# Semantic History: Towards Modeling and Publishing Changes of Online Semantic Data

Jie Bao, Li Ding, and Deborah L. McGuinness

Tetherless World Constellation
Rensselaer Polytechnic Institute
Troy, NY, 12180-3590, USA
{baojie,dingl, dlm}@cs.rpi.edu

**Abstract.** Effective revision tracking is important to maintain and use Semantic Web data for both publishers and readers. Information related to revisions in this setting often contains basic context information, semantic difference summary, and rationale summary. In this work, we present a general architecture for modeling and publishing revision history of social semantic Web data. This model has been implemented as an extension to an existing infrastructure, namely Semantic MediaWiki. We show a variety of applications that can be built using the framework, including provenance tracking, statistics, temporal reasoning and explanation.

## 1 Introduction

The Social Web enables collaborative content publishing on the Web. When enhanced with Semantic Web elements, e.g., RDFa or embedded/synchronized RDF, the conventional text based Web pages are enriched into Social Semantic Web pages with additional semantic annotations and links to other Web data.

For maintaining and using such social semantic Web pages, effective revision tracking is important for publishers as well as readers. For instance, when a page is maintained by a number of editors collaboratively, an editor often would like to get informed of the recent editing history without having to read through the entire page manually. For another example, upon revisiting a published page, a reader may be interested in knowing whether the page has been changed, how frequently the page has been updated, whether a known error in the page has been cleared, and why the change has been made. Indeed, the value of revision tracking has already been demonstrated through the wide adoption of version control systems in software development and content management systems, e.g., Subversion(SVN), Concurrent Versions System (CVS), and wikis. In tracking history, there are several aspects that we should consider:

– **Reusable history**. History data recorded structurally in conventional version control systems only allows users to access the history data via restricted data access user interfaces. This may be improved by publishing these data on the Semantic Web, thus enabling other applications to reuse these data. For example, one may use the Google Visualization API[1] to compare the editing frequency of editors of a wiki page.

[1] http://code.google.com/apis/visualization/

- **Linked history**. Many elements of history change data may be linked to other Semantic Web data. For example, we may link the editor of a page's revision to a person from a social network, and then answer a query such as "find revisions on my publication page made by a friend of mine".
- **Fine-grained history**. We often need fine-grained history information because the difference between semantic annotations of different revisions can be computed at different levels of desired granularity. For example, one may need to model history about a page or a set of pages, instead of only recording history at the RDF triple level. This would allow us to link and query the page data and history data at the same time. For example, we can query for changes to the affiliation of a person (in page data) in the last month (in history data).
- **User-oriented history**. In addition to revision history annotations, we may add more annotations to better serve end users. Instead of using the history for just version control, we expect the history to be used by end users more frequently. Therefore, the history should be extended to incorporate end-user oriented annotations. For example, automatically generated summaries of revisions will help users grasp what semantic annotations have been updated on a page.

In this work, we present a generic model for semantic history representation, focusing on the practical infrastructure for making semantic history publishable on the social semantic Web and showing the value of semantic history through working demos and deployed applications. Our contributions hence include:

- A general framework for modeling revision history of online Semantic Web data, capturing both temporal changes to the semantic data and annotations to actions that led to the change. (Section 2)
- We demonstrate with a Semantic Media-Wiki-based implementation how to automatically capture revision changes in social semantic Web applications. (Section 3).
- We present several typical usage scenarios of our revision model, including provenance tracking, statistics and visualization, temporal reasoning and explanation, showing the strong modeling ability and practical value of our model. (Section 4)

## 2 From History to Semantic History

Semantic history is not merely a collection of encodings of revision data. It also links revision data as a part of the Semantic Web data cloud and further exposes the currently hidden potential of history data to more end users. Figure 1 illustrates a model of semantic history that highlights its key components. An application that adopts this model is required to first *create* (e.g., by monitoring user actions or comparing revisions) history descriptions for Web entities in a structured way, such as generating some encoding of revision data with links to other relevant Semantic Web data. Second, it needs to *publish* semantic history data in a user friendly form, e.g., HTML, RSS or visualization, or in a format that facilitates easy machine processing like RDF.

