Web Semantics: Science, Services and Agents on the World Wide Web 74 (2022) 100715

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Visualising the effects of ontology changes and studying their understanding with ChImp

Romana Pernisch [a,b,c,*], Daniele Dell'Aglio [d,a], Mirko Serbak [a], Rafael S. Gonçalves [e], Abraham Bernstein [a]

[a] Department of Informatics, University of Zurich, Zurich, Switzerland
[b] Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[c] Discovery Lab, Elsevier, Amsterdam, The Netherlands
[d] Department of Computer Science, Aalborg University, Aalborg, Denmark
[e] Center for Computational Biomedicine, Harvard Medical School, Boston, MA, USA

ARTICLE INFO

Article history:
Received 26 August 2021
Received in revised form 14 February 2022
Accepted 27 April 2022
Available online 5 May 2022

Keywords:
Ontology editing
Materialisation
User study
Ontology evolution impact

ABSTRACT

Due to the Semantic Web's decentralised nature, ontology engineers rarely know all applications that leverage their ontology. Consequently, they are unaware of the full extent of possible consequences that changes might cause to the ontology. Our goal is to lessen the gap between ontology engineers and users by investigating ontology engineers' understanding of ontology changes' impact at editing time. Hence, this paper introduces the Protégé plugin ChImp which we use to reach our goal. We elicited requirements for ChImp through a questionnaire with ontology engineers. We then developed ChImp according to these requirements and it displays all changes of a given session and provides selected information on said changes and their effects. For each change, it computes a number of metrics on both the ontology and its materialisation. It displays those metrics on both the originally loaded ontology at the beginning of the editing session and the current state to help ontology engineers understand the impact of their changes.

We investigated the informativeness of materialisation impact measures, the meaning of severe impact, and also the usefulness of ChImp in an online user study with 36 ontology engineers. We asked the participants to solve two ontology engineering tasks – with and without ChImp (assigned in random order) – and answer in-depth questions about the applied changes as well as the materialisation impact measures. We found that ChImp increased the participants' understanding of change effects and that they felt better informed. Answers also suggest that the proposed measures were useful and informative. We also learned that the participants consider different outcomes of changes severe, but most would define severity based on the amount of changes to the materialisation compared to its size. The participants also acknowledged the importance of quantifying the impact of changes and that the study will affect their approach of editing ontologies.

1. Introduction

Ontologies are used in research and industrial applications, where they often constitute central components of complex data-driven solutions. Since ontologies model a specific domain and its knowledge, they have to evolve, shift, or change over time to accommodate advancements in the respective domains [1]. Ontology engineers and maintainers take care of keeping ontologies up to date based on the requirements of applications or the state of the knowledge they model. Their day-to-day tasks include updating an ontology and releasing new versions thereof [2].

Ontologies are, however, often used by many other parties besides their direct maintainers. Because of the Semantic Web's decentralised nature, ontologies are shared freely online, inviting other people to use them in their applications [3]. This creates a communication gap between ontology engineers and ontology users, where neither of the groups know about the other's needs or progress. There have been previous investigations on bridging this gap and supporting ontology engineers during the change process, but those focus on the process itself rather than the added benefit of displaying more information about changes [4].

Respective changes do, however, affect not only the ontology itself (e.g., its consistency and quality), but also the services built on top of it. For example, the querying benchmark QUALD [5],

* Corresponding author at: Department of Informatics, University of Zurich, Zurich, Switzerland.

E-mail addresses: r.pernisch@vu.nl (R. Pernisch), dade@cs.aau.dk (D. Dell'Aglio), mirko.serbak@gmx.net (M. Serbak), rsgoncalves@gmx.com (R.S. Gonçalves), bernstein@ifi.uzh.ch (A. Bernstein).

https://doi.org/10.1016/j.websem.2022.100715

which provides a large set of natural language questions with their corresponding SPARQL queries, was developed to be compatible with both Wikidata [6] and DBpedia [7]. However, these two large-scale knowledge graphs are edited by the community and evolve at a fast rate. Depending on these changes, some parts of this benchmark will need to be updated. Inexperienced engineers may lack the expertise to fully grasp all the consequences of their actions. Moreover, experienced engineers are likely to work with ontologies they do not know well and might, therefore, not fully understand the effect of changes. We argue that engineers need a better understanding of the effect of their changes using multiple perspectives, including the change's semantic and structural consequences while they are changing the ontology. Therefore, in this research, we tackle the problem of *lessening the gap between ontology engineers and users by investigating ontology engineers' understanding of ontology changes' impact at editing time*.

To investigate the aforementioned problem, we require a tool with which we can provide engineers with summarised information about changes and its effects on the materialisation. Hence, in this paper we introduce ChImp (*Ch*ange *Imp*act), a Protégé [8] plug-in to display information related to the changes. Before building the ChImp plugin, we went through a design process to answer the following research question:

RQ1: What do ontology engineers want to see in a Protégé plugin which summarises changes and the effect of changes?

Departing from the assumption that a plugin displaying information would be useful to assess change impacts, we gathered requirements using an online questionnaire. The *requirements survey* contained mock-ups of change visualisations for rating, opportunities to provide detailed explanations of preferences, and general questions about demographics, already established practices, and the topic itself. Based on the responses we received, we built ChImp offering three perspectives on changes: a summary of the performed changes, the consistency of the ontology and materialisation impact measures, and changes to ontology measures (such as number of classes or properties).

With ChImp, we can then investigate the problem of communication and understanding of ontology change effects by asking the following three research questions with a user study, to which we will be referring to as the impact understanding study.

RQ2: Do ontology engineers understand the effect of changes on the ontology and on the materialisation better when using ChImp than without?

RQ3: What does severe impact on the ontology and on the materialisation mean to ontology engineers?

RQ4: Are the materialisation impact measures useful and informative for ontology engineers?

The impact understanding study was executed independently and after the requirements survey. We used a within-subject design, where our 36 participants solved two predefined tasks (one with and one without ChImp in random order) using the Pizza Ontology.[1] We then analysed the participants' answers qualitatively. Throughout the impact understanding study, participants realised that understanding the effect of changes is important, and ChImp provided valuable information to them to think about the effects of changes constructively. "Severity of impact" means something different to every ontology engineer, but the most common metrics were the consistency of ontology and the number of changes to the materialisation or ontology. Additionally, participants agreed that the impact measures are intuitive, useful, and informative. Having the measures displayed in ChImp helped them get an intuition of how much the materialisation is changing, in turn, also making them understand the consequences of their actions better.

Given the above, our contributions are[2]:

- the requirements for a Protégé plugin, which summarises changes and the effect of changes
- the ChImp plugin, which is based on the elicited requirements, and
- using the qualitative evaluation with ChImp, we gained various insights about ChImp and impact understanding such as:
  - Most participants found ChImp to be useful in informing them about the effect of changes, helping them keep an overview of changes and their consequences.
  - Most participating ontology engineers defining severity of impact based on the amount of changes to the materialisation, which directly coincides with the introduced impact measures.

In the next section, we introduce the related research on ontology change impact and ontology editing. We then introduce the impact measures in Section 3. In Section 4, we present our first contribution, the requirements survey and the found requirements. In Section 5, we detail the implementation of ChImp, our second contribution, and compare it to other tools. Subsequently, we focus on the third contribution in Section 6, which includes the impact understanding study setup, approach as well as results and discussion. In Section 7, we present limitations and future work, and we conclude in Section 8.

2. Related research

This section addresses two related research topics: impact of ontology evolution and ontology editing. First, we discuss the more general topic of ontology evolution impact and previous research which mainly focus on other tasks such as functional enrichment analysis [10], indexing [11], alignment [12], or automatic annotation [13]. Second, we present studies which investigate ontology change and tools for supporting the ontology engineering process as a whole [14,15] as well as specifically dealing with quality assurance and consequences of ontology changes [4,16–18], which is related to some extent. To the best of our knowledge, no studies so far have investigated how ontology engineers perceive the impact of their changes, and there are no tools that help engineers in understanding the overall impact either.

2.1. Evolution impact

According to Noy and Klein [19], ontology evolution is not equivalent to database schema evolution. They found that evolutionary consequences are difficult to foresee because of the decentralisation of ontologies. Gonçalves et al. [20] propose a categorisation of changes based on a logical impact. They investigate whether changes affect the set of entailed axioms in the next version, and distinguish between effectual and ineffectual changes. Groß et al. [10] examine how changes in an ontology impact previously conducted functional analyses. They point out that results could be invalidated due to the Gene Ontology's evolution over time. Gottron and Gottron [11] also investigate

[1] https://protege.stanford.edu/ontologies/pizza/pizza.owl.

[2] This paper extends our previous work on ChImp published at the Workshop for Visualisation and Interaction for Ontologies and Linked Data [9] with the impact understanding study investigating ***RQ2–RQ4***.

the impact of ontology evolution using Linked Open Data. They implement twelve different indexing methods and evaluate how respective indices are affected by the evolution of the data using three different measures. dos Reis et al. [12] look into the impact concerning mappings between two evolving ontologies. Qawasmeh et al. [21] investigate the influence of evolution on imported ontologies. Cardoso et al. [13] identify the impact on annotation creation using an evolving ontology. Osborne and Motta [22] present the pragmatic ontology evolution, in which they analyse the selection of concepts for a new version by evaluating the performance of four different tasks.

In previous work [23], we investigate and predict the impact of evolution on knowledge graph embeddings by comparing neighbourhoods. More recently [24], we also investigate the impact on the materialisation and propose new measures to quantify this impact with simple-to-compute measures, which are included in ChImp for further investigation.

2.2. Change visualisations and user studies

Katifori et al. [25] and Dudás et al. [26] provide two important contributions to ontology visualisation. The former covers a range of ontology visualisation methods and techniques; it discusses the strengths and weaknesses of each method and addresses the issue of visualising time-related data. The latter presents the current state of the art in ontology visualisation. It states that there is no de-facto standard of visualisation, which has been accepted by the Semantic Web community, due to the fact that there is no single solution that fits all applications. There are multiple plugins available that partially address either ontology difference [27,28] or change tracking [29–32]. To contrast them with our contribution, we will use these plugins as a baseline and discuss them in more detail in Section 5.3.

Multiple studies focus on the ontology engineering and the ontology evolution process; Researchers have developed and tested tools, but have not addressed change effect understanding so far. The study of Vigo et al. [14] focuses on ontology authoring to provide new guidelines and recommendations for the entire process. Mohsen et al. [15] provide another study of ontology engineering from the perspective of the SCRUM methodology. These studies investigate the editing process as a whole, in contrast to our work, where we investigate the understanding of change effects.

Specific tools, which are more related to ChImp, have also been tested in a user setting. Davies et al. [33] investigate a fest-first approach using their Protégé plugin TDDonto2 and found that participants were able to author changes quicker and with fewer mistakes using TDDonto2 compared to only using Protégé. However, they do not consider the understanding of consequences. Another approach for better ontology engineering processes is fact-oriented, where Leenheer and Debruyne [17] provide a collaborative tool for ontology evolution. Denaux et al. [18] argue for a more interactive approach, visualising consequences as soon as changes are made. They present a framework supporting this approach; however, they do not provide a use case-oriented evaluation and only conduct interviews. Because of the missing in-depth evaluation in [18], Matentzoglu et al. [34] present the Protégé plugin *Inference Inspector*, which they evaluate with an exploratory user study. The Inference Inspector's goal is to present the consequences of changes in the entailment interactively, inform the ontology engineer about respective consequences, and ease the debugging and the verification process. In [16], the authors provide a controlled study to verify that the Inference Inspector satisfies the intended goal and they show that their tool performs better than simply using Protégé. The tools used in these studies [16–18,33] aid in resolving conflicts or providing a better environment to avoid unwanted consequences of changes. Alrabbaa et al. [35] also provide a tool for the resolution of defects in an ontology by visualising the inferences and helping the user to solve the issue. Our plugin ChImp, in contrast, is developed with the goal to inform the users about the executed changes to raise understanding of possible impact at editing time.

3. Materialisation impact

In many engineering disciplines, it is typical to measure both the artefact constructed and the impact of changes. Such metrics help engineers to keep track of relevant aspects. It is also important to note that ontologies typically entail some semantics. Consequently, there is a difference between what is explicitly stated in an ontology and what can be inferred, which is what we refer to as materialisation throughout this paper.

We regard the ontology O_i at a time instant i as a set of axioms. The difference between O_i and O_j at the next evaluated time instant j is captured by $\delta_{i,j}$, denoting the set of axioms that changed between i and j. Note that there is a difference between added ($\delta^+_{i,j} = O_j \setminus O_i$) and deleted ($\delta^-_{i,j} = O_i \setminus O_j$) axioms. The cardinality $|\delta_{i,j}|$ is the sum of added and deleted axioms $|\delta^+_{i,j}| + |\delta^-_{i,j}|$. Further, we indicate with M_i the set of axioms inferred (or materialised) from O_i, such that M_i does not include the axioms from O_i, ergo $M_i \cap O_i = \emptyset$. Throughout this document, M refers to the set of axioms inferred by a reasoner and not the set of all axioms of the ontology (i.e. both raw and inferred ones). We define $\Delta_{i,j}$ as the difference between M_i and M_j. Again, we differentiate between added $\Delta^+_{i,j}$ and deleted $\Delta^-_{i,j}$ materialisation axioms.

The impact metrics which we study in this work have been presented in our previous work [24]. To our knowledge, these are still the only measures that are simple enough to compute and capture the impact on the materialisation between versions at run-time. In [24], we defined the measures and used them in an analysis of nine open biomedical ontologies. In this work, we want to evaluate the acceptance and usefulness of these metrics by ontology engineers. Below, we introduce some necessary terminology for the reader to understand the definitions of the impact measures, which are stated at the lowest part of Table 2.

The *size-based impact* σ measures how much the materialisation's size is changing between two versions of the ontology. It is defined as:

$$\sigma_{i,j} = \frac{|\Delta_{i,j}|}{|M_i \cap M_j|} \tag{1}$$

This metric indicates the overall size of the materialisation's change between two versions. Further, we assume that the intersection between M_i and M_j is not empty.

