to that actor (or a set of actors). Adaptations to an ontology are not necessarily returned to the version from which they derive or persist when the next version of the ontology is created. The changes in the adapted ontology are not propagated to other systems and actors. This self-motivated and semi-private change can be seen as a critique of practice or use, by which versions are first adopted and then adapted for a specific application. Feedback is then given, an evaluation of the change is made, and a reassessment is conducted, possibly followed by more adaptations.

A distinguishing feature of adaptation, when considering ontology change and growth, is the necessity of user intervention [12]. Supervision by users is necessary to resolve orphaned concepts and properties, semantically assessing the propagation of changes to parents of revised concepts, validating adaptation in a localized scope, and subjectively determining how lenient a system will be on instance conformity.

As ontologies grow larger, managing the changes and tracking numerous related elements within a domain ontology implies a heavy cognitive load on the maintainer [13]. Tools to lift this burden have been proposed to visualize ontologies and those changes that may result in side effects. These user-centric tools seem to indicate a user-driven approach and characterization of ontology adapation.

Stojanovic *et al.* [14] discuss tools for user-aided ontology adaptation in resolving instance conflicts and logical inconsistencies. In addition, they propose tools for semi-automated (i.e., user interface driven) discovery of changes and ontology requirements causing changes. These tools elicit feedback from users to drive the conflict-resolution functionality and confirm its validity, relative to the user's conceptualization of the domain.

Where adaptations persist in future versions and change management is integrated into the process, ontology adaptation morphs into ontology evolution.

D. *Ontology Evolution*

Noy and Klein argue that the distinction between versioning and evolution does not exist for ontologies as it does for

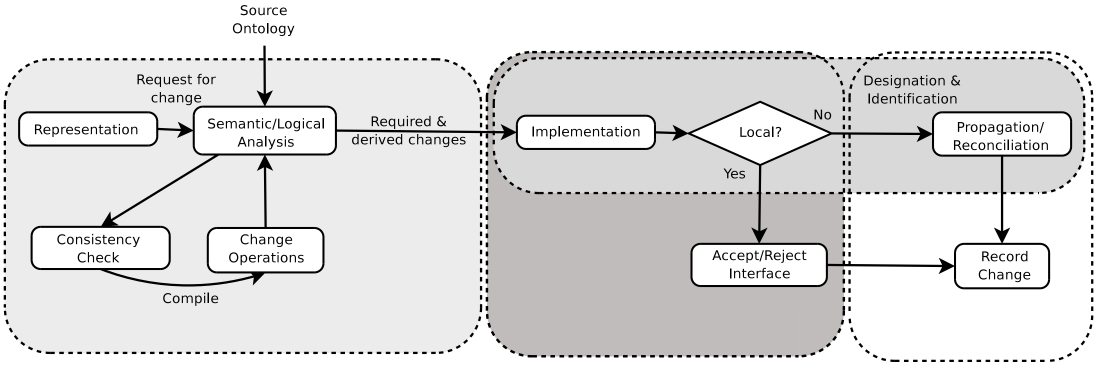

Fig. 2. From ontology revision to ontology evolution

schemas [15], a distinction is made and supported throughout the literature.

Ontology evolution is the cummulative adaptation of an ontology to changing stimuli in an environment, oscillating trends and patterns in the domain of application, the management and dissimination of the changes to the descendants of an ontology [14], and the monitoring of how instances of ontologies are used [16]. Evolutionary ontologies gradually adapt a set of ontological representations, conceptualizations and specifications of the concepts and properties of a domain in which the ontologies live [3].

The predominant aspect of what is currently termed ontology evolution is the management of the changes that take an ontology from one version to another. Over time, multiple versions arise and the evolution process is responsible for being able to take the original ontology all the way through to the most recent revision. Often, a more interesting requirement seeks the source of the ontology derivation—the ontology geneology. Approaches to managing changes and relationships between concepts have been proposed. Some include operation logging [4] and others graph-search techniques [17], while others dismiss the management of changes and instead present a means of migrating the instances of a version to the next version [18].