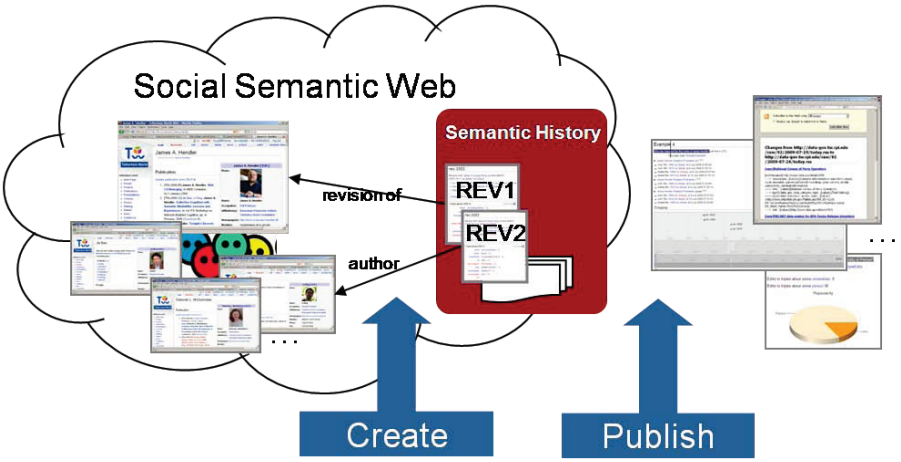


**Fig. 1.** A General Application Model of Semantic History

A published Web page (with URL), potentially with some semantic annotations, may change over its lifetime. A change history description includes (among other information) two main groups of information: revision action descriptions and temporal data descriptions.

**Revision action description**. An action description records information of the event that causes the revision, including the following information:

- Basic context annotation: examples include the subject of the change, the revisions identification, author, timestamp, and user provided revision notes. These basic annotations are used to capture provenance information about the revision, e.g., who generated the identified revision using which online data set at what time. Most of the annotations can be automatically captured by a version control system except for the human input revision notes.
- Revision summary: this gives a summary of resulting changes that may help users comprehension, potentially at different levels of granularity, e.g., at document level and at the instance level.
- Rationale: this provides additional explanation about the action capturing the motivation. For example, it may contain user contributed annotations about the nature of the revision, and sources supporting the revision. It may also contain additional facts, either manually input or automatically generated, that may help identify the cause and impact of the revision action, e.g., syntactical or structural descriptions of data involved.

**Temporal data description**. It describes the modification of semantic data (e.g., RDF triples) associated with the revision. It enables users to retrieve prior versions of the semantic data and compare any two prior versions. Our framework does not specify a particular way for representing temporal data, and allows an application to choose from different representations, e.g., Temporal RDF [7] or RSS feed. We also

independently introduced with some practically interesting alternatives which are given in the following sections.

We have implemented this generic application model of semantic history on the Semantic MediaWiki platform, which will be introduced in the next section.

## 3 Semantic MediaWiki Based Deployment

In this section, we show how the basic model of semantic history we just introduced is applied in modeling revision history in Semantic Mediawiki. A demonstration site of its implementation is at `http://tw.rpi.edu/semhis`.

Semantic MediaWiki (SMW) is a semantic wiki system that allows collective authoring of a shared repository of both semantic and non-semantic data. It is an extension of the popular Mediawiki (MW) platform - which powers Wikipedia - thus it also inherits the built-in change management mechanism of MW. These include[2]:

- Page history: for each page, every revision of the page is stored along with information about author, time and size of the revision; the difference between two revisions of a page can be computed.
- Change summary: when a user submits a new edit, the user can input a short explanation in natural language to summarize the change.
- Recent changes: the page "Special:RecentChanges" shows all changes that happened in a selected recent time span; the display can be filtered by page namespace, type of users (e.g., anonymous users or bots), and nature of the change (e.g. minor edits).
- Action logs: the page "Special:Log" supports a limited search interface over all changes that happened in the wiki, e.g., by page title and by time period.