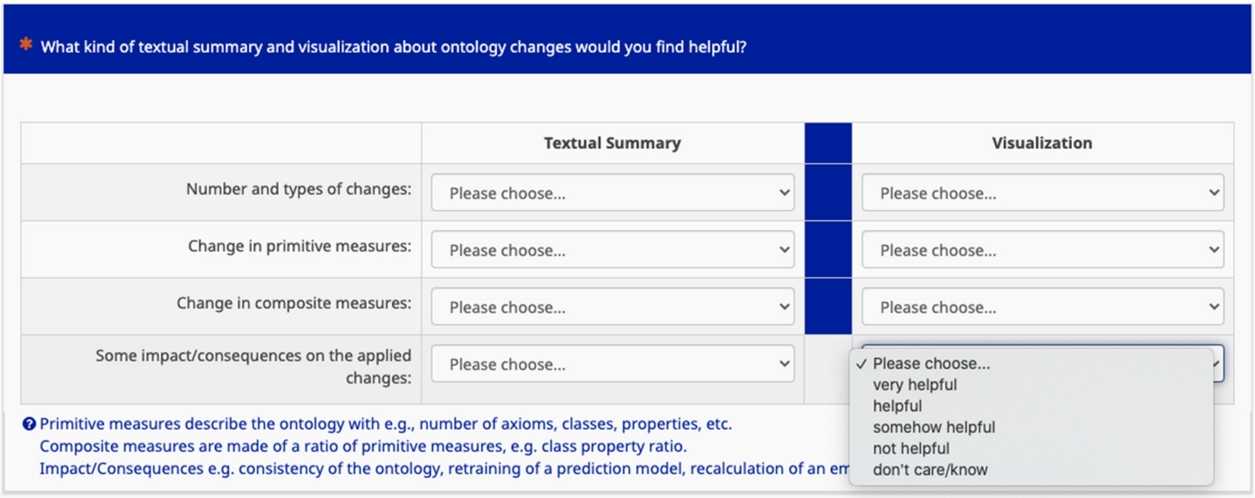
The *change-based impact* γ quantifies how much a change in the ontology impacts its materialisation:

$$\gamma_{i,j} = \frac{|\Delta_{i,j}|}{|\delta_{i,j}|} \tag{2}$$

When γ is close to 1, the amount of changes to the inferred axioms is close to the number of changes to the ontology.

To summarise, the *size-based impact* σ measures how much the materialisation between two versions changes, while the *change-based impact* γ assesses the impact of an average change in the ontology. Additionally, we would like to point out, that these metrics are dependent on the chosen reasoner, entailment selection criteria of the reasoner and other parameters set by users. In this work, we do not distinguish this difference, however, it is very important to keep this in mind, when using different approaches to optain M_i and M_j.

Besides impact measures, in [24] we collected a selection of other measures, with which one can describe ontologies. Will also



* What kind of textual summary and visualization about ontology changes would you find helpful?

	Textual Summary	Visualization
Number and types of changes:	Please choose...	Please choose...
Change in primitive measures:	Please choose...	Please choose...
Change in composite measures:	Please choose...	Please choose...
Some impact/consequences on the applied changes:	Please choose...	✓ Please choose... very helpful helpful somehow helpful not helpful don't care/know

Primitive measures describe the ontology with e.g., number of axioms, classes, properties, etc.
Composite measures are made of a ratio of primitive measures, e.g. class property ratio.
Impact/Consequences e.g. consistency of the ontology, retraining of a prediction model, recalculation of an em

Fig. 1. Last question from the second part of the requirements survey, "Changing an Ontology". Each drop-down shows the same selection options, and each requires an answer.

use them in this work. We differentiate between primitive measures and composite measures. Primitive measures are simple counts of classes, properties, or annotations, whereas composite measures are ratios, like class to property ratio and the number of annotations per class.

4. Requirements elicitation

The first step of the ChImp design process was the elicitation of requirements. While we defined the ones related to the behaviour of ChImp through an internal design process, we collected the requirements about the visualisation through the requirements survey. We first present our survey, focusing on three specific questions, which we present below. We formulated the related requirements that drove the development of ChImp.

4.1. Survey structure

The requirements survey consists of four main sections and can be accessed within the supplemental material of this submission. The first section contains questions to collect demographic information. We use it to weigh the responses based on the self-declared expertise of the participants. The second section, titled "Changing an Ontology", collects participants' experience on editing ontologies. It asks questions about different change types, to determine which are the most common. It also inquires about tools (plug-ins or visualisations) related to changes that participants may already use. This part of the requirements survey collects participants' preferences on the information they are interested in monitoring while developing ontologies. The third section, titled "Mock-ups of a Prototype", collects opinions on visualising the changes and their impact. It presents mock-ups visualising Boolean metrics (such as consistency), numerical values (impact of changes, primitive, and composite ontology measures), and categorical variables (e.g., change type). The last section collects feedback and provides a wrap-up. It asks participants about their interest and opinion on ontology evolution and tools to monitor it. Additionally, it inquires about availability to participate in follow-up studies. The collection of the requirements mainly rely on three questions, one in the second section of the requirements survey and two in the third one.

Fig. 1 shows the first question we analyse, which we label the *helpfulness question*. It investigates the degree of helpfulness (from not helpful to very helpful) of individual features describing ontology changes and how to visualise them. The information about changes includes number and types of changes, a variation of primitive measures, composite measures, and consequences of change like ontology consistency. We propose two visualisation styles: textual and graphical. As the names suggest, the former consists of descriptions, numbers and tables, while the latter includes plots and charts. We decided not to provide any visual aids to avoid driving participants towards specific types of plots or text. For each type of information and visualisation style, the participants express its helpfulness using a drop-down menu. All the answers are mandatory to nudge participants to consider each option, instead of simply skipping certain ones. Among the possible answers, participants can pick "don't care/know", to capture the cases where they do not have any opinion or interest in the metrics.

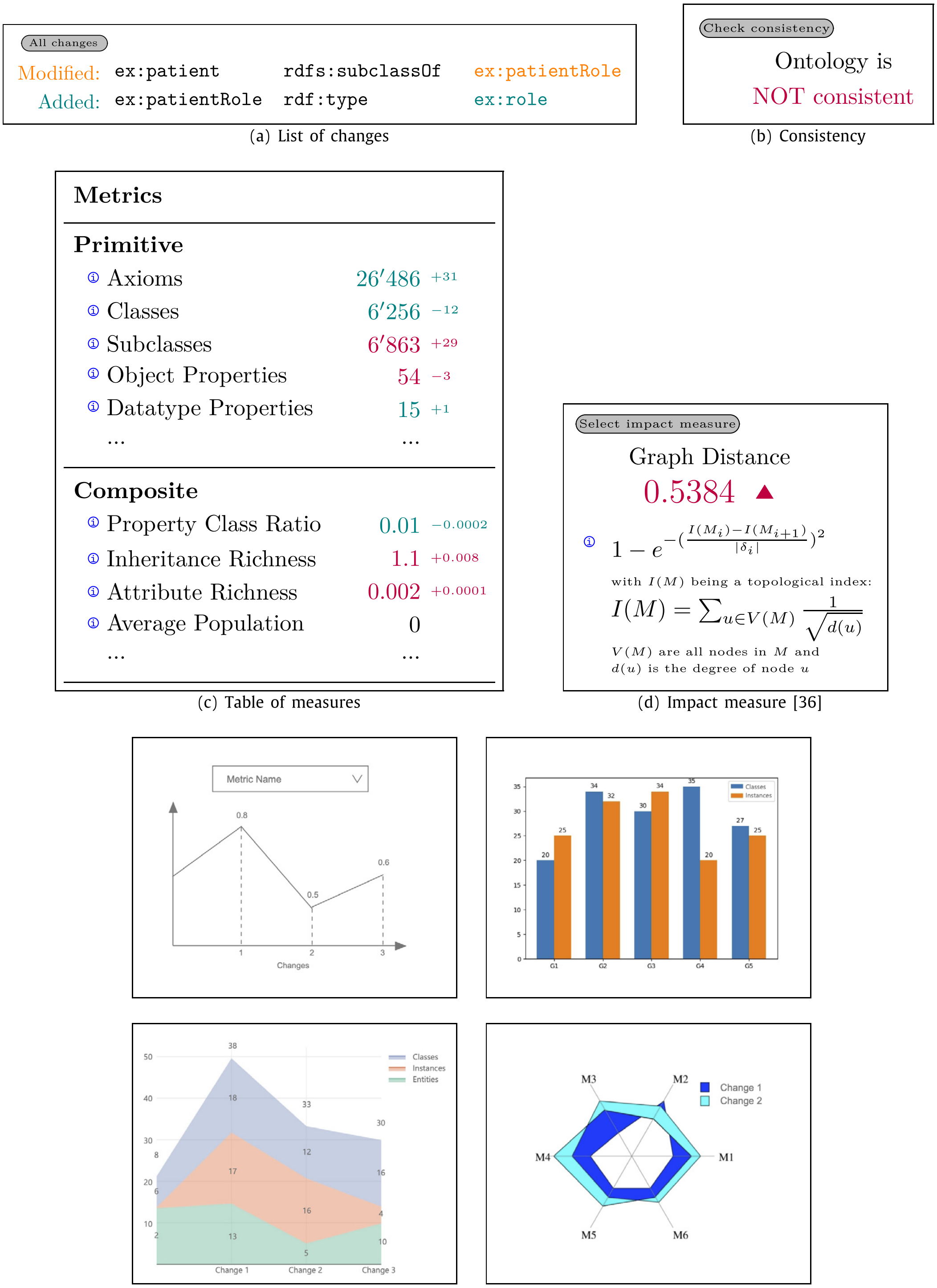
The second question is the *mock-up question*, where participants observed five mock-ups. Fig. 2 shows five mock-ups, each showing a different aspect, such as impact, consistency, changes, and measures [36]. The participants judge each mock-up with a score from 1 (not at all informative) to 5 (very informative). The participants could also choose to not assign any score.

4.2. Participants demographics

We invited semantic web practitioners to answer the requirements survey. We distributed it among the authors' contacts and asked them to share it with colleagues who edit ontologies. 20 people signed up, out of which 12 completed the requirements survey. The remaining eight did not complete it.

The average age is 38.33 with standard deviation *SD* 7.1. Participants claim to have worked with ontologies for 10 years on average ($sd = 5.29$). 42% of participants are working on professorial level, 33% on PhD and Post-Doc level, and the remaining 25% on other research positions, most of which in industry. All participants work in research, except one that works in engineering. Moreover, all participants either still change ontologies regularly or did it in the past. Ten out of twelve participants have used Protégé to change ontologies.

When asking about previously used tools and measures to communicate or visualise changes, only a few participants answered. Specifically, one participant specified that they add an informal description of what has been changed in the `README` document when releasing a new ontology version. Others mentioned using Protégé to check consistency and other requirement



All changes

Modified:	ex:patient	rdfs:subclassOf	ex:patientRole
Added:	ex:patientRole	rdf:type	ex:role

(a) List of changes

Check consistency

Ontology is NOT consistent

(b) Consistency

Metrics

Primitive		
ⓘ Axioms	26′486	+31
ⓘ Classes	6′256	−12
ⓘ Subclasses	6′863	+29
ⓘ Object Properties	54	−3
ⓘ Datatype Properties	15	+1
...	...	
Composite		
ⓘ Property Class Ratio	0.01	−0.0002
ⓘ Inheritance Richness	1.1	+0.008
ⓘ Attribute Richness	0.002	+0.0001
ⓘ Average Population	0	
...	...	

(c) Table of measures

Select impact measure

Graph Distance

0.5384 ▲

ⓘ $1 - e^{-(\frac{I(M_i)-I(M_{i+1})}{|\delta_i|})^2}$

with $I(M)$ being a topological index:

$$I(M) = \sum_{u \in V(M)} \frac{1}{\sqrt{d(u)}}$$

$V(M)$ are all nodes in M and $d(u)$ is the degree of node u

(d) Impact measure [36]

(e) Plots

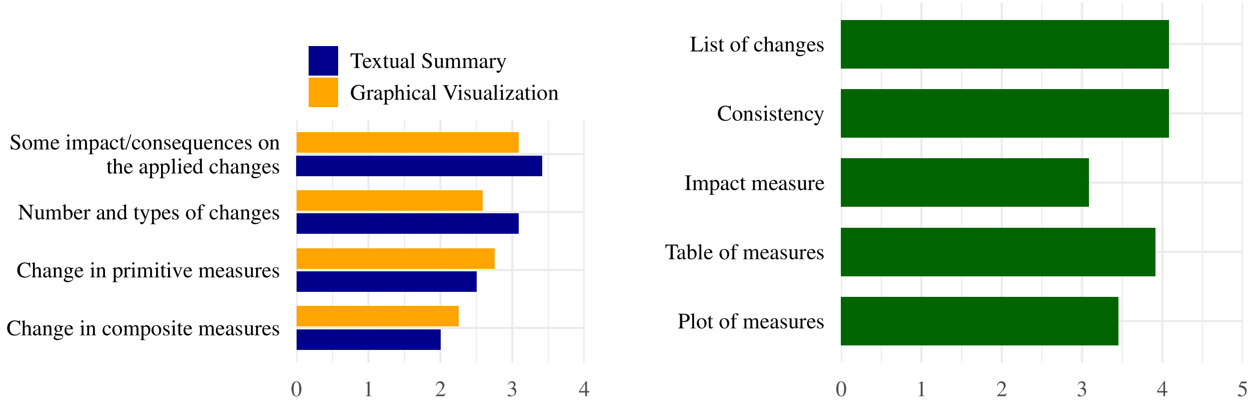
Fig. 2. Mock-ups used in the requirements survey. Participants rated each mock-up from 1 (not at all informative) to 5 (very informative).

compliance before a release. One participant uses a UML-like graphical representation of the changes to inform users about the new version.

