ChImp: Visualizing Ontology Changes and their Impact in Protégé

Romana Pernisch[1][0000−0001−8590−1817], Mirko Serbak[1], Daniele Dell'Aglio[2,1][0000−0003−4904−2511], and Abraham Bernstein[1][0000−0002−0128−4602]

[1] Department of Informatics, University of Zurich, Zurich, Switzerland
`{pernisch,dellaglio,bernstein}@ifi.uzh.ch`, `mirko.serbak@uzh.ch`
[2] Department of Computer Science, Aalborg University, Aalborg, Denmark
`dade@cs.aau.dk`

Abstract. Today, ontologies are an established part of many applications and research. However, ontologies evolve over time, and ontology editors—engineers and domain experts—need to be aware of the consequences of changes while editing. Ontology editors might not be fully aware of how they are influencing consistency, quality, or the structure of the ontology, possibly causing applications to fail. To support editors and increase their sensitivity towards the consequences of their actions, we conducted a user survey to elicit preferences for representing changes, e.g., with ontology metrics such as number of classes and properties. Based on the survey, we developed ChImp—a Protégé plug-in to display information about the impact of changes in real-time. During editing of the ontology, ChImp lists the applied changes, checks and displays the consistency status, and reports measures describing the effect on the structure of the ontology. Akin to software IDEs and integrated testing approaches, we hope that displaying such metrics will help to improve ontology evolution processes in the long run.

Keywords: Ontology evolution · Evolution Impact · Change visualization · Protégé plugin.

1 Introduction

In recent years, ontologies and controlled vocabularies have gained popularity. They can be viewed as graphs that contain concepts and individuals, which are put into various relations with each other using properties. They can represent portions of the real world, e.g., products, people, cities, and emotions. Various communities invested time, money, and resources into building such graphs and using them they can improve search engines, recommenders, and research.

The management of ontologies has inspired several studies, which analyze different aspects of them, such as creation, usage, and contents. However, most studies either focus on the usage, disregarding its evolution, or vice versa. When studies only consider the usage of ontologies and how to improve upon existing applications, they often assume a static knowledge graph. This assumption does not correspond to reality since those graphs are continuously evolving, e.g., by adding new gene interactions or updating the composition of national governments over time. Only a small number of studies consider both usage and evolution at the same time [12, 20, 11, 22].

Ontology engineers are often unaware of all the consequences they cause while editing an ontology. Their main focus is on the knowledge being integrated. 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. Inexperienced engineers may lack the expertise to fully grasp all the consequences of their actions. Moreover, experienced editors are likely to work with ontologies they do not know well and might, therefore, be unaware of the effect of changes. We argue that editors need to be made aware of the effect of their changes using multiple perspectives, including the change's semantic and structural consequences while they are changing the ontology.

In this article, we propose *ChImp* (*Ch*ange *Imp*act), a Protégé [18] plug-in to display information related to the changes. ChImp's goal is to increase the understanding of ontology changes, making editors more aware of consequences and impact of their changes. To collect requirements for ChImp, we conducted a survey asking ontology editors about ontology evolution and its consequences. The survey contained mock-ups of change visualizations 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 a plug-in offering three perspectives on changes: a summary of the performed changes, the consistency of the ontology, and changes to ontology measures (such as number of classes or properties).

Therefore, our contributions are: (1) a list of requirements for visualizing and informing about changes and (2) the ChImp plug-in for Protégé providing real-time information to the ontology engineer about the applied changes.

In the next section, we present related research. Section 3 introduces the survey, shows its results, and summarizes the requirements for the plug-in. In Section 4, we present our implementation in detail. We address conclusions and future work in the last section.

2 Related Research

This section addresses two aspects of the related research. The first is research on the impact of ontology evolution, which is fairly young. As artefact evolution is a known phenomenon, we take a step further and address research on the consequences of their evolution. Second, we discuss visualizations of ontologies with a focus on change visualization. We also acknowledge the many published Protégé plug-ins dealing with visualization of differences between ontology versions. However, none of them focus on the visualization of changes applied during a users current Protégé session.