The MW change management mechanism is limited in several ways when being used with SMW.

1. MW revision logs only record changes to a *page*, but in SMW we often need fine-grained information about addition and deletion of *semantic annotations* on a page.
2. The user submitted change summary is for human consumption only and its meaning is not formally captured; thus, it lacks a built-in automated search or query revisions based on the change summary. For instance, one may want to find all revisions about a page that involve only editorial changes (e.g., typo fixing) but not factual changes.
3. Querying revisions is limited and cannot utilize knowledge (e.g., classification of pages) in the wiki. For instance, one may wish to query about changes about a logic topic made by users who are computer scientists. Another example is to ask about the list of countries on Wikipedia with missing GDP figures or whose GDP figures have not been changed in the past year.
4. Facts that are time-sensitive may be buried in revision history and thus cannot be easily used. For example, one may want to ask for the set of pages that belong to the category "Living people" on Wikipedia as of Jan. 1st 2007.

[2] In this paper, we study MediaWiki 1.15.0 and Semantic MediaWiki 1.4.3.

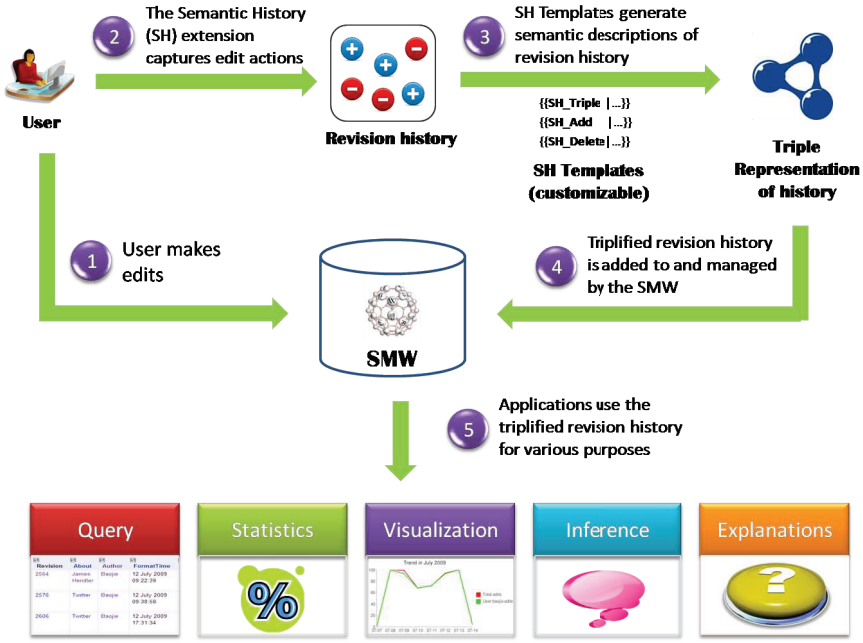


**Fig. 2.** Workflow of the Semantic History Extension of SMW

To address these issues, we developed an extension to SMW called "Semantic History" that can automatically capture revision history of wiki pages, which itself is stored as semantic data in the wiki. In the following we give the details about the extension.

Figure 2 describes the general workflow of the the Semantic History (SH) extension for SMW. It contains the following step in modeling and utilizing revision information in a SMW:

1. A user (potentially a software agent) makes some edit actions to a wiki page. Typical edit actions include creating new pages, modifying a existing page, moving a page and deleting/undeleting a page. The SH extension is transparent to the user, thus it does not require the user any special actions, nor breaks the usual workflow of wiki editing.
2. The SH extension monitors user actions and captures various types of revision changes happened on the wiki. At this step, the SH extension will

   – Create a revision instance that contains basic context information of the revision, e.g., revision id, author, timestamp, page location and editing summary (in plain text). This revision instance will be stored as a "hidden" wiki page. For example, an editing revision with id "1234" may be stored as a page "rev:1234", where "rev:" is a name space for all revision pages and can be protected according to the wiki's access control policy.