4.3. Survey analysis

We report the results of the three questions – helpfulness, mock-up, and plot – in Fig. 3. Fig. 3(a) shows the mean rating for the helpfulness question. We assign values between 0 and 4 to the answers of this question. Values 1 to 4 map to the answers from "not helpful" to "very helpful", while the value 0 maps to "don't know/care". The information about impact/consequences received the highest ratings, followed by the one about the number and types of changes. For these information types, participants prefer textual summaries more than visualisations. We observe the opposite situation for the primitive and composite measures; in these cases, participants prefer visualisation to textual summaries.

Due to the participants' preference for the number and types of changes, we formulate the following requirement:



(a) Results of the helpfulness question. Participants rated the options from "very helpful (4)" to "not helpful (1)" along two axis: textual summary and graphical visualisation. The option "Do not know/care" was rated as 0.

(b) Results of the mock-ups question where mock-ups of different visualisations were presented to the participants. They rated them from.

Fig. 3. Results of the three questions on helpfulness, mock-ups and plots of the requirements survey showing average rating.

R1: ChImp should list the applied changes.

ChImp will report all changes applied during the current session, highlighting the most recent one.

Protégé changes should be grouped based on the action the user has taken, e.g., deleting a class, to make it simpler to recall the actions performed by the engineer. In the background, Protégé might execute more changes triggered by the action of the user. Such subsequent changes should be displayed and grouped with the action of the user.

To decide on further requirements about displaying impact/-consequences as well as primitive and composite measures, we consider the results of the mock-up question as well. The ratings of the answers in the mock-up question already range from 1 to 5. We do not consider the empty answers in these questions. Fig. 3(b) shows the answers to the mock-up question. We observe that participants prefer the answers: consistency, list of changes, and table with measures. **R1** already covers the list of changes. Thus, we derive a second requirement about consistency:

R2: ChImp should inform the user about the consistency of the loaded ontology.

The consistency needs to be explicitly stated and synchronised with the reasoner.

The rating of the impact mock-up is the lowest. We believe that this is because there is not much research on ontology evolution impact. Therefore, we will not formulate a requirement for the impact measure.

In the helpfulness question, we ask the participants about two specific options: a table showing numbers and plots of the metrics. The participants clearly preferred a graphical visualisation. However, while answering the mock-up question, participants do not perceive the plots showing ontology measures as informative and favour the tabular visualisation. Seemingly, the results from the helpfulness and mock-up questions contradict each other. We assume this to be due to lack of contextual information when displaying the plot mock-ups. We decided to only use a table to visualise the change in numbers since this answer was rated higher in the mock-up question of the requirements survey:

R3: ChImp should show primitive and composite measures in a table, visualising the new value and its difference to the old value based on the applied changes.

Section 5 discusses the implemented measures in more detail.

Two participants commented on the choice of colours in the table mock-up. They pointed out that colours are suitable for understanding, but the choice of colour is essential. Therefore, we formulate the following requirement:

R4: ChImp should use colours to indicate changes.

However, the choice of colour should not imply additional meaning, e.g., "good" or "bad". Therefore, we will avoid colours like red and green, also to accommodate colour blindness.

In general, all participants found that the topic of impact and consequences of ontology editing is important, yet there is no universal way of representing consequences. Impact can be very domain-dependent. In the last part of the requirements survey, participants elaborated on this with detailed comments. With biomedical ontologies, engineers might be interested in the impact of changes on the class hierarchy. Where prediction models make use of ontologies, the engineer might want to know when the model needs to be re-learned because the ontology has changed significantly and thus would produce inaccurate predictions. Some participants find ontology consistency to be sufficient, where others suggested including the number of other ontologies and systems which will be (specifically) affected by the changes.

In the open questions, requirements survey participants also suggested adding change logs into versioning systems. These logs could also include changes in primitive measures, as well as some indication of impact. We therefore formulate the following requirements:

R5: Ontology release notes should include the number and types of changes.

These can include the number of additions and deletions of axioms and annotations. They could also be more specific and indicate additions of classes or hierarchy changes. Further, ontology measures such as the number of classes, properties, annotations, or individuals could be reported together with the number of changes. Release notes should also include impact or consequences:

R6: Ontology release notes should include the result of a consistency check.

R7: Ontology release notes should report changes to the materialisation as indication of consequences.

R5, **R6** and **R7** are not required to study the understanding of change effects with ontology engineers.

Other requirements. We also formulate requirements regarding users' interactions with ChImp, as well as its responsiveness. They are based on the authors' experience and best practices.

R8: ChImp should allow the user to choose between the presentation of metrics either in absolute values or as percentages.

Engineers have different preferences in the presentation of numbers. Using percentages has advantages, just like absolute numbers do as well. The particular ontology size can also influence this preference. Therefore, we want to leave the choice of presentation of numbers up to the users.

R9: ChImp should let the user choose using either the last change or all changes for the calculation of primitive and composite measures

While it makes sense to display impact measures cumulatively, we see the potential for both types of calculations regarding ontology measures. The user should be able to make this choice on the fly.

During the time, engineers change the ontology, ChImp executes many calculations in the background and displays them as soon as they are available. We need to ensure that ChImp does not block Protégé while calculating and waiting to display new numbers. Responsiveness is essential for good user experience, particularly when working with large ontologies. Therefore, we capture the following requirement:

R10: ChImp should be responsive

and should not block usage of Protégé while calculating the inference, consistency, or measures.

At the same time, this requirement also covers the update of displays at editing time.

Hence, we can answer ***RQ1***: *What do ontology engineers want to see in a Protégé plugin which summarises changes and the effect of changes?* We elicited seven requirements through the requirements survey, and added three through an internal design process. All ten requirements are listed in Table 1 for reference and overview.

5. The ChImp plug-in

Here, we introduce ChImp's interface and its implementation. ChImp is distributed under the Apache 2.0 licence and can be downloaded from the project website[3] or directly through the Protégé's auto-update plugin library. We aimed to leverage existing code and libraries, following good software engineering practices. Where possible, we based our calculations on already available methods from Protégé [8]. Lastly, we compare ChImp to other Protégé plugins.

Table 1
ChImp requirements as elicited with the questionnaire.

	ChImp requirements
R1	ChImp should list the applied changes.
R2	ChImp should inform the user about the consistency of the loaded ontology.
R3	ChImp should show primitive and composite measures in a table, visualising the new value and its difference to the old value based on the applied changes.
R4	ChImp should use colours to indicate changes.
R5	Ontology release notes should include the number and types of changes.
R6	Ontology release notes should include the result of a consistency check.
R7	Ontology release notes should report changes to the materialisation.
R8	ChImp should allow the user to choose between the presentation of metrics either in absolute values or as percentages.
R9	ChImp should let the user choose between using only the last change or all changes for the calculation of primitive and composite measures.
R10	ChImp should be responsive.

5.1. The view of ChImp

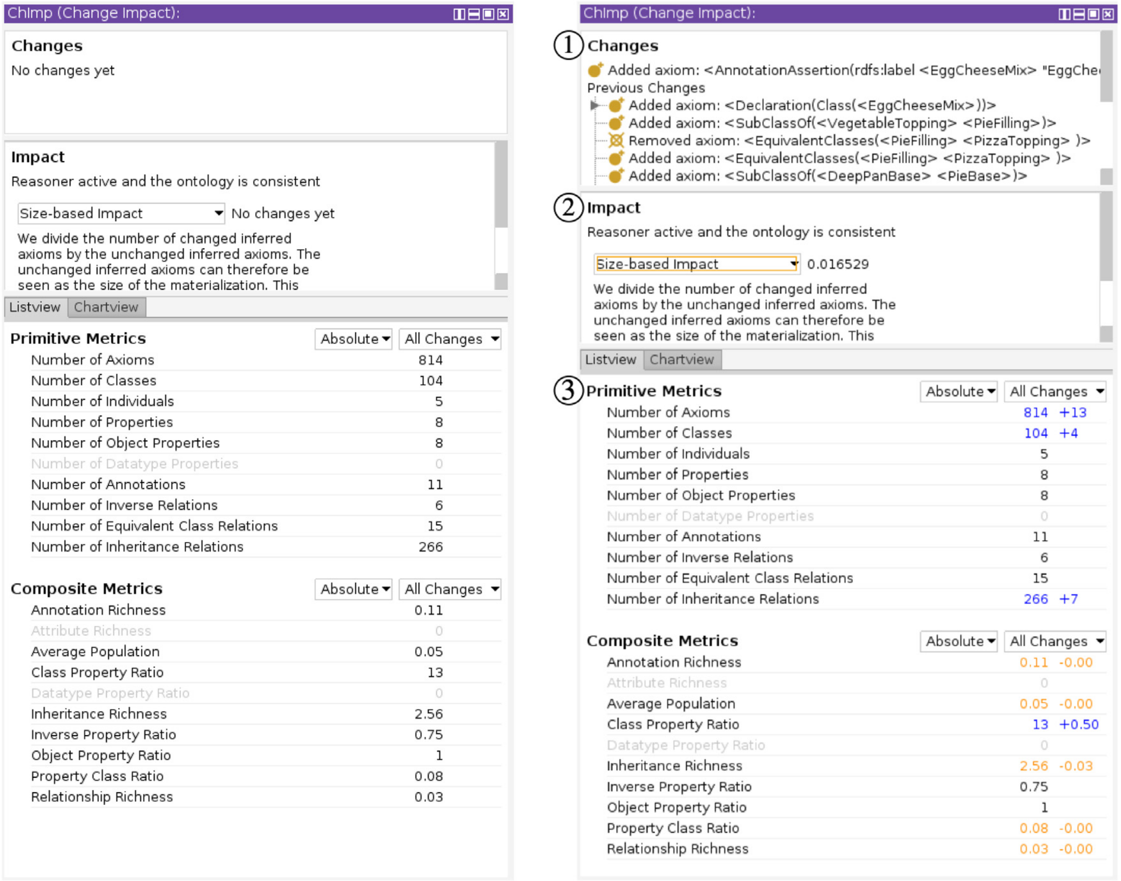
We designed ChImp based on the requirements R1–R10. ChImp is a Protégé plug-in implemented as a *view component*, which is a building block for workspace tabs. Any tab can display ChImp, which provides with three views: ① Change Display, ② Impact Display, and ③ Metrics Table, as shown in Fig. 4.

The Change Display (①) is split into two parts: the *last change* and *previous changes*. The former, situated at the top right, reports the most recent change, e.g., deletion of a class, and all the consequent automatic changes executed by Protégé, e.g., deletion of type axioms for individuals of the removed class. The latter, below, lists all the previous changes performed in the current session. The grouping remains the same as within the *last change* part. When the engineer applies a new change, the last change is updated and the former one is pushed into the list of previous changes. This display acts as a stack and addresses **R1**.

To address **R2**, ChImp reports the consistency status in the Impact Display (②). ChImp uses the internal reasoner, in this case HermiT [37], to check for consistency. Consistency is not automatically checked, but has to be synchronised using Protégé's reasoning menu. This display will alert the user if the reasoner has not been started. It also includes the materialisation impact measures, which we introduced in Section 3. They are listed again in Table 2 at the bottom. We operationalise the impact measures σ and γ by comparing respective values for both the first version of the ontology which was loaded into Protégé ($i = 0$) and the current snapshot.

The Metrics Table (③) has two parts, primitive and composite metrics, to address **R3**. The current version of ChImp shows the metrics listed in Table 2. The top section of the table explains primitive measures. For each of them, Table 2 reports the Protégé methods we used to retrieve respective values. The composite measures, in the middle part, are combinations of primitive measures and capture structural aspects of the ontology. We opted for only five measures, since some participants commented that more metrics do not necessarily provide additional information. ChImp uses colours to display the number and the delta when a measure is affected by the changes and, therefore, satisfies **R4**.

[3] https://gitlab.ifi.uzh.ch/DDIS-Public/chimp-protege-plugin.



ChImp (Change Impact):

Changes

No changes yet

Impact

Reasoner active and the ontology is consistent

Size-based Impact ▾ No changes yet

We divide the number of changed inferred axioms by the unchanged inferred axioms. The unchanged inferred axioms can therefore be seen as the size of the materialization. This

Listview | Chartview

Primitive Metrics	Absolute ▾ All Changes ▾
Number of Axioms	814
Number of Classes	104
Number of Individuals	5
Number of Properties	8
Number of Object Properties	8
Number of Datatype Properties	0
Number of Annotations	11
Number of Inverse Relations	6
Number of Equivalent Class Relations	15
Number of Inheritance Relations	266

Composite Metrics	Absolute ▾ All Changes ▾
Annotation Richness	0.11
Attribute Richness	0
Average Population	0.05
Class Property Ratio	13
Datatype Property Ratio	0
Inheritance Richness	2.56
Inverse Property Ratio	0.75
Object Property Ratio	1
Property Class Ratio	0.08
Relationship Richness	0.03

ChImp (Change Impact):

① **Changes**

Added axiom: <AnnotationAssertion(rdfs:label <EggCheeseMix> "EggChe

Previous Changes

- Added axiom: <Declaration(Class(<EggCheeseMix>))>
- Added axiom: <SubClassOf(<VegetableTopping> <PieFilling>)>
- Removed axiom: <EquivalentClasses(<PieFilling> <PizzaTopping>)>
- Added axiom: <EquivalentClasses(<PieFilling> <PizzaTopping>)>
- Added axiom: <SubClassOf(<DeepPanBase> <PieBase>)>

② **Impact**

Reasoner active and the ontology is consistent

Size-based Impact ▾ 0.016529

We divide the number of changed inferred axioms by the unchanged inferred axioms. The unchanged inferred axioms can therefore be seen as the size of the materialization. This

Listview | Chartview

③ Primitive Metrics	Absolute ▾	All Changes ▾
Number of Axioms	814	+13
Number of Classes	104	+4
Number of Individuals	5	
Number of Properties	8	
Number of Object Properties	8	
Number of Datatype Properties	0	
Number of Annotations	11	
Number of Inverse Relations	6	
Number of Equivalent Class Relations	15	
Number of Inheritance Relations	266	+7

Composite Metrics	Absolute ▾	All Changes ▾
Annotation Richness	0.11	-0.00
Attribute Richness	0	
Average Population	0.05	-0.00
Class Property Ratio	13	+0.50
Datatype Property Ratio	0	
Inheritance Richness	2.56	-0.03
Inverse Property Ratio	0.75	
Object Property Ratio	1	
Property Class Ratio	0.08	-0.00
Relationship Richness	0.03	-0.00

Fig. 4. Screenshots of ChImp, loaded in the ontology overview tab, after execution of some changes.

