Change Management for Metadata Evolution

Diana Maynard[1], Wim Peters[1], Mathieu d'Aquin[2], and Marta Sabou[2]

[1] Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
{diana, wim}@dcs.shef.ac.uk
[2] Knowledge Media Institute, Open University,
Walton Hall, Milton Keynes, UK
{m.daquin, r.m.sabou}@open.ac.uk

Abstract. In order to meet the demands of the Semantic Web, today's ontologies need to be dynamic, networked structures. One important challenge, therefore, is to develop an integrated approach to the evolution process of ontologies and related metadata. Within this context, the specific goal of this work is to capture the evolution of metadata due to changed concepts, relations or metadata in one of the ontologies, and to capture changes to the ontology caused by changes to the metadata. After a short discussion of the nature of metadata, we propose a methodology to capture (1) the evolution of metadata induced by changes to the ontologies, and (2) the evolution of the ontology induced by changes to the underlying metadata. This will lead to the implementation of an approach for evolution of metadata related to ontologies.

1 Introduction

Support for metadata evolution becomes extremely important in a distributed, dynamic environment. Change management should warrant the continuity of data access, i.e. all data previously associated with an older version of an ontology should be accessible and interpretable through the new version. When ontologies evolve, these changes should be propagated to all the information objects that are dependent on them, such as local copies of ontologies and annotated texts, although this may not always be possible in practice, particularly where networked ontologies are concerned. After the detection of changes in conceptual structure between two versions of an ontology, the ontology management system must enable the update of metadata affected by the changes in the ontology, in order to maintain a consistent link between ontology and metadata. This link is defined by a number of attributes, amongst which are the URLs for ontology, annotated text and concept. On the other hand, the system must also be capable of enabling updates to the ontology which may be necessary when the metadata changes: for example, the processing of new documents might show the emergence of new concepts and new relations, reflecting an evolution of the domain itself, or some concepts may become less significant. Similarly, the mismatch between the results of two different annotators in a collaborative annotation environment might require the merging of two subconcepts in an ontology. In these scenarios, the evolution of ontologies should be guided by changes in the metadata, to keep the knowledge they contain up-to-date with respect to the considered domain.

With respect to ontology evolution, we therefore distinguish two types of changes: top-down and bottom-up [12]. Top-down changes are explicit changes in the ontology, to which the metadata needs to be adapted. These will be discussed in Section 3. Bottom-up changes occur when distributed metadata need to be reflected by changes to an ontology, and will be discussed in Section 4. First, however, in Section 2 we take a brief look at semantic metadata, in order to fully understand the role of its creation, existence and need for maintenance.

Finally, we would like to clarify some terminological ambiguity when talking about metadata. This is due to the use of the term in various areas of knowledge engineering. First, there is the notion of metadata as ontology metadata. Ontology metadata provides information about the ontology, e.g. who created it, how many concepts it contains, etc. A proposal exists for a standard description for this type of metadata information, the Ontology Metadata Vocabulary (OMV) [5]. Changes in the ontology may affect the value of these standardised attributes: for instance, the addition of another natural language for the labels will lead to a change in the OMV language attribute. The second notion of metadata stems from the area of natural language processing, and is often called semantic metadata or annotations. This type of metadata concerns concept instantiation in the form of the annotation of textual (or other forms of) data, and therefore contains information about the linguistic content of the ontology. Note that within this work, we are concerned primarily with text rather than other forms of media such as images or videos, so where the question of media is left unspecified, we shall be referring to textual forms of media. In this sense of metadata, data annotation concerns the task of adding semantic metadata to text. In this context it concerns the linking of instances in the text to concepts in the ontology, and potentially also finding relations between such concepts. This is known as semantic metadata creation. It is applied to ontology evolution in the form of bottom-up change discovery and ontology population, which concerns adding instances from the text to concepts in the ontology. In this paper, we will be discussing the dynamics of semantic textual metadata: hereafter when we refer to metadata, unless otherwise stated, we mean semantic metadata.