- Compare triple changes before and after the editing, identify all triple additions and deletions during the process. Each triple is identified by a unique id and stored as a "hidden" wiki page along with its addition and deletion history.

The output of this step thus is a set of wiki pages (i.e., revision pages and triple pages), marked up using a set of templates (details given below). We deliberately separate the template-based storage of revisions and the triple representation of revisions (which is generated in the next step) for the benefits of customization and extensibility.

3. Generate triple representation for the revision history via a set of predefined templates. For example, the template "SH_Triple" will preform a reification of a triple into three triples, such as `{{SH_Triple|s|p|o}}` on a page "triple:123" will generate :
   - `triple:123 subject   s`
   - `triple:123 predicate p`
   - `triple:123 object    o`

   Other types of information, such as timestamp of a revision or a deletion revision associated with a triple, will be triplified in a similar fashion.
4. Add the triplified revision history back to the triple store of the SMW using SH templates.
5. Enable applications to query the SMW triple store about the semantic representation of the revision history for various purposes, e.g., statistics, visualization, temporal inference and explanation. Details of some typical use cases are discussed in the next section.

Several issues in the workflow need further explanations.

**Using SMW for Data Storage.** One design choice is how to store the revision history data. The SMW triple store is actually created on the top of a relational database. The SH extension, instead of directly storing data as additional tables in the relational database, relies on SMW as a meta layer for data storage. A set of templates are used as a meta model of revision history. They include:

- Templates for revisions: "SH_Rev" (basic information of a revision), "SH_Minor" (if the revision is major), "SH_Summary" (plain text summary of the revision)
- Templates for triples: "SH_Triple" (basic information of a triple) , "SH_Add" (link to a revision that adds this triple), "SH_Delete" (link to a revision that deletes this triple), "SH_Obsolete" (flag for an obsolete triple, i.e., it has been deleted and not yet restored).

Such a design brings a couple of advantages. First, this allows us to seamlessly integrate existing SMW-based tools (e.g., semantic query engine) with the SH extension since all revision related data is also in the SMW triple store. Second, the SH templates can be easily customized or extended based on a specific domain need. For instance, one may choose a vocabulary (category and property names) that is best aligned with other ontology terms on the wiki. Finally, since a template can render both semantic data and non-semantic text (e.g., layout and visual elements), using a template based

approach makes it easier to create a user interface for browsing revisions and changed triples.

**Identifying a Triple**. Each triple is identified by an id computed using a hash function from its subject, predicate and object values. Currently we use SHA-1 (a cryptographic hash function) to generate 40-character ids for triples.

**Parsing Semantic Editing Summary**. We deliberately do not specify a syntax or a parser including how to explain the meaning of an editing summary. This would allow an application to implement their own syntax. Here we show two example syntaxes that have been proven useful on social Web applications.

- The SMW annotation syntax[3]: For example, one edit summary is:

  ```
  reason::data is outdated; source::CIA World Factbook;
  category:Fact Update; Update GDP numbers with the 2008 data
  ```

  It contains three semantic annotations in a format similar to the SMW syntax, and one non-semantic sentence explaining the change. This may be parsed into SMW scripts:

  ```
  [[reason::data is outdated]]
  [[source::CIA World Factbook]]
  [[category:Fact Update]]
  ```

  The non-semantic text will not be parsed and is stored in the original form. By semanticizing the editing summary, we will be able to perform more powerful queries over the revision data. For example, one may ask for revisions about a country that uses some information from an almanac (e.g., CIA World Factbook), we may use the following SMW query (based an ontology that contains categories Revision, Countries and Almanacs):

  ```
  {{#ask: [[Category:Revision]]
       [[about::<q>[[Category:Countries]]</q>]]
       [[source::<q>[[Category:Almanacs]]</q>]]
  }}
  ```

- The Twitter-style syntax: one may use "#" to add tags to a summary. For instance, a sentence "fact update for 2008 #gdp #infobox change" maybe generate SMW annotations:

  ```
  [[tag::gdp]]
  [[tag::infobox]]
  ```

  (this revision is an infobox editing, and is about GDP number change).