Table 2
Description and implementation of the primitive and composite metrics in ChImp. o is the instance of the ontology within each metric implementation.

	Description	Implementation
c	Number of classes	`o.getClassesInSignature().size()`
i	Number of individuals	`o.getIndividualsInSignature().size()`

p	Number of properties	`o.getObjectPropertiesInSignature().size()`
		`+ o.getDataPropertiesInSignature().size()`

h	Number of subclasses	`o.getAxioms(AxiomType.SUBCLASS_OF).size()`
a	Number of annotations	`o.getAnnotations().size()`

inv	Number of inverse relations	`o.getAxioms(AxiomType.INVERSE_FUNCTIONAL_OBJECT_PROPERTY).size()`
		`+ o.getAxioms(AxiomType.INVERSE_OBJECT_PROPERTIES).size()`

	Average population	$^{i}/_{c}$ [38–40]
	Inheritance richness	$^{h}/_{c}$ [40–42]
	Annotation richness	$^{a}/_{c}$ [38,40]
	Property class ratio	$^{p}/_{c}$ [38–40,43]
	Inverse property ratio	$^{inv}/_{p}$ [39–41]
$\sigma_{i,j}$	Size-based materialisation impact	$^{\|\Delta_{i,j}\|}/_{\|M_i \cap M_j\|}$ [24]
$\gamma_{i,j}$	Change-based materialisation impact	$^{\|\Delta_{i,j}\|}/_{\|\delta_{i,j}\|}$ [24]

To fulfil **R8**, the user can choose to display the change in metrics using absolute numbers or percentages also through a drop-down menu. Moreover, the user can access either the last change or all changes, according to **R9**. The former only shows the difference in metrics for the last change, while the latter is cumulative and displays the changes in metrics since the start of the session.

Even though we did not formulate a requirement, we also implemented a simple line chart which shows the change of the metrics. It is available as a tab in the Metrics Table (③) and it offers a drop-down to select the metric to be displayed. The y- and x-axis adjust automatically as editing progresses. A screenshot can be found in Fig. A.12 in the Appendix.

To export the information visible in ChImp, we implemented a simple "copy-to-clipboard" method for each of the panels. A screenshot of this functionality is available in Fig. A.13. A right-click into each of the panels triggers a menu where one can select the copy-to-clipboard option. The Change panel (①) offers a simple list of all changes without grouping. This implementation satisfies **R5**. The Impact panel (②) exports the status of the reasoner. If the reasoner is initialised and therefore also consistent, the export includes the impact metrics in a CSV-like format with a header after the reasoner status. If the reasoner is out of sync or inconsistent, the measures are not exported. This functionality satisfies **R7**. Lastly, the Metrics panel (③) exports all displayed standard and ratio metrics into a CSV format with a

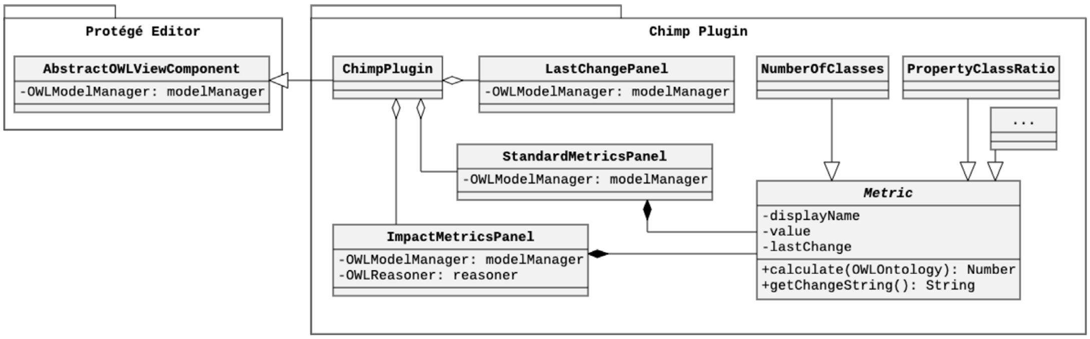


Fig. 5. UML Class diagram of the ChImp implementation, showing its architecture.

header. Hence, we also implemented **R6**. The header of the impact metrics and the standard metrics is the same and includes the initial value, new value, change in absolute numbers and change in percentage. All values are reported for the standard metrics. Some exported values are constant, e.g., the initial value of the impact metric (0.0) as there is no impact when there are no changes applied. The Listing A.1 in the Appendix A shows an export examples for each display.

5.2. Architecture

The plug-in is implemented for and based on Protégé 5.1. The application consists of two main parts: The display panels and the calculation logic of the metrics. All metrics – primitive, composite, and impact – extend the abstract `Metric` class that enforces the implementation of the method `calculateMetric()`. This abstraction functions as a strategy and enables an implementation independent interaction with the individual metrics. Additionally, it implicitly enforces private fields by requiring constructor arguments that define the name and description of the metric. Fig. 5 shows a diagram of the application.

The main class is `ChimpPlugin`. It extends `AbstractOWLViewComponent` so that it can be displayed as a view component in the Protégé editor. This relationship allows access to the `OWLModelManager` and the internal ontology.

Three individual classes, one for each panel, implement the user interface. The `LastChangePanel` holds all implementation of displaying and managing the change stack. The `ImpactMetricsPanel` takes care of the reasoner and consistency checking. If impact metrics were available, this panel would instantiate and display them. Primitive and composite metrics hold all information in their respective implementation of the metric interface. Therefore, the `StandardMetricsPanel` only needs to create instances of the metrics for the display. As the ontology is changed, each panel within the interface is updated by using a change listener on the `OWLModelManager`. This means that the plug-in can react to all change events fired by the main Protégé editor. Within the class `ChimpPlugin`, there is also a change listener that listens to changing ontologies. It is configured to reload the plug-in if the user switches the ontology.

The `OWLModelManager` also enables access to a reasoner if one is loaded in the Protégé editor. All reasoner plug-ins implement the `OWLReasoner` interface provided by the OWL API [44], and can therefore be used interchangeably. However, since their individual implementations and capabilities differ widely, they vary regarding results as well as performance. The impact panel leverages such a reasoner if it is available to determine consistency and calculate the impact measures.

For ChImp to track changes, the user needs to load the plug-in into Protégé and open the view once. After that, it starts recording the applied changes and displaying the changes, even if it is not in focus or visible. Protégé's change listener is used for this purpose.

5.3. Comparison of plugins

In addition to the *Basic Ontology Metrics* and *Ontology Diff* erence tool, which Protégé offers out of the box, Table 3 shows a comprehensive comparison of available plugins and tools. The Basic Ontology Metrics are equivalent to what we previously introduced as primitive metrics.

We divide the plugins into three distinct groups: plugins that calculate differences between ontologies, plugins that track changes, and others.

Ontology diff plugins. *OWLDiff* [28] and *LogDiffViz* [27] are meant for comparison of ontologies. It follows that their functionality is different from what ChImp aims for. The visualisation of *OWLDiff* [28] consists of a list of differences. This practical tool serves for the comparison of ontologies rather than investigating changes. It allows loading a second ontology and also offers the option of merging the detected differences into the original ontology. The Logical Difference Visualiser (*LogDiffViz*) [27] is noteworthy in terms of its capabilities for comparing ontology versions. Unfortunately, this plug-in does not update the visualisation based on changes applied during the Protégé session. One can only compare two ontologies, both stored in files, and visualise the differences afterwards. *OwlDiff* and *LogDiffViz* have not been kept up to date and are no longer available. However, Protégé [8] also comes with a build in ontology comparison, so there is no need for ChImp to have this capability.

Change tracking plugins. *Change-Analysis* [29], *Change View* [30], and *Changes Tab* [32] track and list changes. The *Change-Analysis* plugin is an addition to the Change Management Plugin used by Falconer et al. [29] and enables the exploration of changes and annotations using different aspects, such as authors or terms. It provides a browsing functionality. *Changes Tab* [32] is the only plugin among the three that provides an export functionality of the tracked changes. Similarly, *Change View* [30] is the only plugin of this group which is still available and compatible with the latest Protégé release. However, it does not have a browsing functionality, as it only tracks and lists changes. ChImp should also be capable of tracking and displaying changes (recall the requirement R1), however, this part of the plugin could easily be substituted by the *Change View* [30], which is available by default in Protégé 5. Therefore, in a future implementation, we want to enable the user to choose if they want the list of changes displayed or not, to prevent double listing of changes, if an engineer is used to work with the *Change View* [30] already. Since Protégé has been integrating often-used plugins, we hope they will integrate ChImp in the future or even merge ChImp with *Change View* [30]. Another plugin called *Change Capturing* [31] provides tracking of changes, also across multiple collaborators. However, this plugin is not part of Protégé but is based on the *NeOn Toolkit* editor [45], and therefore, not as relevant for this research.

Table 3
Comparison of plugins and tools which deal with ontology changes and visualisations.

	Ontology Diff				Change Tracking			Other		
	Protégé Ontology Metrics [8]	Protégé Ontology Differences [8]	OWLDiff [28]	LogDiffViz [27]	Change-Analysis [29]	Change View [30]	Changes Tab [32]	TDDonto2 [33]	Inference Inspector [16]	ChImp
R1: List of changes		✓	✓	✓	✓	✓	✓	✓		✓
Visualisation or browsing of changes			✓	✓	✓					
R2: Consistency	(✓)							✓	✓	✓
R2: Materialisation impact										✓
Visualisation or browsing of effects	(✓)							✓	✓	
R3: Ontology metrics	✓									✓
R3: Change in metrics										✓
R4: Colours			✓	✓		✓		✓	✓	✓
R5-7: Export functionality	✓						✓			✓
R8-9: Choice of presentation										✓
R10: Ad-hoc calculations	✓				✓	✓	✓	✓	✓	✓
Last update	2019	2019	2010	2011	2010	2010	2008	2016	2018	2021
Licence	BSD	BSD	LGPL v2			LGPL	MPL	LGPL v3		ASL 2.0
Availability	✓	✓				✓		✓	✓	✓

Other plugins. The last group of plugins includes *TDDonto2* [33] and *Inference Inspector* [16]. We already introduced them in Section 2 because they are examples of tools evaluated through user studies. *TDDonto2* [33] provides the engineer with an interface to ease the authoring or changing of an ontology. Its goal is to display the status of user defined tests, and not to provide an overview of changes and its consequences. The plugin updates the status of tests upon reasoner synchronisation. Therefore, this plugin shows consequences of changes on the reasoning and consistency of the ontology based on the entered tests. The *Inference Inspector* [16] plugin, as the name suggests, is an aid to browse the materialisation and how it changes based on the applied changes to the ontology. This is a very useful plugin to resolve inconsistencies and to gain detailed insights about the materialisation consequences of the edits. These objectives of the last two tools [16,33] are orthogonal to ChImp ones, as they provide details and allow browsing of the consequences of changes. ChImp's goal is to summarise this information and to give the ontology engineer an overview and an understanding of the consequences of changes as a whole.

To summarise, based on requirements we introduced in Section 4, we developed ChImp, which we then compared with comparable plugins with regard to the requirements. Having established its adherence to the requirements and also answering ***RQ1***, we now turn to ***RQ2***–***RQ4*** to not only investigate the understanding of change effects and the usefulness of materialisation impact measures, but also to evaluate ChImp empirically.

6. Impact understanding study

We experimentally evaluated ChImp with the following two goals: First, we assess the expert ontology engineers' understanding of the effect of their change edits and study if this understanding can be improved with the ChImp Protégé plugin. Second, we investigate the usefulness of impact measures, which signal the effect on the materialisation. In contrast to our previous evaluation [24], where we only assessed the impact measures by applying them to ontologies and investigating the impact of evolution as such, this section investigates how these measures are perceived by ontology engineers. This section introduces the impact understanding study design, chosen approach for the analysis of the collected responses, as well as presents and discusses the results.

6.1. Study design and experimental procedure

We designed our experiment to address the two research questions following a software benchmark approach [46]. We chose a within-subject design because it requires fewer participants than a between-subject design. We required specific expertise in our subjects, which has narrowed our participants pool significantly. Further, not many experts have time to participate in an hour-long study. Paired with complications of the COVID-19 situation, we chose to conduct the experiment as a "survey", where participants would install ChImp locally, solve it at a time of their choosing, and interrupt certain elements when needed. The survey which guides participants through the impact understanding study is available in the supplemental material with this submission.

The procedure of our impact understanding study can be found in Fig. 6. It is comprised out of an introduction, two tasks (called *Task 1* and *Task 2*), questions about the task difficulty and learning effect (called *Task Comparison*), questions about the impact measures (*Impact*), and finally a few closing questions (*Closing*).[4]

Introduction. The introduction includes an explanation of the impact understanding study, a privacy disclaimer, and requests some basic demographic information. We collect information about users' experience with Protégé and ontologies, as well as their education. We did not ask the participants about their age or gender, because they did not seem relevant to our research questions. If a participant did not agree to the data collection or did not fulfil the requirements, they could not proceed with the impact understanding study, and we screened them out at this point.

Tasks. Following this methodology, each participant had to solve two ontology editing tasks, called Vegan Pizza and Savoury Pie. In one of the tasks, participants were required to use ChImp. Both the order of the tasks and the availability of ChImp were randomised, leading to four different groups of participants, as shown in Fig. 6 (G1–G4). Participants in *G1* and *G2* solved the first task with ChImp and the second task without it. *G1* subjects first solved Vegan Pizza and then Savoury Pie, and *G2* solved the

[4] All the questions from the impact understanding study can be accessed at: https://bit.ly/3myzXQm.