Underlying these approaches is the need to match, align, prioritize or merge ontologies. Many techniques have solved this problem [7], [10], [19], [20], [21], [22], [23], at least in part, and are indeed considered pivotal in ontology evolution. This success has made management of versions possible. In matching we verify that two ontologies are of the same origin or are compatible and can be aligned. Alignment allows comparison between two ontologies during decision-making or data migration. Prioritization defines a partial ordering of the axiomatic statements of an ontology to avoid ambiguous or inconsistent resolution of knowledge [24] for the dynamic and evolving ontologies. Merging combines two ontologies or knowledge bases to form a single unit representative of the concepts of the originals with conflicts resolved[1]. Ontologies are merged by describing commmon, immutable properties of the ontologies in a mathematical algebra [25]. These four activities have been realized with the aid of the RDF(S) [26] and OWL [27] technologies in the Semantic Web framework.

The characteristics that categorize a process as ontology evolution have been discussed by many authors [3], [14], [16]. Subsequently, the requirements of a system to deal with ontology evolution have been discussed in even more depth [13], [28], [15], [12].

IV. Discussion

Some interesting features of ontology evolution are implicitly located throughout the literature. Firstly, ontologies that undergo significant changes over time result in ontologies that do not resemble the original. This may be a consequence of extreme ontology adaptation and further evolution from adapted ontologies when one is taken from a domain and applied to a related, but individiual domain. Changes that occur before the domain change are inherently different than those of the new domain. As a result, ontology divergence occurs and we witness new non-interoperable ontologies that derive from the same version.

Secondly, revisions to ontologies are not necessarily carried through to future versions, but may if there are significant advantages in doing so. Thus, when an application makes a change and chooses to make the change known to the general community, the benefits of the change may warrant the incorporation of those changes into future versions. The adapted and customized version is used in future versioning and its sibling-versions are lost.

Thirdly, management of changes may deteriorate over time when we consider that user-driven processes may introduce inconsistencies and thus logical conflicts in the ontology. Management of the changes then becomes a futile and fundamentally flawed process, in the extreme cases. If a mapping between versions cannot be made because of arbitrary or subjective decisions made during the adaptation process, the management becomes a detrimental and inaccurate overhead.

The four processes—revisioning, versioning, adaptation, and evolution—are inextricably linked and can be summarized, together in context, as in Figure 2. This figure is a combination

[1] There is much debate about the resolution process and how concepts are merged and discarded. This debate is not discussed in this paper

of those found in the collected works by Stojanovic *et al.* [14], [17], [16] and Noy *et al.* [29].

Here, requested changes made to an ontology cause a revision. A request is made and the semantic and logical outcomes of the change are determined. The requested change is verified against the source ontology to ensure the ontology is not left in an inconsistent state. The operations required to fulfill the change request are then generated. These required and derived operations are used to implement the revision. If the changes are intended to be local, the modified ontology is optionally validated by an actor and the resultant ontology is made available. If the change is not local to an application or instance and if significant changes are made, as determined by the ontology engineer, a new designation is made and the resultant version is given a unique identifier. The new ontology is propagated to the applications, dependant ontologies, and the instances that use it. In either case, the changes can be recorded that result in the evolution of the original version of the ontology to the new version.

A. *Traits of an Ontology Development Process*

From examining the literature some attributes of the four processes arise. These attributes can be used to characterize a branch of ontology development as one of these four processes. The attributes, or traits, are outlined in Table I.

From Table I we again see the nesting of the processes: revisioning within versioning, versioning within adaptation, and adaptation within evolution. This nesting is not perfect as can be seen, but produces a sufficient hierarchy of the processes within ontology development.

B. *Natural Ontology Evolution*

With this rudimentary understanding of natural evolution we now discuss the similarities of natural evolution to what we propose as ontology evolution.

Modern ontologies are large and complex. Enterprise applications witness rapidly changing needs thus the requirements and specifications of ontologies much grow in some way as to remain compatible with these needs. This daughting task of revising ontologies while keeping them consistent may be eased by observing patterns that arise in natural evolution. We argue that observing these patterns and processes will provide insight into an alternative model for ontology evolution. Subsequently, we argue it will provide for intelligent ontology change management, versioning, and alternative merging and alignment techniques that consider the semantics of the concepts being merged and aligned.

An ontology may be seen as a living organism. This organism grows and evolves to consolidate heterogeneous and dynamic data while being forced to be distributed, by its very own nature. We see the semantic web, most notably the ontological aspect of it, and the biological world as having many evident similarities.

