| Change operation | Description |
|---|---|
| *addC(c), delC(c)* | addition/deletion of concept *c* |
| *toObsolete(c), revokeObsolete(c)* | set/revoke ‚to obsolete' status of *c* |
| *split(s, T)* | split a source concept *s* into several target concepts *T* |
| *merge(S, t)* | merge several source concept *S* into one target concept *t* |
| *substitute(c, c')* | substitute concept *c* by concept *c'* |
| *move(c, P, P')* | move a concept *c* from parents *P* to parents *P'* |
| *addR(r), delR(r)* | addition/deletion of a relationship *r* |
| *chgAttValue(c, att, v1, v2)* | change value of *att* in *c* from *v1* to *v2* |
| *addA(a), delA(a)* | addition/deletion of an attribute *a* |

**Fig. 5.** Change operations in COnto-Diff [27,28].

Moreover, Dos Reis et al. defined lexical and semantic change patterns based on the evolution of several medical ontologies [31]. These change patterns allow to characterize the way attribute values of concepts evolve, e.g., if the observed changes are likely to modify the meaning of an attribute value. The change operations and change patterns as determined by diff algorithms and other methods are very useful to maintain ontology-based mappings and other dependent applications.

The need for a retrospective identification of differences between versions can be avoided during editing when ontology changes are well-documented including reasons of changes. For instance, it is important to document change operations in an upper level formal ontology such as Basic Formal Ontology (BFO) in order to allow for an appropriate change propagation into dependent domain ontologies [32]. Evolutionary terminology auditing (ETA) allows for measuring the quality improvements of formal ontologies and different kinds of terminologies over successive versions, and requires that ontology editors keep track of changes and their motivation for the respective changes (e.g., [33,34]). However, so far there is no standard language for documenting ontology evolution.

### *Visualizing ontology evolution*

It has become more and more important to provide intuitive ways of visualizing ontology evolution (e.g., [18,35–39]). For users it is particularly important to understand the evolution of ontologies they use in order to be able to assess possible influences on their ontology-based applications. For instance, quite recently a new version of the widely used ontology lookup service[1] [18] was introduced to inform users about ontology change histories. The CODEX[2] [35] tool allows users to explore complex changes computed by COnto-Diff. WebProtégé[3] [36] supports the tracking of ontology changes and provides precisely defined, OWL-related ontology changes and change lists. Diff Abstraction Networks [37] were introduced to summarize, visualize and highlight ontology changes. It further seems intuitive to provide a dynamic graph visualization perspective for time-varying ontologies [40]. For instance, the tool REX[4] [38] gives an aggregated view on differently evolving ontology regions and allows users to navigate from the root into stable or strongly evolving ontology regions using a fish-eye zoom. However, still much work needs to be done to improve ontology evolution visualization techniques allowing for compact as well as detailed views e.g. on precisely defined changes of axioms in formal ontologies.

### *Ontology change prediction*

In the last years, the tracking and prediction of ontology evolution has gained attention. This is of special interest for collaborative ontology editing and development as well as for the migration of ontology-based applications. Also change prediction methods can not guarantee to be perfectly correct and precise, they can support users in planing and managing adaptation processes, e.g. by precociously indicating possibly impacted parts of dependent mappings and applications. Current relevant work includes [41] where the authors focused on tracking the collaborative processes behind the evolution of an ontology, i.e., the changes made by contributors over time. Wang et al. investigate the way ontology editors behave when they modify an ontology and predict future modifications [42]. Moreover, Pesquita & Couto used machine learning techniques to predict which branch of the Gene Ontology is likely to expand in the future release using supervised learning methods [43]. Tsatsaronis et al. implement temporal classifiers to predict future extension of the MeSH controlled terminology using MeSH-indexed PubMed articles [44].

## 4. Adaptation of ontology-based mappings

One of the additional challenges of ontology evolution is to keep dependent artifacts such as ontology-based mappings up-to-date. Several evolution studies in the life science domain (e.g., [45–47]) showed frequent and continuous changes for both, the considered ontologies and ontology-based mappings. In particular, the results in [46] showed significant instabilities for mappings created by automatic ontology matching techniques, e.g., utilizing the similarity of concept names and their synonyms for deriving correspondences. These observations underline the importance of (semi-) automatic adaptation strategies that can reuse and extend previous mappings instead of completely recomputing the mappings when an ontology changes.

In the following we will first discuss requirements for the adaptation of ontology-based mappings. In Section 4.2 we will then discuss adaptation strategies for ontology-based mappings in the context of ontology evolution and compare them based on the introduced requirements. Approaches for the more general problem of mapping maintenance and repair are discussed in [48].

### *4.1. Requirements*

(Semi-) automatic mapping adaptation strategies need to achieve several requirements to be useful for applications and users:

- **Mapping quality:** Mapping adaptation methods need to determine high-quality mappings. The correspondences in migrated mappings need to be correct and complete, i.e., methods need to achieve high precision and recall values.
- **Mapping validity:** An adapted mapping needs to cover solely correspondences to valid concepts from the new ontology versions. Mappings must not contain any inconsistent correspondences, e.g., to obsolete or deleted concepts.
- **Inclusion of added concepts:** Mapping adaptation methods need to involve ontology extensions such as concept additions in order to obtain a complete result mapping. This is especially relevant for highly volatile domains such as the life sciences where ontologies are heavily extended.
- **Reduction of manual effort and user involvement:** The adaptation process should be largely automatic to limit the manual effort, especially for very large ontologies and mappings. One main aim is to reuse large parts of an existing mapping and avoid a full redetermination. User involvement is very important, but should mainly be restricted to verify and potentially revise automatically updated mappings.
- **Scalability and efficiency:** Mapping adaptation approaches should be efficient and scalable to process large ontologies and mappings as common in the biomedical domain.

1 http://www.ebi.ac.uk/ols.
2 www.izbi.de/codex.
3 http://webprotege.stanford.edu.
4 www.izbi.de/rex.