2 Creation of semantic metadata

The Semantic Web aims to add a machine tractable, repurposable layer to complement the existing web of natural language hypertext. In order to realise this vision, the creation of semantic annotation, the linking of web pages to ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes. An important aspect of the Semantic Web revolution is that it is based largely on human language materials, and in making the shift to the next generation knowledge-based web, human language will remain crucial. In the context of new work on distributed computation, Semantic Web Services go beyond current services by adding ontologies and formal knowledge to support description, discovery, negotiation, mediation and composition. This formal knowledge is often strongly related to informal materials. To make these types of services cost-effective, we need automatic knowledge harvesting from all forms of content that contain natural language text or spoken data.

Semantic annotation is essentially the task of assigning to the entities in the text links to their semantic descriptions. This kind of metadata provides both class and instance information about the entities. Semantic annotations enable many new applications to be performed, such as highlighting, indexing and retrieval, categorisation, generation of more advanced metadata, and a smooth traversal between unstructured text and available relevant knowledge. Semantic annotation can be applied to any kind of text - web pages, regular (non-web-based) documents, text fields in databases, etc. - or even to non-textual forms of data (although, as mentioned earlier, we shall restrict ourselves here to textual content). Furthermore, knowledge acquisition can be performed on the basis of the extraction of more complex dependencies, such as analysis of the relationships between entities, event and situation descriptions, and so on.

Automatic semantic annotation of textual data is generally carried out by means of some kind of ontology-based information extraction (OBIE). While semantic annotation can of course be performed manually, it is a time-consuming and laborious task, which certainly cannot scale to the demands of real world applications on the web. Therefore at the least, a semi-automatic, if not fully automatic, process is required. Ontology-based IE differs from traditional IE in a number of ways. First, it makes uses of a formal ontology rather than a flat lexicon or gazetteer, which may also require reasoning to be carried out. Second, it not only finds the (most specific) type of the extracted entity, but it also identifies it, by linking it to its semantic description in the instance base. This allows entities to be traced across documents and their descriptions to be enriched through the IE process. This more semantic form of IE is therefore a much harder task than the traditional one: see for example [7], which describes the extension of a traditional IE system into one which performs a more semantic extraction, comparing the two tasks, systems and results.

The automatic population of ontologies with instances from the text requires the existence of an ontology and a corpus. From this, an OBIE application identifies instances in the text belonging to concepts in the ontology, and adds these instances to the ontology in the correct location. It is important to note that instances may appear in more than one location in the ontology, because of the multidimensional nature of many ontologies and/or ambiguities which cannot or should not be resolved at this level. For examples of OBIE applications, see for example [9, 2].

3 Top-Down metadata evolution

As metadata creation is an expensive task, it is important that sets of ontology metadata and document annotations are kept in sync with an evolving ontology. As far as possible, we do not want to have to reannotate a whole corpus every time the ontology changes in some way (although in some cases this is inevitable). The evolution of the related metadata has to be synchronised with the evolution of the ontologies for the purpose of preserving instance data and compatibility between ontology versions . Therefore, methods for evolving metadata automatically and in parallel with the networked ontologies are required. In the presence of networked ontologies, this includes the synchronisation of distributed metadata.

Consequently, changes in ontology structure need to be captured by means of evolution operations, and described in some standardised fashion. One solution to this problem is to keep the metadata static and keep track of the (specific) version of the ontology used for the text annotation. In this case, we assume that annotations are stable but contextual, and thereby manage the evolution of the ontology only at the ontological level by means of links between the old and new versions of the ontology, or the link between different ontologies. In this case we could study how the annotations according to one ontology can be imported into a new ontology (linked to the other ontology in a formal way). However, this approach leads to a high level of complexity if there are many versions of the ontology and also makes automatic processing more difficult (for example, using the populated ontology for tasks such as information retrieval and question answering). Furthermore, it is not evident how this could be achieved when material is added to or deleted from the ontology, as links cannot exist to items which are non-existent. So it would only be useful for certain types of change and not as a complete framework for change (assuming that we wish to avoid data loss).

3.1 Related Work

The main candidate for the change capture and description phases is [6], who propose a framework for ontology evolution that integrates all sources of ontology change information. A so-called transformation set encompasses all changes that have occurred between an old and a new version of the ontology. The changes that can occur in OWL ontologies have been gathered into a change typology and made available on the web. The change typology is an ontology of change operations, and covers basic change operations such as `delete superclass`, and complex operations, e.g. `add an entire subtree`. The ontology of basic change operations contains `add` and `delete` operations for each feature of the OWL knowledge model. Complex operations consist of a combination of basic operations. We have evaluated this ontology in the light of the requirements for capturing the dynamics of metadata.

