Protégé4US: Harvesting Ontology Authoring Data with Protégé

Markel Vigo, Caroline Jay, and Robert Stevens

School of Computer Science, University of Manchester, Manchester (United Kingdom)
{markel.vigo|caroline.jay|robert.stevens}@manchester.ac.uk

Abstract. The inherent complexity of ontologies poses a number of cognitive and perceptual challenges for ontology authors. We investigate how users deal with the complexity of the authoring process by analysing how one of the most widespread ontology development tools (i.e. Protégé) is used. To do so, we build Protégé4US (Protégé for User Studies) by extending Protégé in order to generate log files that contain ontology authoring events. These log files not only contain data about the interaction with the environment, but also about OWL entities and axioms. We illustrate the usefulness of Protégé4US with a case study with 15 participants. The data generated from the study allows us to know more about how Protégé is used (e.g. most frequently used tabs), how well users perform (e.g. task completion times) and identify emergent authoring strategies, including moving down the class hierarchy or saving the current workspace before running the reasoner. We argue that Protégé4US is an valuable instrument to identify ontology authoring patterns.

1 Introduction

Semantic technologies present interaction challenges for both developers of semantic artefacts and their users. This is particularly true for authors of ontologies authored in the Web Ontology Language (OWL), where complex systems of axioms are asserted and then used to draw implications about that ontology. Ontologies are complex artefacts and this complexity comes from different dimensions: the domain, the size of the artefact, the semantics and the interface of the development environment, be the latter a text processor or an IDE for ontology development.

In an initial study with 15 ontology experts we found that, indeed, these complexities cause problems to users along all stages of the authoring process [6]. Exploration, search, building, reasoning, debugging and evaluation pose a number of challenges that users overcome by employing authoring strategies that may be suboptimal. In that study some of the most relevant outcomes in terms of reported problems, and missing or poorly supported features of tools are:

– Making sense, getting an overview and exploring an ontology is difficult if the ontology is large or the user is not familiar with it.

- The consequences of minor edits can have enormous implications on the underlying ontology. Since current tools are not able to convey these changes the situational awareness of users is dramatically reduced.
- Searching for ontologies and, especially, retrieving components of an external ontology into the active ontology is a missing feature.
- The efficient addition of axioms to large ontologies remains a challenge.
- On-the-fly reasoning capabilities is reported as a desirable feature as users tend to want to frequently run the reasoner to test their latest update or set of modifications.
- Explanations for inconsistencies and inferences tend to be perceived to be overloaded with information, making the debugging of ontologies hard.
- Having predefined 'unit' tests would help in assessing the validity and completeness of ontologies.

Beyond this, little is known about the activity patterns of the ontology authoring process, the details of problems authors encounter and how these difficulties are overcome. The analysis of the literature indicates that tools are normally evaluated against a number of established criteria. What is more, user involvement in the assessment of authoring tools is scarce (see Section 2). This may be a consequence of a lack of interest in the human factors of the ontology authoring process as Human-Computer Interaction practices do not pervade all computing disciplines. Additionally, this could also happen because there are not enough instruments to run ontology authoring studies and experiments.

We address this lack of instrumentation by building Protégé4US (Protégé for User Studies), a modification of one of the most widely used ontology authoring environments [1], Protégé. In Section 3 we describe how these modifications generate log files containing the interactions of users with the environment and the authoring activities they undertake. Then, in order to corroborate the findings of our preliminary study [6] and find patterns of activity, we run a user study in which 15 participants[1] carry out three authoring tasks with Protégé4US (see Section 4). Results in Section 5 illustrate the potential of this approach as it facilitates the computation of performance metrics, the number and variety of commands invoked and the patterns in the authoring process. Consequently, the contributions of this paper are twofold:

- We provide an instrument that allows harvesting of ontology authoring events and interaction events.
- A study with 15 users carrying out ontology authoring tasks shows the potential of such a tool to expand our knowledge about the ontology authoring process.