Evolution Impact. Noy and Klein [19] already made clear that ontology evolution is not the same as database schema evolution. They point out that evolution's consequences are difficult to foresee because of the decentralisation of ontologies. Gonçalves et al. [10] propose a categorization 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. [12] examine how changes in an ontology impact previously conducted functional analysis. Also Gottron and Gottron [11] investigate the impact of knowledge graph 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. Osborne et al. [20] 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. Pernischová et al. [22] investigate and predict the impact of knowledge graph evolution on embeddings by comparing neighbourhoods. Unfortunately, most of these approaches are too complex and calculation-intensive to be implemented as an interactive Protégé plug-in. Analog to Pernischová et al. [22] using embeddings, we will calculate the materialization and, therefore, check the consistency of the ontology. The consistency status is considered the impact of changes and can be determined after each edit.

Change Visualization. Katifori et al. [15] and Dudás et al. [4] are two important contributions to ontology visualization. The former covers a range of ontology visualization methods and techniques; it discusses the strengths and weaknesses of each method and addresses the issue of visualizing time-related data. The latter presents the current state of the art in ontology visualization. It states that there is no de-facto standard of visualization which has been accepted by the Semantic Web community, due to the fact that there is no single solution fits all applications.

Visualizations are not new in Protégé [18]. Nonetheless, none of the available plug-ins deal with the direct visualization of changes at editing time. Calculation of impact is also a recent research, and no plugin has addressed it so far. We specifically focus on plugins that visualizes the changes. ChangeAnalysisTab, addition to the Change Management Plugin used in Falconer et al. [6], enables the exploration of changes and annotations using different aspects, such as authors or terms. The Logical Difference Visualizer (LogDiffViz) is noteworthy in terms of its capabilities in comparing ontology versions [8]. Unfortunately, this plug-in does not update visualization based on changes applied during the Protégé session. One can only compare two ontologies and visualize the differences afterwards. Further, OWLDiff [16] does not provide much of a visualization at all. This practical tool serves the comparison and merging of ontologies rather than investigating changes. Lastly, Change View [3] provides a straightforward list of changes but does not visualize them beyond a simple grouping.

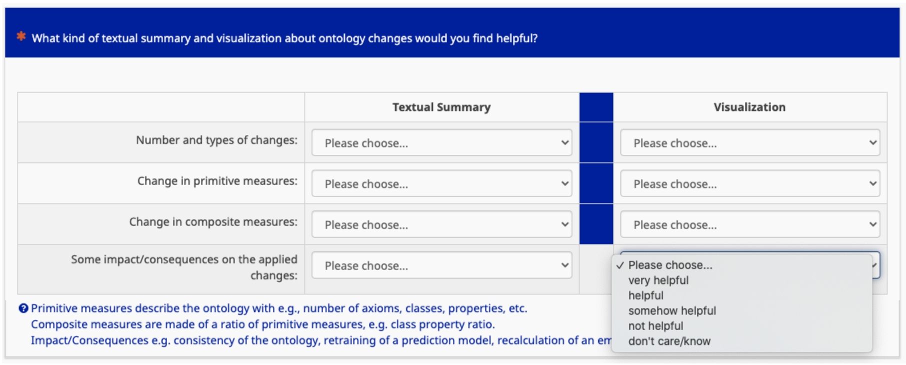
3 Requirements Survey

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 visualization through a survey. We first present our survey, focusing on three specific questions which we present below. Subsequently, we formulated the related requirements that drove the development of ChImp.

Survey structure. We conducted a survey, which consists of four main sections.[3] 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 inquiries about tools (plug-ins or visualizations) related to changes that participants may already use. This part of the 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 visualizing the changes and their impact. It presents mock-ups visualizing 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 survey and two in the third one.

Figure 1 shows the first question we analyze, 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 visualize them. The information about changes includes number and types of changes, a variation of primitive measures (e.g., numbers of classes, properties, or annotations), composite measures (e.g., class to property ratio or the number of annotations per class), and consequences of change (e.g., ontology consistency). We propose two visualization 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 visualization 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.

[3] We have published a static version of the survey on our web site at https://files.ifi.uzh.ch/ddis/chimp/reqsurvey.pdf


* 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...

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 survey "Changing an Ontology." Each drop-down shows the same selection options and each requires an answer.

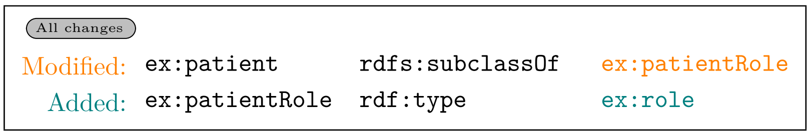
The second question is the *mock-up question*, where participants observed five mock-ups. Figure 2 shows five mock-ups, each showing a different aspect, such as impact, consistency, changes, and measures. The participants judge each mock-up with a score from 1 (not at all informative) to 5 (very informative).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

Participants Demographics. We invited semantic web practitioners to answer the survey. We distributed the survey 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 survey. The remaining eight did not complete the survey.