The main element missing from Klein and Noy's typology is the networked nature of ontologies. For example, the addition of a new ontology to the network would require the addition of new instances and possibly other changes such as relational information linking instances to different parts of the network. We do not deal here specifically with network-related changes, preferring to focus in the first instance on the changes that stem from one ontology; networked ontology evolution will be part of future work.

The other difference between Klein and Noy's typology and our proposed framework is that they basically developed the change ontology for their own metamodel of OWL, which was built with different assumptions and design considerations that are partially incompatible with ours, since our work is within the scope of the NeOn project[3], which relies on a particular metamodel for OWL. For example, the references to slots are more concerned with a frame-based model and are not really appropriate here, although many of them can be translated into changes that affect instances.

Finally, we also aim to provide a different kind of categorisation of the changes: for example, distinguishing between changes that stem from or affect the network, ontology, concept or instance, because these may have different causes and different effects. We also need to characterise the actions associated with the changes, rather than just specifying the information loss: for example, addition of a new concept in the ontology may require additional annotation in order to find the relevant metadata (instances); deletion of a concept may involve an automatic deletion of its relevant instances; merging two concepts requires merging of their instances, and so on.

For this purpose we need to establish:

1. which change classes are relevant for metadata evolution;
2. what effect such changes have on the metadata
3. what action should follow these changes.

For instance, if a concept is moved, references to the concept should be found and modified appropriately; if a concept is deleted, annotations referencing the concept should be changed to reference its superclass, etc.

3.2 A framework for capturing metadata evolution

We classify the changes according to those which stem principally from the concept, instance or property level. First we look at the effect on the instances caused by changes to the ontology / concepts. We describe the change and the effect on the data (largely as established by Klein and Noy, though with some differences related to the frame-based vs OWL implementation), and propose the actions that should be taken. Note that our aim here is to attempt to specify the changes and actions that should take place – some of these will most likely occur as a matter of course (such as merging associated instances when their respective classes are merged).

[3] http://www.neon-project.org

It must be borne in mind that the change typology represents an initial framework for capturing changes in ontologies. Its envisaged role is to serve as the basis for further discussion and a more formal specification.

Most changes originating at the property level do not really affect the metadata as such: for example, changing a property means that a new property name will be attached to the instance, but does not affect the instance per se. However, if a new property is added, reannotation may be necessary to acquire new instances. On the other hand, if a property is deleted, the instance should automatically inherit the property of its superclass.

Changes occurring at the concept level are the ones most likely to influence the metadata: in some cases, the instances may have to be moved to new classes; in other cases, reannotation may be required in order to acquire new instances (e.g. when new classes are added to the ontology).

Thus we see that for each change, a set of 3 possible actions exists:

1. do nothing;
2. some manual action is required;
3. some automatic action is required.

Do Nothing: The first action is self-explanatory and requires no further discussion. In some cases, a degree of data loss is inevitable.

Manual Action: The second action almost always requires some kind of reannotation of the corpus. This can be for several reasons:

First, existing information in the ontology needs to be reclassified. This could be the case where a new subclass is added. Imagine we have an ontology which contains the class `comestible` and the subclass `food`. Now imagine we add a new subclass of `comestible`, `drink`. Before this subclass was added, all instances of `drink` that were not also instances of `food` would have been classified simply as `comestible`, because that was the most specific class to which they could belong in the ontology. Once `drink` is added to the ontology, such instances need reclassifying under the `drink` class. However, this is almost impossible to do automatically because we have no way of knowing which instances of `comestible` should be moved and which should not, unless we return to the text for further analysis, or unless the ontology provides us with further information.

Second, information may be missing from the ontology. This could be the case when a new top class is added to the ontology.

Finally, it may also be a combination of the two factors. For example, in the case where a new superclass is added to the ontology, some information may be missing and some may need reclassification.