DNA may be seen to be metaphorically similar with ontologies. For example, an ontology can be envisioned describing the human genome. Changes and mutations to that ontology may emanate when changes occur in the human genome[2]. Genes are synonomous with concepts and properties of an ontology, and mutations with ontological revisions.

In ontological evolution, new ontologies are created over time as a result of persisting beneficial factors of an ontology. These advantageous factors make the ontology superior to others and thus ontological natural selection gradually replaces the old ontologies with the new. This is akin to the "survival of the fittest".

When ontologies adapt to local environments, as proposed in this paper, the localized version chains emerge. After a time, the ontologies are no longer interoperable but derive from the same ontology. This phenomenon may occur when an ontology is introduced into a new, possibly related, domain. The result is two (or more) *species* of the same ontological ancestor. Like the example of the dogs discussed above, ontological-speciation produces similar but non-interoperable decendants of an ontology.

Rapid changes (analogous to detrimental mutations, in most cases) in the environment that cause rapid changes to the ontology will likely not be sustained and thus not appear in future copies of the ontology.

Grounding the model is the synonomous notion of families of ontologies. We define ancestors as those versions of ontologies from which current ontologies derive. Decendants are those ontologies that stem from an ontology through the process of merging. Merging, in this model, is defined as being like reproduction (either sexually or asexually). A species of an ontology is an identifiable group of ontologies that are capable or "breeeding". Species of an ontology are disjoint and are not interoperable with other species.

An example of where this model fails is in considering how revisions (mutations) in the ontology (DNA) emerge. In natural evolution these mutations appear randomly or as a result of some environmental influence, such as cosmic rays. It is not exactly reasonable or practical to propose that changes to ontologies arise sporatically and randomly. Nor is it rational that the merging of two mutated ontologies could produce ontologies that are superior to orchestrated ontologies, even if the change is locally beneficial. More appropriately, we consider that mutations are stimulated by the demands of applications or of the intervention of an ontology engineer. "Breeding" may arise when two existing ontologies are applied to a domain that abridges the two source domains.

In following natural evolution, we propose that ontological evolution is not the process of managing versions of ontologies and their interrelations but is instead the preservation of beneficial ontology changes over time through the merging and alignment of interoperable ontologies. This process is not facilitated by users nor is it observed to occur in some predefined, linear order. It occurs "naturally" in that it is gradual, multi-faceted (i.e., influenced by many factors of the environment), and is time-dependant.

[2] The timescale we envision for ontological evolution is much shorter than that of natural evolution. The processes, however, share many similarities.

TABLE I

OBSERVED TRAITS OF THE ONTOLOGY DEVELOPMENT PROCESSES

Trait	*Revisioning*	*Versioning*	*Adaptation*	*Evolution*
Change operations	Yes	Yes	Yes	Yes
Consistency checking	Yes	Yes	Yes	No
Unique identifiers for ontology copies	No	Yes	Yes	Yes
Rapid localized change	No	No	Yes	Yes
User intervention	No	No	Yes	?
Dynamic modification to suit changing environment environment	No	No	Yes	Yes
Tracking and managing changes	No	No	No	Yes
Mapping concepts between versions	No	No	No	Yes

V. SUMMARY

Ontology development is an umbrella term over the processes of ontology engineering, revisioning, versioning, adaptation, and evolution. The latter four processes have been reviewed extensively in this paper. We have covered the relationship between the four processes and the distinguishing characteristics of each.

We have proposed that new techniques for ontology evolution can be modeled based on natural evolutionary processes. We have submitted that the similarities between ontology evolution and natural, or biological, evolution are many. Building and evolving ontologies within a framework that models itself after natural evolutionary processes relieves ontology engineers of the complexity and obscurity of the processes that allow ontologies to evolve.

All the potential advantages of natural ontological evolution are not presented here. It will take further work to elaborate how natural ontology evolution can ease the job of ontology engineers and concretize the process of evolution. Our future work will aim for a intelligent system to manifest and simulate the processes of natural ontological evolution and determine how it can best be leveraged in an enterprise setting.

REFERENCES

[1] C. Darwin, *On the Origin of Species*. London, England: John Murray, Albemarle Street, 1859.

[2] M. Klein and D. Fensel, “Ontology versioning on the semantic web,” in *Proceedings of the International Semantic Web Working Symposium*, July 30 – Aug. 1 2001, pp. 75–91.

[3] M. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov, “Ontology versioning and change detection on the web,” in *Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*, October 2002.