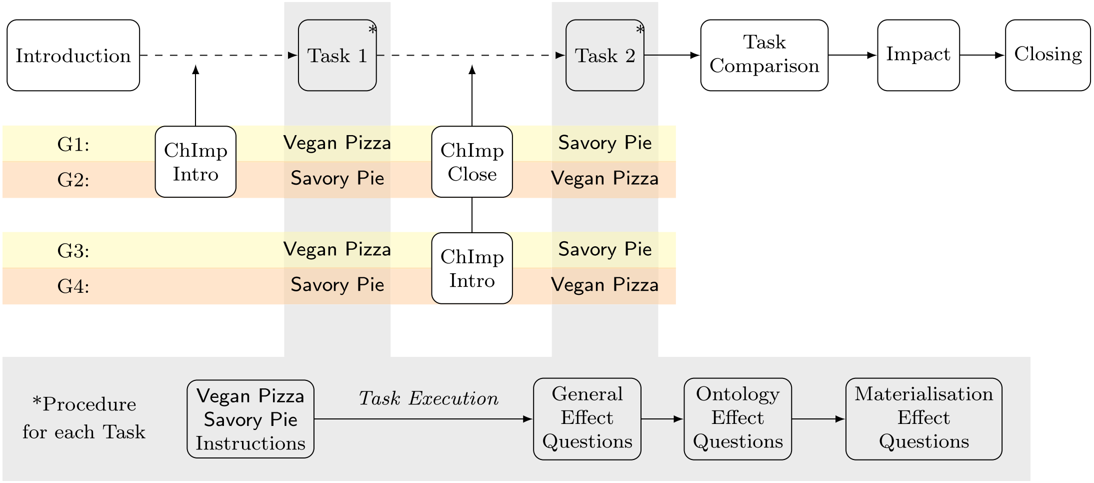


Fig. 6. Study design with four groups (G1–G4) with the assigned tasks and ChImp order and question blocks, to show the overall workflow of the impact understanding study.

Table 4
Task instructions as presented to the participants during the impact understanding study.

Vegan Pizza Task	Savoury Pie Task
• Add *VeganBase* as a subclass of *PizzaBase*, including disjointness to other *PizzaBase classes*. • Remove *VegetableTopping* as subclass from *CheesyVegetableTopping*. • Create the *Marinara* pizza, as subclass of *NamedPizza* consisting of a *VeganBase*, *SlicedTomatoTopping*, *Overtopping*, *GarlicTopping*, and *CaperTopping*. • Define *VeganPizza* as a subclass of Pizza with a *VeganBase* and only *VegetableTopping*: add a Subclass of statement "*hasBase some VeganBase*" and also add a equivalent-to statement "*Pizza and (hasTopping only VegetarianTopping)*".	• Create the class *SavouryPie* as subclass of *Food*, and its components *PieBase* and *PieFilling* also subclasses of *Food*. • Change the range and domain of the object properties *hasBase* and *hasTopping* to include as domain *SavouryPie* (Domain: "*Pizza or SavouryPie*") and as range *PieBase/PieFilling* (Range: "*PizzaBase or PieBase*", "*PizzaTopping or PieFilling*"). • A *SavouryPie* is also a superclass of "*hasBase some PieBase*", same as for Pizza. • Add *PieBase* as superclass of *DeepPanBase* (which is already a subclass of *PizzaBase*). • Add *EggCheeseMix*, as a subclass of *PieFilling*. • Add *PieFilling* as superclass of *VegetableTopping*.

tasks in reverse order. Members of *G3* and *G4* solved the first task without ChImp and the second with, whilst alternating the task's content. This *within subjects design* helps to compare the effect of ChImp whilst balancing out learning and ordering effects as well as possible unwanted interaction effects between the task and the subject's performance.

Chimp Intro/Close. Before executing the task with ChImp, each participant got a general introduction to the plugin (denoted as "ChImp Intro"). If participants were part of *G1* and *G2*, they were asked to close ChImp (denoted as "ChImp Close") before proceeding to the second task.

Task instructions. We designed the two tasks Vegan Pizza and Savoury Pie to be simple with some logics and simple additions to ensure they could be solved within an experimental session. We pretested both tasks with four experts to ensure that the directions are understandable and equally challenging. We present the tasks instructions in Table 4. Our selection of the tasks was solely based on covering as many type of changes which at the same time are not too complex in their execution but still have an influence on the materialisation.

Task questions. Subjects were asked three groups of questions after completing a task: general effect questions, ontology effect questions, and materialisation effect questions. This is represented by the lower part of Fig. 6 with a grey background. The general part included six questions, one of which was a multi-option single response question for seven statements such as "*The executed changes did not affect the class hierarchy of the ontology*". We chose a 4-point Likert scale. In addition, we provided the option of "Don't know" if they did not want to answer. The remaining five questions were open questions. For ontology and materialisation effect questions, we asked about the changes in numbers, followed by a severity rating and an open question to explain what severity meant for the participant. For the tasks performed without ChImp, participants answered an additional question on how they estimated or calculated the values.

Tasks comparison. We asked about the difficulty of the tasks themselves and how the tasks compared to each other. Further, we inquired whether participants perceived a difference between Vegan Pizza and Savoury Pie by answering questions about them. We asked if the difference was due to the second task being easier from a learning perspective, or if it was due to have ChImp at their disposal.

Impact. We directly asked about the perceived usefulness of the two impact measures presented in the plugin to the participant. This section of the impact understanding study also included further explanations of the plugin. The participants were encouraged to share their ideas and give feedback on the measures directly.

Closing. In the *Closing* part of the impact understanding study, we asked participants to provide overall feedback on the study and plugin.

After completion, subjects were redirected to a separate questionnaire concerning compensation. We compensated each participant who completed the impact understanding study with an Amazon voucher worth USD 20.

6.2. Approach

Ontology engineers are often domain experts of specific domains, hence, choosing any given domain would have severely restricted the subject pool. By using the Pizza ontology, we did not restrict the domain of our participants. Additionally, using predefined tasks is also a way of levelling the playing field and shortening the impact understanding study, because participants do not have to figure out what we meant with our free text instructions.

We evaluated the impact understanding study responses quantitatively, using the execution time and response ratings where applicable. To compare and assess the effect on the understanding caused by the ChImp plugin, we first assessed the time it took for the participants to execute the tasks and answer the questions. Time is measured by the study tool, Qualtrics.[5] It tracks the time spend on the displayed page. Therefore, by grouping together certain questions also within blocks, we are able to measure the time specifically for a set of questions, e.g., the general impact questions or task execution. This time measurement does not evaluate the performance of the tool itself, because the measuring is part of the study rather than part of ChImp itself. We also do not use this time to evaluate the performance of ChImp but instead use it to assess how participants interact with Protégé with and without ChImp. This is not a definitive measure, because task execution and question answering time is very individual to each of the participants and their level of expertise.

The more important part of the analysis is the qualitative approach. We used the open questions but also the single-choice questions to gain understanding of the thought process of the participants. This provided more insights into the understanding of change effects and how answers differ between the tasks performed with and without ChImp, respectively. However, due to the small subject pool, we consider the qualitative analysis as the contribution of this online user-study.

6.3. Results and discussion

We recruited 101 subjects by sharing the impact understanding study with the authors' network via email and also openly sharing it via Twitter. 15 were screened out before proceeding to the introduction questions, leaving 86 participants that reached the introduction. Of the 15 screened out participants, ten had never used Protégé or edited ontologies before. One participant did not agree to the data collection. The remaining four did not proceed and did not answer the required question on the introduction of the impact understanding study. Out of those 86 participants, 62 effectively provided answers in the introduction, and only 53 proceeded to the first task. Between the first and second task, we lost 16 additional participants. Participants who completed both tasks also completed the remainder of the impact understanding study. As is visible in Table 5, we had to disregard some participants answers. Seven responses were deleted because they only took below 1 min for either completing the task or answering the change effect questions. Two participants noted that Protégé crashed during the task and could not answer the questions correctly. Six participants used ChImp for both tasks, and their answers had to be excluded. This last case led to an imbalance between usable responses.

Fig. 7 shows the participants' demographics and their corresponding years of experience with ontologies and Protégé. The figure indicates that professional seniority correlates with experience.

Table 5
Number of participants per impact understanding study step and experimental condition.

	Intro.	Task1	Task2	Comp.	Impact	Closing
Recorded	67	53	37	37	37	37
Usable	62	36	25	25	36	36

6.3.1. Task difficulty and learning effect

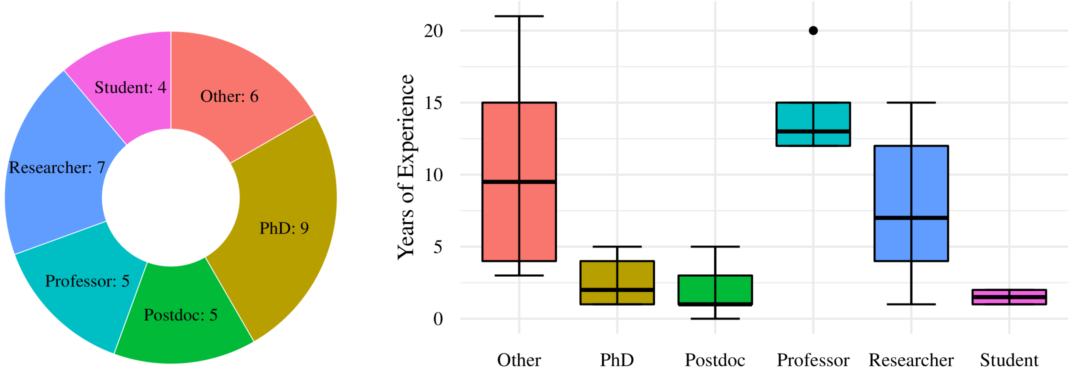
The participants disagreed on the difficulty of the tasks and the relative difficulty between tasks. Rating difficulty on a scale between 1 and 5, five being the most difficult, the mean (m) rated difficulty was 2.56 with a standard deviation (sd) of 1.11.[6] The tasks were found neither easy nor difficult, with some exceptions. However, no participant found them to be extremely difficult (i.e., rate 5) and seven participants found them extremely easy (i.e., rate 1), from the 36 participants that completed the entire impact understanding study. Fig. 8(a) shows task difficulty in red and question difficulty in blue. Vegan Pizza was considered more difficult by twelve participants, and Savoury Pie by five participants. 19 participants felt the difficulty of the tasks was the same. 16 participants thought that the questions for Vegan Pizza were harder. 15 participants thought they could answer questions about Vegan Pizza with more ease than those about Savoury Pie, and five participants thought they were the same.

The self-reported learning effect and effect of Chimp are visualised in Fig. 8(b). Twelve participants indicated that they experienced a learning effect from one task to the second, and twelve did not. Twelve claimed the question did not apply to them because they did not experience a difference in difficulty between the two tasks. Learning effect is shown in Fig. 8(b). The average learning effect is calculated based on the scale shown in the figure and results in $m = -0.125$ ($sd = 1.484$). Given that the average is below 0, do not confirm a learning effect based on the self-assessment. Therefore, as a next step, we consider the time participants took to execute the tasks and answer questions in lieu of the self-assessment, shown in Table 6. As a first step, we identified outliers computationally, and removed them from the dataset. Table 6 clearly shows that in general, the second task was executed and answered quicker than the first. Especially for the question answering, we think this is because of participating needing less time for reading and understanding of the questions. This strengthens the finding of the self-assessment that a learning effect is present.

We checked for the normality of the distributions of each of the groups using the Wilcox–Shapiro test, which we report in Table 7, before moving on to test the different effects. Because not all groups show a normal distribution, we conducted multiple analyses. First we executed the repeated-measures ANOVA to identify the variables (usage of ChImp, task number, task type) which have an effect on the task execution and question answering time. Even though the normality assumption is violated by some data, it does not completely invalidate the results, as some robustness is provided. For the task execution, no variable yielded a significant effect (ChImp: $p = 0.605$, number: $p = 0.164$, type: $p = 0.352$). For the question answering, the task number was significant (ChImp: $p = 0.353$, number: $p = 0.007$, type: $p = 0.294$). This result strongly points towards the presence of a learning effect, at least in the context of answering questions about the tasks. Hence, we exclude the second task and execute a two-way ANOVA as well, only for the first task. However, the results remain unchanged, and no significant effect has been detected for the task execution (ChImp: $p = 0.290$, type: $p = 0.338$)

[5] https://www.qualtrics.org/.

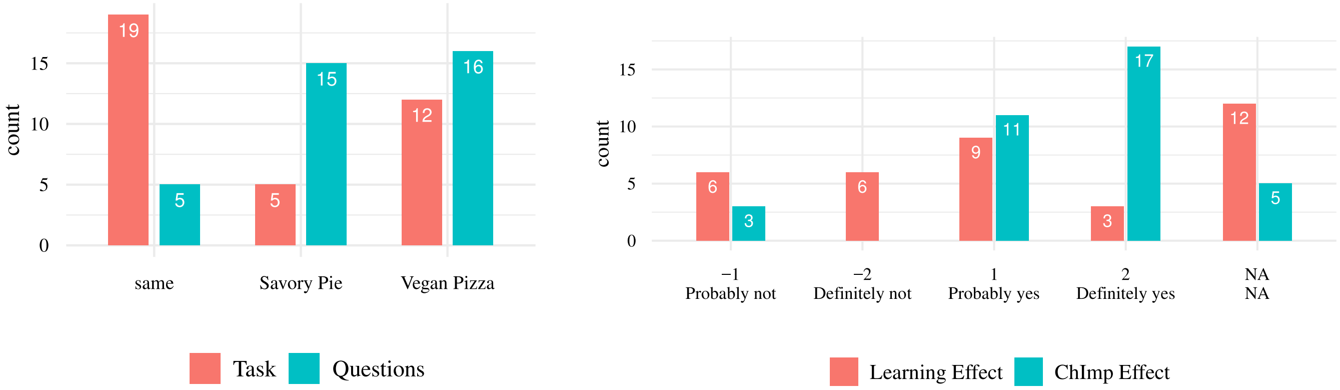
[6] We report statistics by APA standards [47], cf. https://my.ilstu.edu/~jhkahn/apastats.html.



(a) Profession of participants.