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. Only one participant works in engineering, the others in research, and they 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 visualize 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 compliance before a release. One participant uses an UML-like graphical representation of the changes to inform users about the new version.

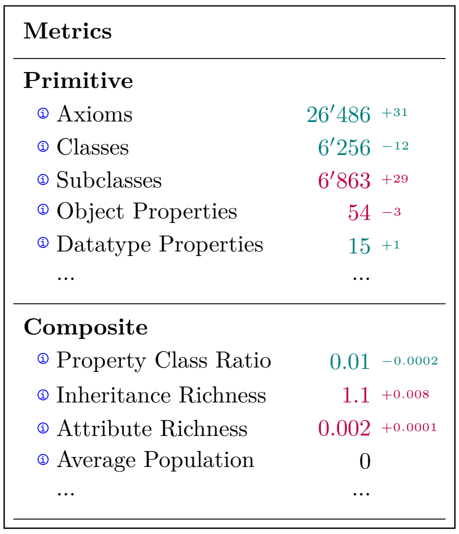
Survey analysis. We report the results of the three questions—helpfulness, mock-up, and plot—in Figure 3. Figure 3a shows the mean rating for the helpfulness question. We assign values between 0 and 4 to the answers of this ques-



All changes

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

(a) List of changes



Metrics

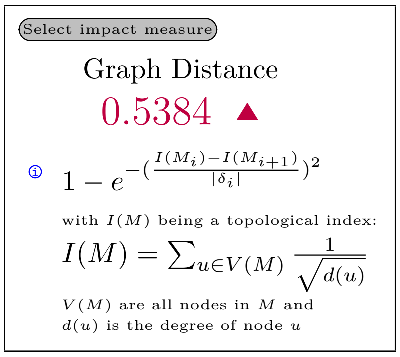
Primitive

ⓘ 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	
...	...	

(b) Table of measures



Select impact measure

Graph Distance

0.5384 ▲

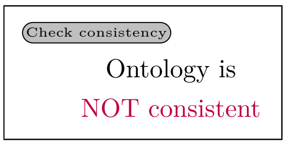
ⓘ $1 - e^{-\left(\frac{I(M_i)-I(M_{i+1})}{|\delta_i|}\right)^2}$

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

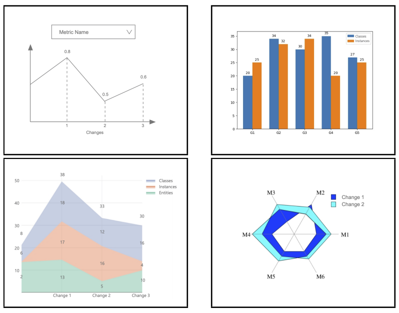
(d) Impact measure [1]



Check consistency

Ontology is NOT consistent

(c) Consistency

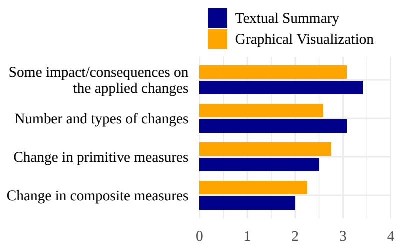


(e) Plots

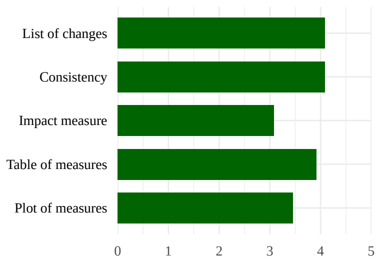
Fig. 2: Mock-ups used in the requirements survey. Participants rated each mock-up from 1 (not at all informative) to 5 (very informative).

tion. 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 visualizations. We observe the opposite situation for the primitive and composite measures; in these cases, participants prefer visualization over textual summaries. Due to the participants' preference for the number and types of changes, we formulate the following requirement:

[R1] *ChImp should list the applied changes.* ChImp will report all changes



(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 visualization. The option "Do not know/care" was rated as 0.