While missing instances can simply be added to the appropriate place in the ontology when found in the text, reclassification is a little more tricky because it consists of a two-stage process: first the system must find the instances in the text and recognise that they are currently not classified in the most appropriate way in the ontology, and second, it must follow the automatic procedure for reclassification as specified below.

Automatic Action: The third action requires a set of procedures to be followed for automating the reclassification of instances in the ontology. Below we give an example of some such possible procedures according to the GATE ontology API: naturally the exact procedures will be implementation-specific according to the ontology model used. Note that some of the actions could be defined simply as natural consequences rather than explicit actions.

- When a class is added
  - It will automatically inherit from its superclasses a set of properties
- When an equivalent class is added
  - It will inherit all the instances from its equivalent class
  - These instances will all have a sameIndividualAs statement added
- When an instance is added
  - It will automatically inherit from its superclasses a set of properties.
- When a class is deleted
  - A list of all its superclasses is obtained. For each class in this list, a list of its subclasses is obtained and the deleted class is removed from it.
  - All subclasses of the deleted class are moved to subclasses of the parent of the deleted superclass. A list of all its disjoint classes is obtained. For each subclass in this list, a list of its disjoint classes is obtained and the deleted class is removed from it.
  - All instances of the deleted class are moved to its direct superclass in the ontology.

4 Bottom-Up ontology evolution

Textual resources and the associated metadata generally evolve faster than the related ontologies. In that sense, they reflect more accurately the evolution of the domain itself, e.g. progresses realised in scientific fields, new trends or obsolete elements. Even if ontologies are supposed to be a stable conceptualisation of the domain, the evolution of the metadata has to be reflected in the related ontologies, to keep the relation between ontologies and metadata up-to-date, and so that the ontologies evolve in accordance with the current state of the domain they represent. Therefore, the dynamics of metadata have to be captured in a way that it can be related to the adequate ontologies and guide their evolution by suggesting corresponding ontology changes, leading to a metadata-driven maintenance of ontologies. We shall therefore investigate which operations are necessary to cover a number of evolution strategies for bottom-up change discovery, e.g. in the case of a required extension or refinement of the ontological structure of a particular ontology component, or the suggestion of the merge of two classes for a particular application if an automatic classifier is unable to distinguish between the two (or if human agreement is not reached) and the concepts in question are considered relatively similar.

In this section, we consider the notion of metadata according to a general definition: information about the content of a resource or a document. Resources and documents can take different forms, (texts, images, etc.), and the associated metadata can be either manual annotations (e.g. folksonomy tags given by the author of an image) or automatically extracted (e.g. using tools for information extraction from texts). In the case of semantic annotations, metadata is represented according to ontology elements. The goal of this section is to show how ontologies can evolve in accordance with the evolution of metadata associated with documents of the domain. The basis of the proposed mechanism consists of relating evolving (non-semantic) metadata with the considered ontologies, in order to assess the insufficiencies of these ontologies in representing the considered metadata as semantic annotations. The process of suggesting changes in an ontology based on changes in the underlying annotated dataset was defined as data-driven change in [8]. It can be argued that the methodology described here is data-driven rather than metadata-driven, in the sense that, in most of the cases, metadata evolves as a consequence of the evolution of the underlying data. However, in the case of manual annotation, for example, the annotator may change the metadata associated with a document, without changing the document itself.

The proposed methodology can be considered as a bottom-up approach for two reasons. First, the basic principle is that changes in metadata would guide the evolution of the ontologies. Second, the definition of the methodology itself is based on experimentations using real life datasets: by studying the evolution of these datasets through the associated metadata, we aim at defining general principles for a metadata-driven ontology evolution. Therefore, we plan to apply this methodology to two concrete case studies using two different datasets and two different forms of metadata: the evolution of ontologies depending on changes in folksonomies and depending on changes in textual documents in a given domain. We first give a brief outline of the methodology we plan to follow, followed by descriptions of two case studies.