(b) Experience in years for each participant group.

Fig. 7. Participant demographics and the corresponding years of experience.



(a) Which task and questions were more difficult to execute and answer?

(b) Learning and ChImp effect.

Fig. 8. Self-assessment and difficulty. The *y*-axis is the count of participants which selected the answers shown on the *x*-axis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Number of participants per group (G1–G4) for both tasks and average time in minutes (and standard deviation) needed for task execution and question answering. Usage of ChImp highlighted in grey. Removed outliers.

	Task 1			Task 2		
		Task	Questions		Task	Questions
G1	5	11.9797 (2.3846)	17.3749 (6.4189)	4	11.6829 (5.9852)	8.4886 (3.0977)
G2	13	11.9360 (6.4952)	10.8785 (4.9896)	7	11.9926 (10.4217)	9.9650 (8.7230)
G3	7	12.2777 (5.8201)	17.0664 (11.5654)	6	6.2074 (1.9383)	9.7874 (7.9436)
G4	11	8.9928 (3.7472)	13.7868 (9.8104)	8	7.9673 (2.2376)	6.4657 (3.6371)

or the question answering (ChImp: $p = 0.501$, type: $p = 0.115$). Because of the non-normal distribution of for some groups, we also executed the Kruskal–Willis test, which does not require a normal distribution, but is not meant for repeated measures. Therefore, we only analyse the first task and use group numbers instead of ChImp and task type to determine the interaction, hence yielding only one value. This second test also did not yield any significant effects (task execution: $p = 0.411$, question answering: 0.355).

6.3.2. Understanding of change effects

We now investigate the understanding of change effects and whether ChImp raised said understanding during the impact understanding study. In the comparison section of the impact understanding study, we asked participants if they *thought* ChImp was the reason, why they could answer the questions for one task with more ease. The results ($m = 1.355$, $sd = 0.915$), as shown in Fig. 8(b), indicate that participants appreciated the help of the plugin as an aid to answer questions about change effects.

Table 7
Shapiro test for normality for each group's first task. We report the p-value, and significant results mean that the distribution is significantly different from a normal distribution.

Group	Task 1	Questions 1	Task 2	Questions 2
G1	0.4598	0.1075	**0.0351**	0.8939
G2	0.2442	**0.0079**	**0.0008**	**0.0015**
G3	**0.0012**	0.3349	0.9761	**0.0184**
G4	0.1740	0.2484	0.1227	0.1745

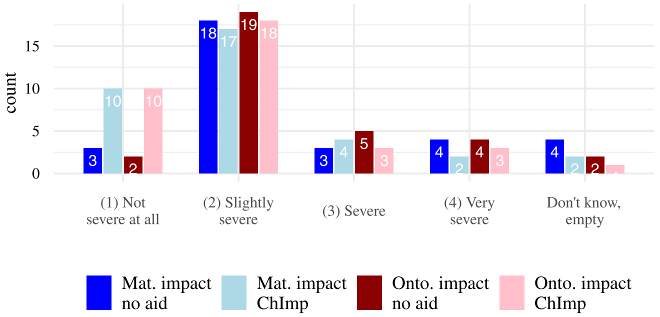


Fig. 9. Severity rating of impact on ontology and materialisation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Given the comparatively smaller reliability of a self-assessment, we analyse the effect on time, understanding, and satisfaction as well.

Effect on time. One way of assessing whether the application of ChImp was beneficial for the participants is to consider the time it took to answer questions about the effect of changes. The respective values are presented in Table 6, where the time is reported for task execution and question answering. Even though, we did not find a significant effect using ANOVA and the Kruskal–Willis test, we can still see a difference of means in Table 6. Task and Questions are executed faster with ChImp than without compared between groups, not within, except for two cases, the execution of the first task for G4 and the answering of the task 2 questions for G3. However, the statistical test is inconclusive, because with a p value of $p = 0.34$ for task execution and $p = 0.41$ for question answering, we cannot reject the null-hypothesis, and therefore, cannot assume that there is a statistically significant difference in the means. However, given that participants could have potentially interrupted measurements of time, we need to disregard the quantitative analysis and rely on the open questions and their qualitative analysis. The participants saw a benefit in using ChImp while answering questions about consequences, as shown with the self-assessment in Fig. 8.

Effect on understanding. As described above, the self assessment results were highly favourable towards the usage of ChImp (see Fig. 8; $m = 1.355$, $sd = 0.915$). Note that the self-assessment could have been biased, since we forced an opinion with the 4-point Likert scale (whilst offering the option "Not applicable"). Participants tend to choose the "nicer" answer, when no middle ground is possible. Looking at the severity rating of the effect of the applied changes in Fig. 9, we observed that participants rate the effect on the ontology (red, pink) and materialisation (dark and light blue) lower when they have ChImp at their disposal. This is visible by the ten participants that rated the effect as not severe at all when using ChImp (in pink and light blue) compared to only three or two participants without ChImp . This could be the result of the information and insights ChImp provides. We expected a low rating on severity for both tasks, as the tasks are not invasive for the ontology and do not have disruptive changes. Lastly, for the question "Will this change how you think about changes in the future?" participants could choose between "Yes", "No", and "Maybe". Fig. 10(b) reports the answers to the above question. Additionally, responses to the last question shown in Fig. 10(b) are relevant, as we can see that participants do think about consequences for other applications as well and regard it thus as relevant to their work.

Satisfaction with ChImp. Overall, the participants were satisfied with ChImp and 24 participants also show interest in continuing to use it in the future as reported in Fig. 10(b) in the right most questions. When asked about understanding of changes during "Task Comparison", the results indicate that 14 participants (about a third of the total) do not often think about the effect of changes, shown in Fig. 10(a). Given a choice from (1) to (4), (1) being equal to "*This is the first time I thought about the effect of change*" and (4) representing "*Most of the time, I consider the effect of the changes.*", we recorded an average of $m = 2.08$ ($sd = 1.079$). Together with the answers shown in Fig. 10(b), we can conclude that participants' understanding was increased. Given that many participants want to use ChImp in the future, we infer that we achieved our goal of raising the understanding of ontology change effects.

In summary, we confirm ***RQ2***: *Do ontology engineers understand the effect of changes on the ontology and on the materialisation better when using ChImp than without?* The results point towards an increased understanding of change effects with ChImp given our qualitative analysis. Our quantitative analysis neither confirmed nor disproved our claim of raising understanding, because of the online and interrupted execution of the impact understanding study.

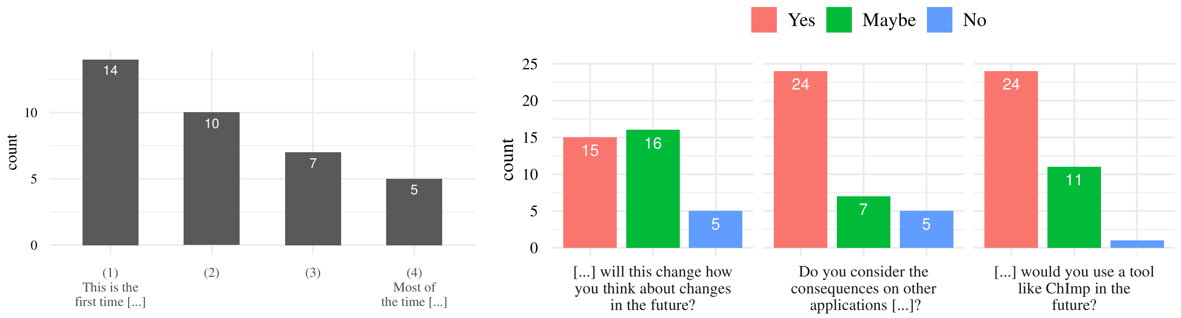
6.3.3. *Meaning of severity of impact*

We asked participants about what severe impact on the ontology and on the materialisation means to them. These were two separate questions in the impact understanding study, each asked after participants assessed the severity of the changes on either the ontology or the materialisation. The questions asked for open text responses, and we analysed their content in detail, categorising the answers into different topics. Some participants left the questions unanswered or left a non-related comment, e.g., "Protégé/Reasoner crashed nothing was displayed in the ChImp plugin". Hence, we analysed the answers of 34 participants, as these were the valid responses. We summarised the responses in Table 8.

We found that 20 participants did not distinguish between the impact on the ontology and the impact on the materialisation. They either stated that for them, it is the same, gave the exact same answer or only provided an answer to one of the questions. For four participants, it was hard to decide if their answers mean something different or not, and ten participants explicitly stated something different in these two questions.

Nine participants thought that the consistency is the best indicator for severity of impact. Seven different participants stated that the impact on the ontology can be measured with the amount of changes in relation to the size of the ontology (30% or 50% were stated explicitly). Therefore, we found that participants either stated reasoning or the number of changes to the ontology as severity of impact on the ontology. Eight participants also explicitly mentioned the structure of the ontology as the determinant for severity of impact, only one of these eight also stating the number of changes as a severity indicator.

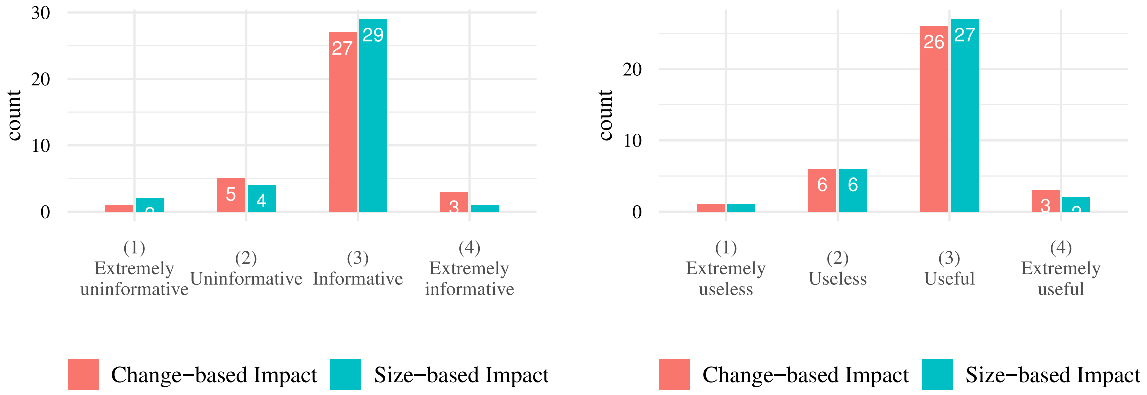
16 participants (out of 34) stated that the severe impact on the materialisation should be determined by the amount of changes in the materialisation. Out of these 16 participants, five explicitly mentioned our proposed measures. The remaining 11 participants described the measures but did not refer to σ or γ explicitly. Three participants mentioned specific numbers for the severity of impact, e.g., 30% and 50%, which are the same as of the amount of ontology change signalling severe impact. We further found that four participants stated that reasoning and the number of change on the materialisation should be used as indicator for severity. There are four participants who explicitly mentioned reasoning and the consistency of an ontology as being the sole



(a) ... was this the first time you considered the consequences of changes, or does this reflect how you work regularly?

(b) Change effect understanding questions.

Fig. 10. Answers of participants about impact of changes, severity and participants' understanding of it previously and in the future. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Informativeness

(b) Usefulness

Fig. 11. Distribution of ratings for the impact measures.

indicator of severity. As severity indicator for impact, the number of changes to the materialisation was mentioned more often (16 out of 19) than the changes to the ontology (seven out of 19) and four participants mentioned both, changes to the materialisation and ontology.

Nine participants mentioned different types of impact, such as errors to existing downstream calculations and competency questions. We are in agreement with the participants, a sentiment, with which we concur. These answers also show that indeed the impact on the downstream task is of importance and that each task requires its own task-specific impact measures.

There is not a straightforward answer to ***RQ3***: *What does severe impact on the ontology and on the materialisation mean to ontology engineers?* As mentioned above, there are different opinions present among our participants. However, the severity of impact on the materialisation should be determined based on the amounts of changes according to 16 out of 32 participants. A threshold like 0.3 or 0.5 for σ (size-based impact), as suggested by the impact understanding study participants, could be used to signal a clear message to the engineer that the impact has reached a critical value, and hence, the recalculation of the materialisation will be necessary for other applications. However, the threshold is individual and application dependent. Therefore, we encourage ontology engineers to share the σ and γ values with ontology users, so that they can decide for themselves how to proceed with the new version of the ontology.

Table 8
Summary of the answers about severe impact. Only the top row adds up to 34, which is the number of answers we analysed. Participants could give multiple answers for both ontology and materialisation severity of impact.

	Yes	No	Unclear
Distinction between impact on ontology and materialisation	10	20	4
Impact on	Onto.	Mat.	
Reasoning as impact (consistency of ontology)	9		
Number of changed axioms	7	16	
Explicitly mentioned:			
Impact on structure	8		
Our impact measures		5	
Errors, change in underlying definitions of concepts or others	9		

6.3.4. Assessment of impact measures

Generally, the impact measures were both overwhelmingly rated useful and informative (see Fig. 11). Participants had to choose a rating between (1) and (4), with (4) being the highest, either very useful or very informative, also shown in the figure. For the size-based impact, this resulted in an average usefulness rating of $m = 2.833$ ($sd = 0.561$) and informativeness rating of $m = 2.806$ ($sd = 0.577$). The change-based impact was rated slightly higher in usefulness with $m = 2.861$ ($sd = 0.593$) and also in informativeness with $m = 2.889$ ($sd = 0.575$). Two participants pointed out they liked the measures, but additional information is needed for a better understanding of the impact. Other four participants commented that the measures are not useful for them in their daily work, and they would not use them. Further, two participants noted that the change-based impact was less intuitive than size-based impact.

One comment of a participant about the diminished utility of the metrics struck us as very important. The participant pointed out that the metric is useless if the ontology becomes inconsistent. Inconsistency is a severe impact on the materialisation, but will not be reflected by either of the impact measures we introduced. We agree with the assessment, and would like to point out that ontology consistency could be seen as a binary impact measure in itself. It does not quantify the change to the materialisation, like the size-based and change-based impact measures, but it is by no means less important. However, our goal was to assess the impact measures we proposed, and the consistency of ontology is not a novel indicator of impact. Additionally, the Impact Display in ChImp shows already both the consistency of the ontology and an impact measure.