(b) Results of the mock-ups question where mock-ups of different visualizations were presented to the participants. They rated them from "very informative" (5) to "not informative" (1).

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

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. 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. Figure 3b 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 checked after every change. 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.

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 visualization. However, while answering the mock-up question, participants do not perceive the plots showing ontology measures as informative and favor the tabular visualization. Seamingly, 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 have decided to only use a table to visualize the change in numbers since this answer was rated higher in the mock-up question of the survey:

[R3] *ChImp should show primitive and composite measures in a table, visualizing the new value and its difference to the old value based on the applied changes.* Section 4 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 awareness, but the choice of colour is essential. Therefore, we formulate the following requirement:
[R4] *ChImp should use colors to indicate changes.* However, the choice of colour should not imply additional meaning, e.g., "good" or "bad". Also to accommodate colorblindness, we will avoid colours like red and green.

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 survey, participants elaborated on this with detailed comments. With biomedical ontologies, engineers might be interested in the impact 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 to include the number of other ontologies and systems which will be (specifically) affected by the changes.

In the open questions, survey participants also suggest 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 materialization*, as indication of consequences. As we explain in Section 4, **R5**, **R6** and **R7** are not available in the current version of ChImp, but they will be addressed in future work.

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 chose 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 between using only 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.

While engineers change the ontology, ChImp executes many calculations in the background. 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. Using Protégé and OWL native calls will help in achieving this requirement.

4 The ChImp Plug-in

ChImp is distributed under the Apache 2.0 license an can be be downloaded from the project web site.[4] 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 [18].

4.1 Plug-in Implementation

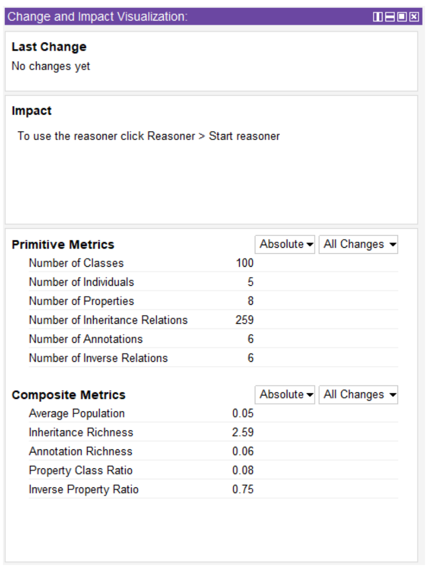
ChImp is a Protégé plug-in implemented as a view component, which is a building block for workspace tabs. Any tab can display ChImp, which provides with three views: Change Display, Impact Display, and Metrics Table, as shown in Figure 4.

The Change Display is split into two parts: the *last change* and *previous changes*. The former, situated right at the top, reports the most recent change, e.g., deletion of a class, and all the dependent 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 is retained 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. ChImp uses the internal reasoner, in this case HermiT [9], to check for consistency. Consistency is not checked automatically, but has to be synchronized using Protégé's reasoning menu. This display will also alert the user if the reasoner has not been started.

The Metrics Table has two parts, primitive and composite metrics, to address **R3**. The current version of ChImp shows the metrics listed in Table 1. The top section of the table explains primitive measures. For each of them, Table 1 reports the Protégé methods we used to retrieve respective values. The

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


Change and Impact Visualization:

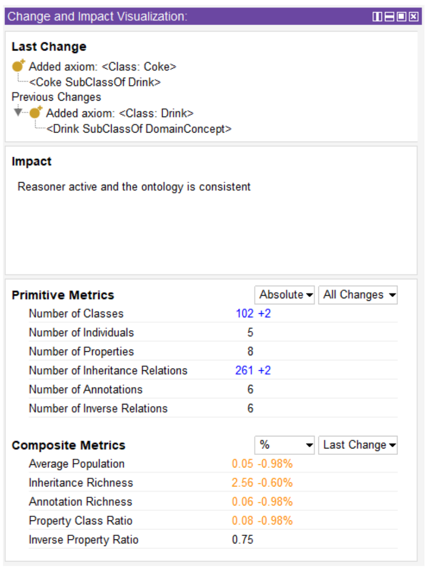
Last Change
No changes yet

Impact
To use the reasoner click Reasoner > Start reasoner

Primitive Metrics	Absolute / All Changes
Number of Classes	100
Number of Individuals	5
Number of Properties	8
Number of Inheritance Relations	259
Number of Annotations	6
Number of Inverse Relations	6

Composite Metrics	Absolute / All Changes
Average Population	0.05
Inheritance Richness	2.59
Annotation Richness	0.06
Property Class Ratio	0.08
Inverse Property Ratio	0.75

(a) ChImp after initialization.