  The source code of the Semantic History extension has been release at the MediawWiki site[4].

[3] Its parser is implemented at http://tw.rpi.edu/proj/semhis.wiki/index.php/Template:SH_Summary
[4] http://www.mediawiki.org/wiki/Extension:SemanticHistory

## 4 Usage Scenario Examples

Applications may use revision history data for a variety of purposes. We demonstrate some of the SH extension's potential with several hypothetical applications[5].

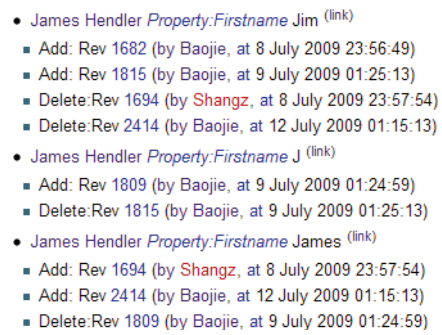


**Fig. 3.** Provenance Tracking

**Provenance Tracking**: One may be interested in tracking the triple changes such as who has changed it? when it was changed? Fig 3 shows an example that asks "Who has changed the first name of James Hendler?". All triples related to this are retrieved by a semantic query. An generalization of the example is a "Semantic Recent Changes" page that lists all page-level and triple-level revisions in inverse chronological order.

**Statistics and Visualization**: SMW provides tools for statistic queries and visualization of query results. We show two examples in Figure 4. In (a), we query about relative popularity of two types of pages, university and person, on the basis of total triple-level revision numbers to their instances. The query result is shown in a pie charter. In (b), we count the number of daily revisions for a specific week and visualize the result in a line charter.

**Temporal Reasoning**: This example (Figure 5) shows the use of revision timestamps in inferring time-sensitive facts. To make the RPI Tetherelss World group publication list, we need to know the affiliation history of a person. Since Jie Bao became a number of the RPI in 2008, only the publications that are published after this date should be added to the list. This query can be done in two steps:

- Look up triple changes with subject "Jie_Bao" and predicate "affiliation", and get the time span(from datatime value 20080226045135 to current) when the object was change to "RPI";
- Use the time span as a filter in querying Jie Bao's publications. As shown in Figure 5, only papers published after time 20080226045135 qualify in the list.

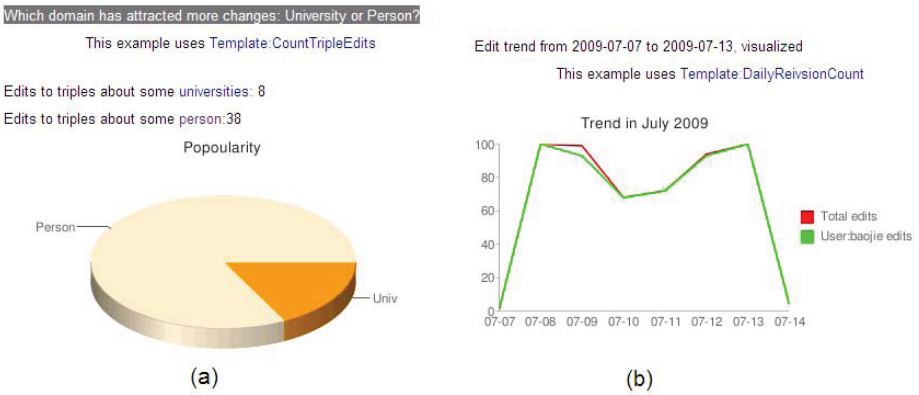


**Fig. 4.** Statistics and Visualization Application Examples

| Subject | Predicate | Object | Revision |
|---|---|---|---|
| Jie Bao | Affiliation | Iowa State University | 20070710044843:add:rev:1986:User:Baojie<br>20080226045135:delete:rev:1998:User:Baojie |
| Jie Bao | Affiliation | RPI | 20080226045135:add:rev:1998:User:Baojie |