Given the reported scores and the participants' general comments, we give a positive answer to ***RQ4***: *Are the materialisation impact measures useful and informative for ontology engineers?*

6.3.5. Other applications and impact indications

We asked what other type of applications/tasks/operations the participants would consider when changing an ontology and how these might be impacted by the applied changes. We found one mention of embeddings, natural language processing tasks and annotations each. The documentation of the ontology and the data used upon the ontology were both mentioned twice, and mappings. Applications, which have a user interface and use the ontology and reasoning in the background, were also mentioned three times. Three participants mentioned SPARQL and/or SHACL queries. Lastly, four participants disclosed that they think about impact on data mappings and tagged data.

Further, we opened the floor for suggestions of other type of impact measures on the materialisation, which would be relevant or interesting to the participants. We received interesting suggestions, which we would like to mention here as well. It was pointed out by four participants that they would like more detailed information than just numbers. We agree with this sentiment, but would like to mention that the Inference Inspector [16] already satisfies this need. One participant suggested indicating the number of inconsistent classes as impact. Another participants would like to see a breakdown per types of axioms in the impact measures. In the same direction, three participants suggested having a specific impact measure for ABox entailment. These suggestions highlight the positive response and willingness to think about the effect of changes, especially on the materialisation. Lastly, two participants suggested a more visual breakdown of the impact on the materialisation. We will add these various improvements into ChImp in the future where possible.

7. Limitations and future work

Our impact understanding study comes with some limitations. Given the choice of within-subject design, there is the potential of a learning effect taking place since every participant solved two tasks with the same ontology. We minimised this effect as far as possible with randomisation of the order of tasks and ChImp, but were not able to eliminate it with this approach. Our statistical analysis, even though inconclusive, indicated that there seems to be an effect on question answering time based on the task number. We anticipate this to be the case, because participants answered the exact same questions twice, hence needing less time to read and understand the questions.

This impact understanding study had to be conducted remotely because of the COVID-19 pandemic. We did not supervise our participants and could not guarantee that they completed the tasks and impact understanding study without interruptions. Analysing the overall time it took participants to finish the impact understanding study without eliminating outliers, we found a mean time of $m = 8.14$ h, with a standard deviation of $sd = 25.74$ h. Due to the agreed upon privacy statement, we were not able to record the sessions or observe the participants in completing the impact understanding study and also are not able to share the collected data of our participants.

Further, we received 36 complete responses in our impact understanding study and about 20% of our participants had limited experience. We disregarded free-text answers from less experienced participants, especially when they stated that they had limited expertise and could not answer some question. Additionally, choosing a toy example as a basis for our task has advantages as well as limitations. Using the pizza ontology and prescribed tasks allowed us to reach a wider pool of participants. However, at the same time, the generalisability to other ontologies and real world scenarios is hampered due to the simplicity of the task. Hence, we propose to study the influence of changes as well as the understanding of the influence further with focus on specific domains, like the biomedical domain. Furthermore, it would enable us to use a realistic task that would be more representative for Protége's real-world usage.

As future work, we are planning a second experiment, during which participants will be closely supervised either in person or remotely. We will use two different ontologies to further minimise learning effects between the conditions and to increase the external validity. Given a supervised setting, we can limit the number of asked questions since we will be able to rely on the time as indicators. A follow-up study of this kind will allow us to validate the results found in this exploratory impact understanding study. Additionally, it will be of interest to know if ChImp shows a more significant benefit for more complex and larger ontologies since we already see a benefit for simple ontologies, such as the Pizza Ontology. Lastly, a further research might show differences due to the experience of the ontology engineer.

Since ChImp was developed with the idea in mind of aiding ontology engineers who use Protégé, the plugin has a large limitation in terms of performance. ChImp can only be used within Protégé and hence, could never support the editing of large popular knowledge graphs such as DBpedia. However, with this impact understanding study, we have shown that the information provided by ChImp is beneficial to the ontology engineer. As the editing of DBpedia or Wikidata occurs on their corresponding platforms, we do not envision ChImp being used with them directly, but would encourage such information as is provided in ChImp to be also implemented for these platforms in the future.

Furthermore, we will continue developing and improving the ChImp plugin, especially considering the feedback received from the impact understanding study participants. One aspired additional feature is to provide a summary of changes and visualise it interactively with the impact. This would allow for more detailed information about the impact on the materialisation and a better overview of the changes. Also addressing the evolution effects on the ABox specifically, as well as imports [21] and mappings [12]. We plan to implement the impact measures from related work in ChImp as future work. Finally, we want to provide the ontology engineer with the option to import previously executed changes. This would allow for display and summary of changes beyond the current Protégé session.

8. Conclusions

As ontologies become more central to the Web, more people edit, maintain and use ontologies daily. Due to the inevitable change to knowledge and, consequently, to the ontologies which model it, ontology users need to update to new versions regularly. However, there is a communication gap between ontology engineers and users, as updates to ontologies happen often.

With the requirements survey, we asked practitioners about their opinion and preferences on visualising changes within Protégé. We formulated ten requirements and were able to implement six of them directly. The ChImp plug-in is the result of this implementation. **R10** addresses responsiveness. Even though, we used only Protégé-native calls and did not make use of additional libraries, responsiveness requires a separate evaluation. As future work, we will evaluate our implementation in a test environment with different ontologies and various change types and sizes. Further, we conducted a hands-on impact understanding study to investigate ontology engineers' understanding of change effects. The impact understanding study included two editing tasks on the Pizza ontology and various questions on the effects of the executed changes. We presented two materialisation measures and asked participants about their perceived usefulness.

We used a within-subject study design and randomised the order of tasks as well as for which of the two ChImp was used. This design allowed us to minimise the possible order effect. It also contributed to minimising the transfer and learning effects across the conditions. In our qualitative analysis, we found that participants were more aware of change consequences and could more easily answer the respective questions when they had ChImp at their disposal. Therefore, we conclude that ChImp increased their understanding (***RQ2***). Additionally, most participants would use ChImp again for a similar task in their daily activities. The meaning of severity of impact on ontology and materialisation varying among the participants. 16 out of 34 participants, who provided valid answers, mentioned either reasoning/consistency (9) or the number of changes on the ontology (7) as the indicator for severity of impact on the ontology. Further, 16 out of 34 participants, found that the number of changes on the materialisation are the indicator of severity for them ***RQ3***. Further, most participants found both impact measures useful and informative ***RQ4***, which shows a benefit for ontology engineers and will also benefit ontology users in the future.

Our results indicate that raising the understanding about ontology change effects (e.g., with ChImp and the presented impact measures) is possible and desirable. In the future, we plan to improve ChImp and also confirm our qualitative findings with quantitatively by conducting a second, more controlled study. Thus, our ongoing research with ChImp and Protégé [8] is a first step towards improving communication and collaboration between ontology engineers and users, helping to facilitate a more stable Semantic Web.

CRediT authorship contribution statement

Romana Pernisch: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Visualization, Validation, Data curation. **Daniele Dell'Aglio:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Mirko Serbak:** Software. **Rafael S. Gonçalves:** Methodology, Software. **Abraham Bernstein:** Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

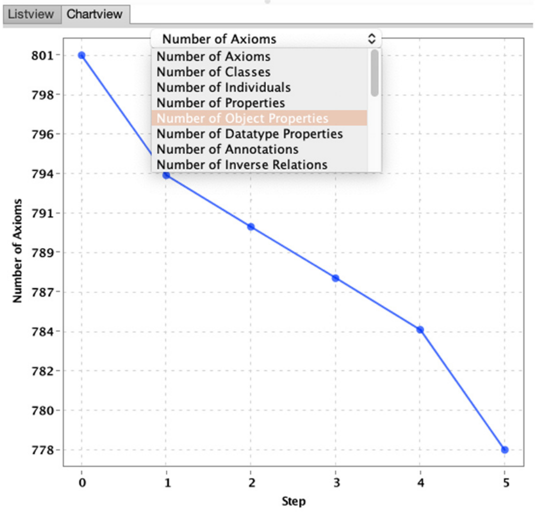
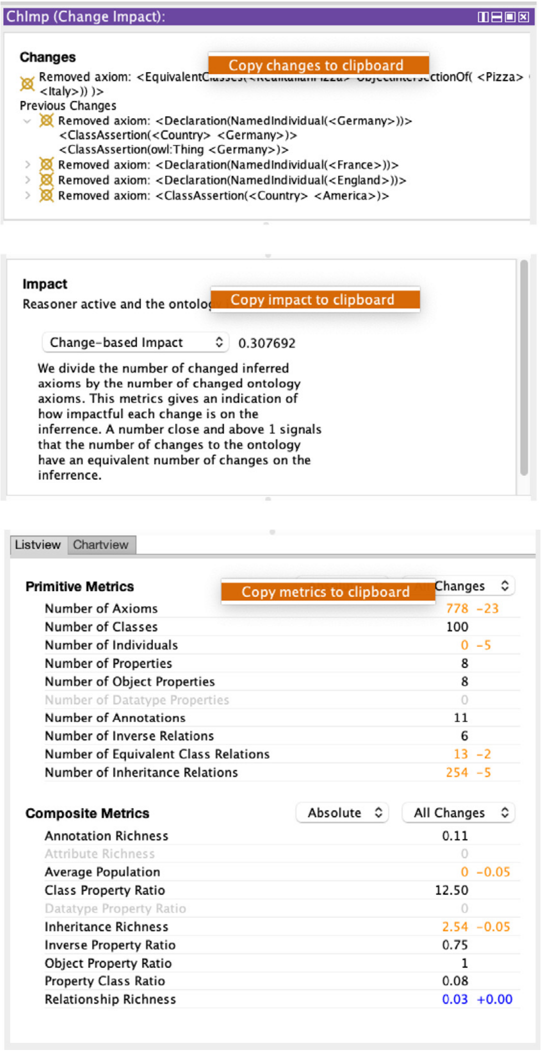


Fig. A.12. Screenshot of the chart tab in the Metrics Display.


ChImp (Change Impact):

Changes

Copy changes to clipboard

☒ Removed axiom: <EquivalentC[illegible]ctionOf(<Pizza>
<Italy>)))>

Previous Changes

☒ Removed axiom: <Declaration(NamedIndividual(<Germany>))>
<ClassAssertion(<Country> <Germany>)>
<ClassAssertion(owl:Thing <Germany>)>

☒ Removed axiom: <Declaration(NamedIndividual(<France>))>

☒ Removed axiom: <Declaration(NamedIndividual(<England>))>

☒ Removed axiom: <ClassAssertion(<Country> <America>)>

Impact

Reasoner active and the ontolog

Copy impact to clipboard

Change-based Impact 0.307692

We divide the number of changed inferred axioms by the number of changed ontology axioms. This metrics gives an indication of how impactful each change is on the inferrence. A number close and above 1 signals that the number of changes to the ontology have an equivalent number of changes on the inferrence.

Listview Chartview

Primitive Metrics	Changes
Number of Axioms	778 −23
Number of Classes	100
Number of Individuals	0 −5
Number of Properties	8
Number of Object Properties	8
Number of Datatype Properties	0
Number of Annotations	11
Number of Inverse Relations	6
Number of Equivalent Class Relations	13 −2
Number of Inheritance Relations	254 −5

Copy metrics to clipboard

Composite Metrics	Absolute	All Changes
Annotation Richness		0.11
Attribute Richness		0
Average Population		0 −0.05
Class Property Ratio		12.50
Datatype Property Ratio		0
Inheritance Richness		2.54 −0.05
Inverse Property Ratio		0.75
Object Property Ratio		1
Property Class Ratio		0.08
Relationship Richness		0.03 +0.00

Fig. A.13. Screenshot of the export functionality for each display.

```
List of changes (chronologically, first to last): 23 changes recorded
RemoveAxiom(ClassAssertion(<http://www.co-ode.org/ontologies/pizza/pizza.owl#Country> <http://www.co-ode.org/ontologies/pizza/
        pizza.owl#America>) OntologyID(OntologyID(OntologyIRI(<http://www.co-ode.org/ontologies/pizza>) VersionIRI(<http://www.co-ode
        .org/ontologies/pizza/2.0.0>))))
RemoveAxiom(ClassAssertion(owl:Thing <http://www.co-ode.org/ontologies/pizza/pizza.owl#America>) OntologyID(OntologyID(OntologyIRI
        (<http://www.co-ode.org/ontologies/pizza>) VersionIRI(<http://www.co-ode.org/ontologies/pizza/2.0.0>))))
RemoveAxiom(EquivalentClasses(<http://www.co-ode.org/ontologies/pizza/pizza.owl#Country> ObjectIntersectionOf(<http://www.co-ode.
        org/ontologies/pizza/pizza.owl#DomainConcept> ObjectOneOf(<http://www.co-ode.org/ontologies/pizza/pizza.owl#America> <http://
        www.co-ode.org/ontologies/pizza/pizza.owl#England> <http://www.co-ode.org/ontologies/pizza/pizza.owl#France> <http://www.co-
        ode.org/ontologies/pizza/pizza.owl#Germany> <http://www.co-ode.org/ontologies/pizza/pizza.owl#Italy>)) )
...

Reasoner status: INITIALIZED and ontology consistent

Metric,FirstValue,NewValue,ChangeAbs,ChangePercentage
Size-based Impact,0.0,0.035087719298245 61,0.0350877192982456 1,0.0
Size Hierarchy Impact,0.0,0.036036036036036036,0.036036036036036036,0.0
Change-based Impact,0.0,0.3076923076923077,0.3076923076923077,0.0
Change Hierarchy Impact,0.0,1.3333333333333333,1.3333333333333333,0.0
Change Noise Impact,0.0,0.0,0.0,0.0

Metric,FirstValue,NewValue,ChangeAbs,ChangePercentage
Number of Axioms,801,778,-23.0,-2.871410736579276
Number of Classes,100,100,0.0,0.0
Number of Individuals,5,0,-5.0,-100.0
Number of Properties,8,8,0.0,0.0
Number of Object Properties,8,8,0.0,0.0
Number of Datatype Properties,0,0,0.0,0.0
Number of Annotations,11,11,0.0,0.0
Number of Inverse Relations,6,6,0.0,0.0
Number of Equivalent Class Relations,15,13,-2.0,-13.333333333333334
Number of Inheritance Relations,259,254,-5.0,-1.9305019305019304
Annotation Richness,0.11,0.11,0.0,0.0
Attribute Richness,0.0,0.0,0.0,0.0
Average Population,0.05,0.0,-0.05,-100.0
Class Property Ratio,12.5,12.5,0.0,0.0
Datatype Property Ratio,0.0,0.0,0.0,0.0
Inheritance Richness,2.59,2.54,-0.04999999999999982,-1.9305019305019238
Inverse Property Ratio,0.75,0.75,0.0,0.0
Object Property Ratio,1.0,1.0,0.0,0.0
Property Class Ratio,0.08,0.08,0.0,0.0
Relationship Richness,0.0299625468164794,0.030534351145038167,5.718043285587657E-4,1.9083969465648805
```

Listing A.1: Example export of each part of ChImp, starting with the list of changes, followed by the impact display and the standard metrics as last.