Change and Impact Visualization:

Last Change
Added axiom: <Class: Coke>
<Coke SubClassOf Drink>
Previous Changes
Added axiom: <Class: Drink>
<Drink SubClassOf DomainConcept>

Impact
Reasoner active and the ontology is consistent

Primitive Metrics	Absolute / All Changes
Number of Classes	102 +2
Number of Individuals	5
Number of Properties	8
Number of Inheritance Relations	261 +2
Number of Annotations	6
Number of Inverse Relations	6

Composite Metrics	% / Last Change
Average Population	0.05 -0.98%
Inheritance Richness	2.56 -0.60%
Annotation Richness	0.06 -0.98%
Property Class Ratio	0.08 -0.98%
Inverse Property Ratio	0.75

(b) ChImp after starting the reasoner and executing some changes.

Fig. 4: Screenshots of ChImp, loaded in the ontology overview tab.

composite measures, in the bottom 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 colors to display the number and the delta when a measure is affected by the changes and, therefore, satisfies **R4**. To fulfil **R8**, the user can chose 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.

4.2 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. Figure 5 shows a diagram of the application.

	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	$^{i}/_{c}$ [5, 7, 23]
	inheritance richness	$^{h}/_{c}$ [2, 17, 23]
	annotation richness	$^{a}/_{c}$ [5, 23]
	property class ratio	$^{p}/_{c}$ [5, 24, 7, 23]
	inverse property ratio	$^{inv}/_{p}$ [2, 7, 23]

Table 1: Description and implementation of the primitive and composite metrics in ChImp. `o` is the instance of the ontology within each metric implementation.

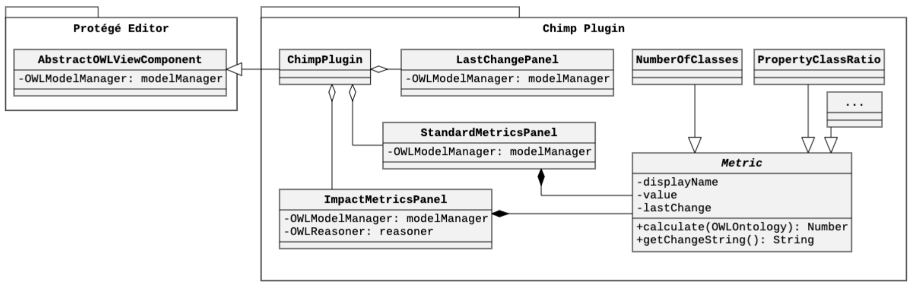


Fig. 5: UML Class diagram of the ChImp implementation, showing its architecture.

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 [14], 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. Currently, it displays whether or not the underlying ontology is consistent. In the future, impact measures would also make use of the reasoner.

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 Conclusions and Future Work

This work aims at closing the gap between ontology engineers and the information about the changes they apply to an ontology. With a requirements survey, we asked practitioners about their opinion and preferences on visualizing 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. **R5** through **R7** require ontology release notes to include information such as number of changes, consistency, and ontology measures. We plan to extend ChImp with a feature which allows exporting this information when saving the ontology for release purposes.

There are various extensions already planned for ChImp. In the future, we would like to give the engineer the option to select the metrics that are displayed in the Metrics Table and also offer more quality- and semantic-oriented measures. In addition, listed changes are currently based on the change listener within Protégé. The inclusion of more complex aggregation and classification of changes, e.g., COnto-Diff [13], is one possible extension. We will investigate the possibility of impact measures using the materialization but also other applications like embeddings or functional enrichment analysis [21]. Depending on the application, impact measures can either be computed directly, approximated, or learnt and then predicted. The latter case would entail the loading of a learnt model for impact prediction [21].

Lastly, we are planning to use ChImp conduct a detailed user study. We will investigate the awareness of engineers concerning the consequences of changes. Participants will be asked to solve specific tasks, during which they will change an ontology and either have access to ChImp or not. With a post-task survey, we will assess their awareness and compare their editing behaviour in detail.