From this, we can get the papers in the time period when he is a member of RPI:

| Title | Author | Year |
|---|---|---|
| On the Decidability of Role Mappings Between Modular Ontologies | Jie Bao<br>George Voutsadakis<br>Giora Slutzki<br>Vasant Honavar | 2,008 |
| Rule Modeling using Semantic MediaWiki | Jie Bao<br>Li Ding<br>Paul R. Smart<br>Dave Braines<br>Gareth Jones | 2,009 |

**Fig. 5.** Temporal Reasoning with the Semantic History Extension

**Explanation**: Due to the fact that a wiki page can use a template, and that template may generate new triples, a triple about page is not necessarily locally generated. For example, page "Jie_Bao" generates the triple "Jie_Bao Property:Member_of ITA"; however, "Jie_Bao" does not contain an explicit assertion for creating a Member_of triple. Thus, we need to explore other possible explanations.

> Step 1: the triple "Jie_Bao Member_of ITA" is created at revision rev:2934, on time 20090713102155, by author User:Baojie; however rev:2934 does not directly generate this triple.
> Step 2: rev:2934 uses template "Template:Member of" of revision 2931.
> Step 3: rev:2931 may generate properties: member,member of
> Step 4; Thus, one possible explanation of "Jie_Bao Member_of ITA" is the combination of rev:2934 to Jie Bao and rev:2931 to Template:Member_of.

This process can be encoded by automated queries and templates. Please see http://tw.rpi.edu/proj/semhis.wiki/index.php/Template:Explain for details.

[5] Live demos of some examples are at http://tw.rpi.edu/proj/semhis.wiki/index.php/Main_Page#Examples

## 5 Related Work

**Temporal Knowledge Representation.** The temporal aspect of data has been investigated in numerous communities including Database [13] and AI (e.g., Temporal Representation and Reasoning [4] and Temporal Description Logic[10]). Recently, there has been some work on adding temporal features to OWL, e.g., [8, 20, 11], to RDF[7], and to SPARQL[17, 15]. This work is useful for representing temporal data descriptions. Our work is different from this work in that we are focused on generating and publishing a Web history, and our work is not tied to one particular temporal knowledge representation formalism. It would be possible to adapt our generic model and the SMW-based implementation to represent wiki revisions with some of the aforementioned formalisms. The focus of our work is also different from the the above in that we present a generic revision change model not only addressing temporal changes in the semantic data, but also revision action descriptions, e.g., rationale information for the change.

**Detecting and Representing Changes**. Automatically generating some meaningful changes of structured data has been investigated in database research[3]. History publishing was discussed in [16, 12] in the Wikipedia context. There are some similar work on Semantic Web data [9, 14]. However, the work [16, 12] does not support structured representation of revision information that can be consumed for general purposes, e.g., query, search and reasoning. The work [9, 14] provides means for detecting changes in ontology evolution, but does not address the publishing the change history information as semantic data. Our work addresses the issues of capturing and publishing revision data in an end user friendly way, which are missing from the aforementioned related work.

**Semantic Difference in RDF**. Revisions on an RDF graph can be directly captured by the addition and deletion of triples introduced by the new version, and there has been a number of work from a syntactic perspective [2] or semantic perspective [6, 19] approaching this "diff" problem [1]. Diff is also investigated in synchronizing RDF graphs [5, 18]. This work can be seen as a complement of our work for detecting triple-level changes. However, our work captures not only triple changes, but also provenance and rationale information associated with these changes.

## 6 Conclusion

In this work, we investigate how to make the revisions of online data available in a meaningful way for common Web users using Semantic Web technologies. We described a generic application model that captures both revision action description and temporal data description for a revision. The model is implemented extending an existing platform, namely Semantic MediaWiki. We show that by encoding history information using semantic Web technologies, several interesting applications can be built, e.g., provenance tracking, temporal reasoning and explanation generation.