4.1 Methodology

This section presents our approach for studying changes in metadata and deriving suggestions of changes in the corresponding ontologies. More precisely, we describe a method to capture (1) the evolution of different types of metadata and (2) the implied changes in the related (networked) ontologies. The proposed methodology relies on a bottom-up approach for capturing the relations between metadata evolution and ontology changes. First, the data contained in a set of existing documents/resources has to be automatically related to the considered ontologies, in order to form an exploitable corpus of ontology-based metadata. Metadata changes can then be captured on the basis of the generated structure, and the study of the implications of these changes on ontologies can be carried out, with the aim of deriving suggestions of corresponding ontology evolutions. We therefore aim to establish what effect the evolution of metadata has on the ontologies, in terms of ontology changes. For instance, if a new prominent term appears from a set of textual resources, a concept should be added in the ontology; if two terms appear to be related in the metadata (e.g. through frequent co-occurrence) then the ontology should be extended with a new relation.

Obtaining this exploitable corpus of metadata, linking the resources to the ontologies, leads to several common, generic or domain-specific issues:

Pre-processing: First, the considered documents, resources and metadata can take different forms, and may contain noise, redundancies, etc. A preprocessing step is generally required in order to obtain an exploitable corpus, including domain (or data)-dependent tasks such as filtering, transformation, etc.

Conceptual Organisation: Once a set of descriptive terms is obtained for each of the considered documents, we need to identify from among these terms those that represent important domain concepts, which may be contained in the ontologies. More importantly, potential relations between these terms, or more precisely between the corresponding concepts, have to be detected. One possibility is to group (or cluster) the terms according to their relations in the documents, suggesting in this way potential relations in the ontologies.

Ontology Matching: Finally, the links between the obtained conceptual structure and the considered ontologies have to be established. This can be realised by using generic ontology matching techniques [1], mapping the organised terms to ontology concepts, and relating them according to ontology relations. It is worth noting that these mappings already provide indications of missing knowledge in the ontologies: the terms and relations for which there is no mapping suggest missing concepts and relations.

These three tasks provide the basic structure which we rely on for capturing metadata changes and the related ontology evolutions.

4.2 Capturing metadata changes

The evolution of the metadata can be captured through the changes occurring in the corresponding set of terms, conceptual structures, and mappings. Documents can be added or removed, leading to a different (either extended or reduced) set of terms. The comparison of the results of the conceptual organisation of the terms obtained at different times allows the consideration of changes from a more abstract (conceptual) point of view, indicating for example previously unknown relations between terms, or ones that have become obsolete. It is worth noticing here that these conceptual structures are not ontologies, but rather intermediary descriptions of the metadata, used to guide the integration (matching) with the considered ontologies, and facilitating the study of metadata evolution. Finally, new mappings to the ontologies may appear from the conceptual structure, and some may be modified or reconsidered. Basically, by comparing the processed metadata at different levels and different time points, we can trace, capture and study these evolutions in an abstract, ontology-related representation. The outcome here is a characterisation of the changes in the metadata that may lead to particular evolution of ontologies.

4.3 Suggesting ontology changes

Capturing the evolution of the metadata is only the first part of the problem. Each evolution may lead to an extension of the existing gap between the metadata and the ontology coverage. For example, if some newly added terms have no correspondence in the ontologies, an extension with additional concepts and relations may be required. On the other hand, if the set of terms generated at any moment in time is smaller than the set of terms generated previously (e.g., because some data has been deleted or because some terms are less significantly represented) then this can lead to the pruning of the ontologies in such a way that they do not contain obsolete concepts/relations. Our goal is to provide general principles for suggesting changes in ontologies in order to fill this additional gap. More precisely, studying the evolution of real-life datasets would allow us to associate suggestions of changes in the ontology to typical changes in the captured metadata, thus providing a basis for guiding the maintenance of ontologies according to the related metadata.

The proposed methodology for the bottom-up, metadata-driven evolution of ontologies is summarized in Figure 1. It is illustrated in the next two sections (4.4 and 4.5) by the application to two real life case studies.