ChImp will enable studies about awareness of consequences of ontology changes. Additionally, we can potentially improve the ontology evolution processes and make both the editing of and transitioning to new versions of ontologies easier and more accessible for ontology users. In the long run, we hope that these kinds of studies will lead to insights that have a comparable impact on the field as software evolution research had on software engineering.

References

1. Dehmer, M., Emmert-Streib, F., Shi, Y.: Interrelations of Graph Distance Measures Based on Topological Indices. PLOS ONE **9**(4), e94985 (Apr 2014). https://doi.org/10.1371/journal.pone.0094985
2. Djedidi, R., Aufaure, M.A.: 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. pp. 286–305 (2010)
3. Drummond, N.: ChangeView (Mar 2011), https://code.google.com/archive/p/co-ode-owl-plugins/wikis/ChangeView.wiki
4. Dudás, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. Knowledge Eng. Review **33**, e10 (2018)
5. Duque-Ramos, A., Fernández-Breis, J.T., Iniesta, M., Dumontier, M., Aranguren, M.E., Schulz, S., Aussenac-Gilles, N., Stevens, R.: Evaluation of the OQuaRE framework for ontology quality. Expert Systems with Applications **40**(7), 2696–2703 (2013)
6. Falconer, S.M., Tudorache, T., Noy, N.F.: An analysis of collaborative patterns in large-scale ontology development projects. In: K-cap. pp. 25–32. ACM (2011)
7. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: European semantic web conference. pp. 140–154 (2006)
8. Gatens, W., Konev, B., Ludwig, M., Wolter, F.: Versioning based on logical difference for lightweight description logic terminologies. Proc. of ARCOE-11 (2011)
9. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: HermiT: an OWL 2 reasoner. Journal of Automated Reasoning **53**(3), 245–269 (2014), publisher: Springer
10. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: CIKM. pp. 1541–1546. ACM (2011)
11. Gottron, T., Gottron, C.: Perplexity of Index Models over Evolving Linked Data. In: ESWC. vol. 8465, pp. 161–175. Springer (2014)
12. Gross, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E.: Impact of ontology evolution on functional analyses. Bioinformatics **28**(20), 2671–2677 (2012)
13. Hartung, M., Gross, A., Rahm, E.: COnto-Diff: Generation of complex evolution mappings for life science ontologies. Journal of Biomedical Informatics **46**(1), 15–32 (Feb 2013)

14. Horridge, M., Bechhofer, S.: The owl api: A java api for owl ontologies. Semantic Web **2**(1), 11–21 (Jan 2011)
15. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.G.: Ontology visualization methods - a survey. ACM Computing Surveys **39**(4), 10 (2007)
16. Kremen, P., Smid, M., Kouba, Z.: OWLDiff: A practical tool for comparison and merge of OWL ontologies. In: DEXA workshops. pp. 229–233. IEEE Computer Society (2011)
17. Lantow, B., Sandkuhl, K.: An analysis of applicability using quality metrics for ontologies on ontology design patterns. Intelligent Systems in Accounting, Finance and Management **22**(1), 81–99 (2015)
18. Musen, M.A.: The protégé project: a look back and a look forward. AI Matters **1**(4), 4–12 (2015)
19. Noy, N.F., Klein, M.C.A.: Ontology evolution: Not the same as schema evolution. Knowl. Inf. Syst. **6**(4), 428–440 (2004)
20. Osborne, F., Motta, E.: Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance. In: ISWC. LNCS, vol. 11136, pp. 495–512. Springer (2018)
21. Pernischová, R.: The Butterfly Effect in Knowledge Graphs: Predicting the Impact of Changes in the Evolving Web of Data. In: ISWC: Doctoral Consortium. CEUR-WS.org, Aukland, NZ (Oct 2019)
22. Pernischová, R., Dell'Aglio, D., Horridge, M., Baumgartner, M., Bernstein, A.: Toward predicting impact of changes in evolving knowledge graphs. In: ISWC satellites. CEUR workshop proceedings, vol. 2456, pp. 137–140. CEUR-WS.org, Aukland, NZ (Oct 2019)
23. Tartir, S., Arpinar, I.B., Sheth, A.P.: Ontological evaluation and validation. In: Theory and applications of ontology: Computer applications, pp. 115–130. Springer (2010)
24. Tempich, C., Volz, R.: Towards a benchmark for Semantic Web reasoners-an analysis of the DAML ontology library. In: EON. vol. 87 (2003)