Acknowledgements

We thank the Swiss National Science Foundation for their partial support under contract number #407550_167177. This project was also partially funded by Elsevier's Discovery Lab.

Appendix A. Additional screenshots of ChImp and export example

See Figs. A.12 and A.13 and also Listing A.1.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.websem.2022.100715.

The supplementary material indludes two documents: the questionaire used in the requirements elicitaion and the survey used in the user study.

References

[1] Fouad Zablith, Grigoris Antoniou, Mathieu d'Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, Marta Sabou, Ontology evolution: a process-centric survey, Knowl. Eng. Rev. 30 (1) (2015) 45–75, http://dx.doi.org/10.1017/S0269888913000349.

[2] Ljiljana Stojanovic, Alexander Maedche, Boris Motik, Nenad Stojanovic, User-driven ontology evolution management, in: Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW), Vol. 2473, Springer, 2002, pp. 285–300, http://dx.doi.org/10.1007/3-540-45810-7_27.

[3] Tim Berners-Lee, James Hendler, Ora Lassila, The semantic web, Sci. Am. 284 (5) (2001) 34–43.

[4] Konstantin Schekotihin, Patrick Rodler, Wolfgang Schmid, Matthew Horridge, Tania Tudorache, Test-driven ontology development in protégé, in: ICBO, in: CEUR workshop proceedings, vol. 2285, CEUR-WS.org, 2018.

[5] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, Jens Lehmann, Lc-QuAD 2.0: A large dataset for complex question answering over wikidata and DBpedia, in: ISWC (2), in: Lecture Notes in Computer Science, vol. 11779, Springer, 2019, pp. 69–78, http://dx.doi.org/10.1007/978-3-030-30796-7_5.

[6] Denny Vrandecic, Markus Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (10) (2014) 78–85, http://dx.doi.org/10.1145/2629489.

[7] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, Christian Bizer, DBpedia - A Large-scale, multilingual knowledge base extracted from wikipedia, Semant. Web 6 (2) (2015) 167–195, http://dx.doi.org/10.3233/SW-140134.

[8] Mark A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (4) (2015) 4–12, http://dx.doi.org/10.1145/2757001.2757003.

[9] Romana Pernisch, Mirko Serbak, Daniele Dell' Aglio, Abraham Bernstein, ChImp: VIsualizing ontology changes and their impact in protégé, in: Proceedings of the Visualization and Interaction for Ontologies and Linked Data, Co-Located with ISWC 2020, CEUR-WS.org, 2020.

[10] Anika Groß, Michael Hartung, Kay Prüfer, Janet Kelso, Erhard Rahm, Impact of ontology evolution on functional analyses, Bioinformatics 28 (20) (2012) 2671–2677, http://dx.doi.org/10.1093/bioinformatics/bts498.

[11] Thomas Gottron, Christian Gottron, Perplexity of index models over evolving linked data, in: Proceedings of the European Semantic Web Conference (ESWC), 8465, Springer, 2014, pp. 161–175, http://dx.doi.org/10.1007/978-3-319-07443-6_12.

[12] Julio Cesar dos Reis, Cédric Pruski, Marcos Da Silveira, Chantal Reynaud-Delaître, Understanding semantic mapping evolution by observing changes in biomedical ontologies, J. Biomed. Inform. 47 (2014) 71–82, http://dx.doi.org/10.1016/j.jbi.2013.09.006.

[13] Silvio Domingos Cardoso, Cédric Pruski, Marcos Da Silveira, Ying-Chi Lin, Anika Groß, Erhard Rahm, Chantal Reynaud-Delaître, Leveraging the impact of ontology evolution on semantic annotations, in: Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings, in: Lecture Notes in Computer Science, 10024, 2016, pp. 68–82, http://dx.doi.org/10.1007/978-3-319-49004-5_5.

[14] Markel Vigo, Samantha Bail, Caroline Jay, Robert Stevens, Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design, Int. J. Hum.-Comput. Stud. 72 (12) (2014) 835–845, http://dx.doi.org/10.1016/j.ijhcs.2014.07.005.

[15] Wa'el Mohsen, Mostafa Aref, Khaled ElBahnasy, Scaled scrum framework for cooperative domain ontology evolution, in: Proceedings of the International Conference on Frontiers of Educational Technologies (ICFET), ACM, 2020, pp. 135–143, http://dx.doi.org/10.1145/3404709.3404770.

[16] Nicolas Matentzoglu, Markel Vigo, Caroline Jay, Robert Stevens, Inference Inspector: Improving the verification of ontology authoring actions, J. Web Semant. 49 (2018) 1–15, http://dx.doi.org/10.1016/j.websem.2017.09.004.

[17] Pieter De Leenheer, Christophe Debruyne, DOGMA-MESS: A tool for fact-oriented collaborative ontology evolution, in: OTM Workshops, Vol. 5333, Springer, 2008, pp. 797–806, http://dx.doi.org/10.1007/978-3-540-88875-8_104.

[18] Ronald Denaux, Dhavalkumar Thakker, Vania Dimitrova, Anthony G. Cohn, Interactive semantic feedback for intuitive ontology authoring, in: Proceedings of Frontiers in Artificial Intelligence and Applications FOIS, 239, IOS Press, 2012, pp. 160–173, http://dx.doi.org/10.3233/978-1-61499-084-0-160.

[19] Natalya Fridman Noy, Michel C.A. Klein, Ontology evolution: Not the same as schema evolution, Knowl. Inf. Syst. 6 (4) (2004) 428–440, http://dx.doi.org/10.1007/s10115-003-0137-2.

[20] Rafael S. Gonçalves, Bijan Parsia, Ulrike Sattler, Categorising logical differences between OWL ontologies, in: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), ACM, 2011, pp. 1541–1546, http://dx.doi.org/10.1145/2063576.2063797.

[21] Omar Qawasmeh, Maxime Lefrançois, Antoine Zimmermann, Pierre Maret, Observing the impact and adaptation to the evolution of an imported ontology, in: IC3K 2019-11ht International Joint Conference on Knowledge Discovery, Knowledge Enginnering and Knowledge Management, 2019, p. 11p, http://dx.doi.org/10.5220/0008064700760086.

[22] Francesco Osborne, Enrico Motta, Pragmatic ontology evolution: Reconciling user requirements and application performance, in: Proceedings of the International Semantic Web Conference (ISWC), in: LNCS, vol. 11136, Springer, 2018, pp. 495–512, http://dx.doi.org/10.1007/978-3-030-00671-6_29.

[23] Romana Pernischová, Daniele Dell'Aglio, Matthew Horridge, Matthias Baumgartner, Abraham Bernstein, Toward predicting impact of changes in evolving knowledge graphs, in: Proceedings of the International Semantic Web Conference (ISWC) Satellites, in: CEUR workshop proceedings, 2456, CEUR-WS.org, 2019, pp. 137–140.

[24] Romana Pernisch, Daniele Dell'Aglio, Abraham Bernstein, Beware of the hierarchy - an analysis of ontology evolution and the materialisation impact for biomedical ontologies, J. Web Semant. (2021) http://dx.doi.org/10.1016/j.websem.2021.100658.

[25] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, Eugenia G. Giannopoulou, Ontology visualization methods - a survey, ACM Comput. Surv. 39 (4) (2007) 10, http://dx.doi.org/10.1145/1287620.1287621.

[26] Marek Dudás, Steffen Lohmann, Vojtech Svátek, Dmitry Pavlov, Ontology visualization methods and tools: a survey of the state of the art, Knowl. Eng. Rev. 33 (2018) e10, http://dx.doi.org/10.1017/S0269888918000073.

[27] William Gatens, Boris Konev, Michel Ludwig, Frank Wolter, Versioning based on logical difference for lightweight description logic terminologies, Proc. ARCOE (2011).

[28] Petr Kremen, Marek Smid, Zdenek Kouba, OWLDiff: A Practical tool for comparison and merge of OWL ontologies, in: DEXA Workshops, IEEE Computer Society, 2011, pp. 229–233, http://dx.doi.org/10.1109/DEXA.2011.62.

[29] Sean M. Falconer, Tania Tudorache, Natalya Fridman Noy, An analysis of collaborative patterns in large-scale ontology development projects, in: Proceedings of the International Conference on Knowledge Capture (K-CAP), ACM, 2011, pp. 25–32, http://dx.doi.org/10.1145/1999676.1999682.

[30] Nick Drummond, ChangeView, 2011, https://code.google.com/archive/p/co-ode-owl-plugins/wikis/ChangeView.wiki.

[31] Raul Palma, Change capturing, 2008, http://neon-toolkit.org/wiki/1.x/Change_Capturing.html.

[32] William Liu, Tania Tudorache, Timothy Redmond, Changes tab, 2008, https://protegewiki.stanford.edu/wiki/Changes_Tab.

[33] Kieren Davies, C. Maria Keet, Agnieszka Lawrynowicz, More effective ontology authoring with test-driven development and the TDDonto2 tool, Int. J. Artif. Intell. Tools 28 (07) (2019) 1950023:1–1950023:25, http://dx.doi.org/10.1142/S0218213019500234.

[34] Nicolas Matentzoglu, Markel Vigo, Caroline Jay, Robert Stevens, Making entailment set changes explicit improves the understanding of consequences of ontology authoring actions, in: Knowledge Engineering and Knowledge Management, Springer, 2016, pp. 432–446, http://dx.doi.org/10.1007/978-3-319-49004-5_28.

[35] Christian Alrabbaa, Franz Baader, Raimund Dachselt, Tamara Flemisch, Patrick Koopmann, Visualising proofs and the modular structure of ontologies to support ontology repair, in: Proceedings of the 33rd International Workshop on Description Logics (DL 2020) co-located with the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), in: CEUR Workshop Proceedings, 2663, CEUR-WS.org, 2020.

[36] Matthias Dehmer, Frank Emmert-Streib, Yongtang Shi, Interrelations of graph distance measures based on topological indices, PLoS One 9 (4) (2014) e94985, http://dx.doi.org/10.1371/journal.pone.0094985.

[37] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, Zhe Wang, HermiT: An OWL 2 reasoner, J. Automat. Reason. 53 (3) (2014) 245–269, http://dx.doi.org/10.1007/s10817-014-9305-1.

[38] Astrid Duque-Ramos, Jesualdo Tomás Fernández-Breis, Miguela Iniesta, Michel Dumontier, Mikel Egaña Aranguren, Stefan Schulz, Nathalie Aussenac-Gilles, Robert Stevens, Evaluation of the OQuaRE framework for ontology quality, Expert Syst. Appl. 40 (7) (2013) 2696–2703, http://dx.doi.org/10.1016/j.eswa.2012.11.004.

[39] Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, Jos Lehmann, Modelling ontology evaluation and validation, in: The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, in: Lecture Notes in Computer Science, 4011, Springer, 2006, pp. 140–154, http://dx.doi.org/10.1007/11762256_13.

[40] Samir Tartir, I. Budak Arpinar, Amit P. Sheth, Ontological evaluation and validation, in: Theory and Applications of Ontology: Computer Applications, Springer, 2010, pp. 115–130, http://dx.doi.org/10.1007/978-90-481-8847-5_5.

[41] Rim Djedidi, Marie-Aude Aufaure, ONTO-EVO A L an ontology evolution approach guided by pattern modeling and quality evaluation, in: International Symposium on Foundations of Information and Knowledge Systems, 2010, pp. 286–305, http://dx.doi.org/10.1007/978-3-642-11829-6_19.

[42] Birger Lantow, Kurt Sandkuhl, An analysis of applicability using quality metrics for ontologies on ontology design patterns, Intell. Syst. Account. Financ. Manage. 22 (1) (2015) 81–99, http://dx.doi.org/10.1002/isaf.1360.

[43] Christoph Tempich, Raphael Volz, Towards a benchmark for semantic web reasoners - an analysis of the daml ontology library, in: EON2003, Evaluation of Ontology-based Tools, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools held at the 2nd International Semantic Web Conference ISWC 2003, in: CEUR Workshop Proceedings, 87, CEUR-WS.org, 2003.

[44] Matthew Horridge, Sean Bechhofer, The OWL API: A java API for OWL ontologies, Semant. Web 2 (1) (2011) 11–21, http://dx.doi.org/10.3233/SW-2011-0025.

[45] Michael Erdmann, Walter Waterfeld, Overview of the NeOn toolkit, in: Ontology Engineering in a Networked World, Springer, 2012, pp. 281–301, http://dx.doi.org/10.1007/978-3-642-24794-1_13.

[46] Matthias Rauterberg, Benutzungs-orientierte benchmark-tests: eine methode zur benutzerbeteiligung bei standardsoftware-entwicklungen, in: Software Ergonomie (German Chapter of ACM), 1991, pp. 96–107, http://dx.doi.org/10.1007/978-3-322-94654-6_9.

[47] American Psychological Association, Publication Manual of the American Psychological Association, seventh ed., American Psychological Association, 2000.