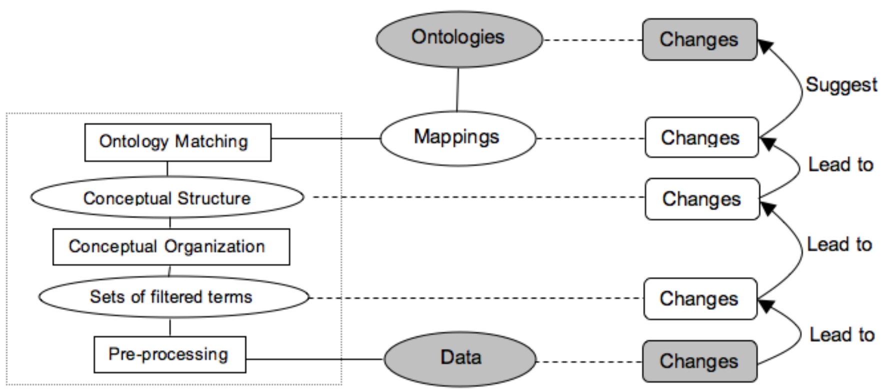

Fig. 1. Schematic view of the bottom-up approach for ontology evolution

4.4 Case Study 1: Evolving ontologies through linking them to folksonomies

Social tagging systems such as Flickr (http://www.flickr.com), for photo-sharing, and Del.icio.us (http://del.icio.us), for social bookmarking, are becoming more and more popular, nowadays covering a wide range of resources and communities, with a huge number of participants sharing and tagging a large number of resources. Due to their popularity, folksonomies are changing rapidly as users add new resources and tags. Because they are updated continuously, folksonomies are up-to-date with respect to the vocabulary used by a wide range of people, thus reflecting new terms that appear. Unlike folksonomies, ontologies are built at a much slower rate and therefore they often lag behind the novel terminology in a given domain. A solution for automatically enriching (and hence evolving) ontologies is to align them to folksonomies and modify them so that they reflect the changes in the folksonomies. This alignment is also beneficial for folksonomies. Current tag sets lack any semantic relations and therefore are hard to use during searching. The alignment to ontologies allows the enrichment of folksonomies with semantic relations so that more semantic searches can be performed.

In this first case study, we investigate how ontology changes can be derived from changes in folksonomies. The process of linking folksonomies to ontologies has been described in [11] and will be used in the first part of the case study (as it covers the first three major steps of our methodology). We investigate the tag sets in Flickr and Del.icio.us, due both to their popularity (with a large number of resources, users, and tags) and availability. In our experiments, we use the same Del.ici.ous tags as [11] and Flickr tags for photos posted between 01-02-2004 and 01-03-2006.

Having derived the metadata following the work of [11], we need to capture the way metadata changes. Our plan is to re-run the process described in their work at different points in time and compare the output (i.e., see how the obtained clusters differ). We will identify clusters that have been added or which have disappeared. At a more fine-grained level, for each cluster we will monitor internal changes, namely additions or elimination of some tags. All these types of changes have important influences on the evolution of the ontology. Our task will be to derive a wide range of metadata changes taking into account the metadata derived in our experiments.

Once a set of changes in the metadata has been identified, these changes can be used to suggest corresponding updates to the ontologies. While this typology of changes will be derived when the experiments are run and therefore will be grounded in actual data, we can already predict some typical changes. For example, if a new cluster is added, then find an ontology that contains elements (concepts/relations) with the same label as the tags in the cluster. If the ontology only covers a subset of the terms, then find methods to extend the ontology with concepts corresponding to the missing tags. If a new tag is added to a cluster, then if no corresponding concept exists in the ontology to which the cluster was aligned, insert the tag in the ontology.

4.5 Case Study 2: Data Driven Ontology Learning on FAO data

The second case study is performed on a different type of data set from the previous case study. In this case, we investigate how ontologies can be evolved through applying ontology learning methods to textual data. The data set used is a collection of textual data from the FAO[4] (United Nations Food and Agriculture Organisation, Fisheries Department). There are several collections, including news items, fact sheets on species and internal documentation.

We envisage that pre-processing will be performed using existing software packages. In particular, we will experiment with three tools: TermExtractor [10] – a web service for the extraction of domain relevant terminology; GATE [4] – a suite of natural language applications for document annotation; Text2Onto [3] – an ontology learning tool that provides a variety of algorithms needed for the entire process of ontology learning. Some of these algorithms deal with the simplest task in ontology learning, that of extracting relevant terms from a corpus.