[4] M. Klein and N. F. Noy, “A component-based framework for ontology evolution,” in *Proceedings of the IJCAI-2003 Workshop on Ontologies and Distributed Systems*, 2003.

[5] T. Gruber, “A translation approach to portable ontology specification,” *Knowledge Acquisition*, vol. 5, pp. 199–220, 1993.

[6] N. Foo, “Ontology revision,” in *Conceptual Structures; Third International Conference*. Springer-Verlag, 1995, pp. 16–31.

[7] G. Wiederhold, “An algebra for ontology composition,” in *Proceedings of 1994 Monterey Workshop on Formal Methods*, July 1994, pp. 56–62.

[8] A. Farquhar, R. Fikes, and J. Rice, “Tools for assembling modular ontologies in Ontolingua,” in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997, pp. 436–441.

[9] B. Smith and C. Welty, “Ontology: Towards a new synthesis,” in *International Conference on Formal Ontology in Information Systems*. ACM Press, October 2001.

[10] J. Heflin and J. Hendler, “Dynamic ontologies on the web,” in *Proceedings of the Seventeenth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI/MIT Press, 2000, pp. 443–449.

[11] J. F. Roddick, “A survey of schema versioning issues for database systems,” *Information and Software Technology*, vol. 37, no. 7, pp. 383–393, 1996.

[12] L. Stojanovic and B. Motik, “Ontology evolution within ontology editors,” in *Conference on the Evaluation of Ontology-based Tools*, September 2002.

[13] B. Blundell and S. Pettifer, “Graph visualization to aid ontology evolution in Protégé,” in *7th International Protégé Conference*, July 2004.

[14] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic, “User-driven ontology evolution management,” in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. London, UK: Springer-Verlag, 2002, pp. 285–300.

[15] N. F. Noy and M. Klein, “Ontology evolution: Not the same as schema evolution,” *Knowledge and Information Systems*, vol. 5, 2003.

[16] L. Stojanovic, N. Stojanovic, and S. Handschuh, “Evolution of the metadata in the ontology-based knowledge management systems,” in *Experience Management 2002, German Workshop on Experience Management*, March 2002.

[17] L. Stojanovic, A. Maedche, N. Stojanovic, and R. Studer, “Ontology evolution as reconfiguration-design problem solving,” in *Conference on Knowledge Capture*. ACM Press, October 2003.

[18] Z. Zhang, L. Zhang, C. X. Lin, Y. Zhao, and Y. Yu, “Data migration for ontology evolution,” in *2nd International Semantic Web Conference*, 2003.

[19] H. Chalupsky, “OntoMorph: A translation system for symbolic knowledge,” in *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 2000.

[20] M. Klein, “Combining and relating ontologies: An analysis of problems and solutions,” in *Proceedings of the IJCAI-2001 Workshop on Ontologies and Information Sharing*, 2001.

[21] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, “An environment for merging and testing large ontologies,” in *Seventh International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, April 2000, pp. 483–493.

[22] P. Mitra, G. Wiederhold, and M. Kersten, “A graph-oriented model for articulation of ontology interdependencies,” in *Proceedings of the Conference on Extending Database Technology*, 2000.

[23] N. F. Noy and M. A. Musen, “PROMPT: Algorithm and tool for automated ontology merging and alignment,” in *Proceedings of the Seventeenth Conference on Artificial Intelligence*. Menlo Park, CA: AAAI/MIT Press, 2000.

[24] W. L. Guilin Qi and D. H. Glass, “Combining individually inconsistent prioritized knowledge bases,” in *International Workshop on Non-Monotonic Reasoning*, 2004.

[25] S. Benferhat and S. Kaci, “Fusion of possibilistic knowledge bases from a postulate point of view,” *International Journal of Approximate Reasoning*, vol. 33, no. 3, pp. 255–285, 2003.

[26] W3C, “Web resource: http://www.w3.org/rdf/.”

[27] ——, “Web resource: http://www.w3.org/2004/owl/.”

[28] J. Heflin, “OWL web ontology language use cases and requirements; http://www.w3.org/tr/webont-req/,” February 2004.

[29] N. F. Noy, S. Kunnatur, M. Klein, and M. A. Musen, “Tracking changes during ontology evolution,” in *Third International Semantic Web Conference*. Springer-Verlag, November 2004.