Our future work will focus on developing additional services that may utilize semantic history information, including trust computations on semantic wikis, advanced

explanation for revisions, and publishing of the revision data of Wikipedia as a part of the linked data cloud.

## Acknowledgments

This work is partially supported by NSF #0524481, DARPA #FA8650-06-C-7605, #FA8750-07-D-0185, #55-002001, #F30602-00-2-0579, and ITA project W911NF-06-3-0001.

## References

1. T. Berners-Lee and D. Connolly. Delta: an ontology for the distribution of differences between rdf graphs. `http://www.w3.org/DesignIssues/Diff` (last visited on Oct 5 2009, Revision: 1.114), 2004.
2. J. J. Carroll. Signing RDF graphs. Technical Report HPL-2003-142, HP Lab, Jul 2003.
3. S. S. Chawathe and H. Garcia-Molina. Meaningful change detection in structured data. In *SIGMOD Conference*, pages 26–37, 1997.
4. M. Fisher, D. Gabbay, and L. Vila. *Handbook of Temporal Reasoning in Artificial Intelligence (Foundations of Artificial Intelligence (Elsevier))*. Elsevier Science Inc., New York, NY, USA, 2005.
5. J. N. Foster and G. Karvounarakis. Provenance and data synchronization. *IEEE Data Eng. Bull.*, 30(4):13–21, 2007.
6. C. Gutierrez, C. Hurtado, and A. O. Mendelzon. Foundations of semantic web databases. In *PODS '04: Proceedings of the 23rd ACM symposium on principles of database systems*, pages 95–106, 2004.
7. C. Gutiérrez, C. A. Hurtado, and A. A. Vaisman. Temporal rdf. In *ESWC*, pages 93–107, 2005.
8. J. R. Hobbs and F. Pan. An ontology of time for the semantic web. *ACM Trans. Asian Lang. Inf. Process.*, 3(1):66–85, 2004.
9. M. C. A. Klein, A. Kiryakov, D. Ognyanov, and D. Fensel. Finding and characterizing changes in ontologies. In *ER*, pages 79–89, 2002.
10. C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *TIME*, pages 3–14, 2008.
11. V. Milea, F. Frasincar, and U. Kaymak. Knowledge engineering in a temporal semantic web context. In *ICWE*, pages 65–74, 2008.
12. S. Nunes, C. Ribeiro, and G. David. Wikichanges - exposing wikipedia revision activity. In *Procceedings of the 2008 International Symposium on Wikis (WikiSym) Porto, Portugal*, 2008.
13. N. Pelekis, B. Theodoulidis, I. Kopanakis, and Y. Theodoridis. Literature review of spatio-temporal database models. *Knowl. Eng. Rev.*, 19(3):235–274, 2004.
14. P. Plessers, O. D. Troyer, and S. Casteleyn. Understanding ontology evolution: A change detection approach. *J. Web Sem.*, 5(1):39–49, 2007.
15. F. Rizzolo, Y. Velegrakis, J. Mylopoulos, and S. Bykau. Modeling concept evolution: a historical perspective. In *28th International Conference on Conceptual Modeling (ER)*, 2009.
16. M. Sabel. Structuring wiki revision history. In *WikiSym '07: Proceedings of the 2007 international symposium on Wikis*, pages 125–130, New York, NY, USA, 2007. ACM.
17. J. Tappolet and A. Bernstein. Applied temporal rdf: Efficient temporal querying of rdf data with sparql. In *ESWC*, pages 308–322, 2009.

18. G. Tummarello, C. Morbidoni, R. Bachmann-Gmür, and O. Erling. Rdfsync: Efficient remote synchronization of rdf models. In *ISWC/ASWC*, pages 537–551, 2007.
19. M. Völkel, C. F. Enguix, S. R. Kruk, A. V. Zhdanova, R. Stevens, and Y. Sure. Semversion - versioning rdf and ontologies. Knowledge Web Deliverable 2.3.3.v1, University of Karlsruhe, June 2005.
20. C. A. Welty and R. Fikes. A reusable ontology for fluents in owl. In *FOIS*, pages 226–236, 2006.