The next step is one of conceptual organisation, which aims at deriving some meaningful structure between the identified terms. Again, we envisage the possibility to use two different kinds of packages to perform this task: clustering algorithms for when TermExtractor is used, leading to clusters similar to those obtained in case study 1; and the ontology learning algorithms from Text2Onto to derive an ontological structure between the identified terms (including subsumption relations, mereological relations, general relations).

Depending on the conceptual structure derived previously, we can use a variety of ontology matching approaches to align these structures to ontologies. Simple, string-similarity based methods can be used to find correspondences between terms in a cluster and the labels of ontology elements. If the derived structures are richer than sets of terms, we can also employ structural matching methods to find alignments between the derived ontologies and the ontologies (ontology networks) that need to be updated.

Capturing metadata changes and linking them to updates in the corresponding ontologies will be the topic of our investigation after running the above mentioned experiments. We can already say, however, that these types of changes will somehow depend on the kind of conceptual structure that the metadata takes. If we derive clusters of terms, then we can provide a similar typology of changes as in case study 1. If the conceptual structures derived are ontologies, then we can re-use and extend Klein's typology of changes to capture changes between ontologies derived at different moments in time and then suggest adequate changes in the ontologies.

5 Conclusion and Future Directions

In this paper we have described a framework and methodology to capture the dynamics of metadata, consisting of two main aspects: the evolution of metadata when an ontology changes, and changes to the ontology resulting from metadata evolution. The first aspect involves using the Klein and Noy ontology as a starting point, and proposing changes in order to make it more suitable for our needs. We then discuss how this will be implemented. For the second aspect, we have

[4] http://www.fao.org

described a methodology for capturing changes in metadata with the aim of suggesting changes in the related ontologies. We adopt a bottom-up approach, studying the evolution of real-life datasets to come up with general principles for metadata-driven ontology evolution. In this line of approach, we propose to apply this methodology to two concrete case studies, using two different datasets, and two different forms of metadata: tags of folksonomies, and texts related to the agricultural domain. Therefore, the obvious next step of this work is to run the experiments corresponding to these case studies, capturing the evolution of the metadata by relating the considered resources to the considered ontologies at different points in time. We believe that the study of these ontology-based metadata evolutions will allow us to understand and potentially capture the implications of metadata changes on the related ontologies. The concrete outcome of this work should be general rules for suggesting ontology changes to reflect metadata changes, providing a basis for a metadata-driven ontology evolution.

Since metadata changes lead to ontology changes, and ontology changes lead to metadata changes, the interactions between these two processes have to be considered. We can therefore imagine an integrated process that would alternatively suggest changes in ontologies and metadata until a stable version of both elements is obtained.

6 Acknowledgements

This work was partially supported by the EU project NeOn (IST-2005-027595).

References

1. J. Banerjee, W. Kim, H. Kim, and H. F. Korth. Semantics and implementation of schema evolution in object-oriented databases. In *Proceedings of SIGMOD*, 1987.
2. K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10(3/4):349—373, 2004.
3. P. Cimiano and J. Voelker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, Alicante, Spain, 2005.
4. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
5. J. Hartmann, Y. Sure, P. Haase, and M. del Carmen Suarez-Figueroa. OMV – Ontology Metadata Vocabulary. In *ISWC 2005*, 2005.
6. M. Klein and N. F. Noy. A Component-Based Framework for Ontology Evolution. In *Workshop on Ontologies and Distributed Systems, IJCAI*, 2003.
7. D. Maynard, K. Bontcheva, and H. Cunningham. Towards a semantic extraction of named entities. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Proceedings of Recent Advances in Natural Language Processing (RANLP'03)*, pages 255–261, Borovets, Bulgaria, Sep 2003. `http://gate.ac.uk/sale/ranlp03/ranlp03.pdf`.

8. D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON)*, Edinburgh, Scotland, 2006.
9. D. Maynard, M. Yankova, A. Kourakis, and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch. In *ESWC Workshop "End User Apects of the Semantic Web")*, Heraklion, Crete, 2005.
10. R. Navigli and P. Velardi. Learning Domain Ontologies from Document Warehouses and Dedicated Websites. *Computational Linguistics*, 4(2), 2004.
11. L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In *ESWC*, Innsbruck, Austria, 2007.
12. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 285–300, Siguenza, Spain, 2002.