2 Related Work

Usability, interoperability, and portability are some of the dimensions identified in a survey about semantic authoring tools of textual content [3]. Only two papers

[1] Four of these participants also took part in the initial interview study [6].

out of the 175 papers that were analysed involved end-users in evaluating the user interface. In both cases enquiry methods were employed *a posteriori* in order to provide evidence about user acceptability of the evaluated tools. Even if the mentioned work focuses on a particular area of the semantic web, it is an indicator of the lack of involvement of users in the development process of ontology authoring tools.

Nevertheless, there are some notable exceptions in the evaluation of ontology authoring tools with users: eight non-expert users carried out a number of basic tasks including ontology loading, entity addition, modification, and removal [4]. After completing their tasks, users were given questionnaires in order to quantify the relevance of tools to accomplish tasks, tool efficiency, user attitude towards tools, and learnability aspects. In another study 28 participants with basic OWL knowledge carried out three tasks with TopBraid Composer and Protégé [2]. Tasks included creating classes and properties, as well as adding subsumption, equivalence, and range axioms. Afterwards, participants were given questionnaires to measure the effectiveness, efficiency, and user experience with the mentioned tools.

A major criticism of existing studies is their sole reliance on questionnaires. In addition to their questionable reliability due to their subjective nature, a key weakness of questionnaires is that preconceived questions get predefined responses and participants are unable to express additional thoughts that would have been interesting for researchers. This is something we addressed in [6] by interviewing ontology authors. While self-reported data can be invaluable in analysing the subjective dimensions of experience, the sole reliance on questionnaires shows an incomplete picture of behaviour analysis. We argue that self-reports should preferably be complemented with objective interaction data. In this sense, the advent of web based authoring environments such as WebProtégé [5] will bring about not only the remote involvement of users, but will also facilitate the analysis of server log data as shown in [7]. While server log analysis allows to increase the involvement of geographically distributed participants, it has several downsides including the lack of contextual information or the difficulty in identifying the tasks and goals of users.

In this paper we present a tool that logs objective interaction and authoring data, Protégé4US; additionally, we conduct a feasibility study of the tool by analysing the data generated by real users carrying out real tasks.

3 Instrumentation of Protégé

We modified Protégé 4.3 by adding event listeners to the interface elements and by logging every class that implemented any user activity. As a consequence, a CSV file that contains the timestamps and information about the events is generated as shown below in an extract from a log file. The first value in each row shows the timestamp in milliseconds, while the second value indicates the type of event. Even though a broad range of events is collected, we can classify these in to three main types: interaction events, authoring events on the current ontology and the commands invoked in the working environment —we assign to each event a value of *1*, *2* or *3* respectively. The third column denotes the Protégé tab that was active when the event occurred as depicted by Figure 1. The fourth value describes the event and the last column indicates the object of the event: i.e. the specific class that was edited, the reasoner that was invoked or the class that was hovered over.

```
1:  76585,2,Classes,Element edited,Juliette subclass of: Potato and hasCroppingTime
some ‘Main cropping’
2:  77786,3,Classes,Save ontology,http://owl.cs.manchester.ac.uk/ontology/start-here.owl
3:  80204,3,Classes,Reasoner invoked,HermiT 1.3.8
4:  80647,1,Classes,Mouse entered, Class hierarchy (inferred)
5:  82910,1,Classes,Element hovered,Early_cropping_potato
6:  83049,1,Classes,Element selected,Early_cropping_potato
7:  83661,1,Classes,Hierarchy expanded,Early_cropping_potato
```

From this we can later reconstruct and analyse the authoring process: line 1 of the log file shows —using a potato ontology we describe later in Section 4— how the class of Juliette potatoes is harvested between September and November, as indicated by the value of the hasCroppingTime property, which is ‘Main cropping’. After that, line 2 and 3 tell that the ontology was saved and that the HermiT reasoner was invoked right after. Once the reasoner was finished the user entered into the inferred class hierarchy (line 4), and selected the Early_cropping_potato class (line 5-6) in the inferred class hierarchy. Finally line 7 indicates that the hierarchy under this class was expanded.

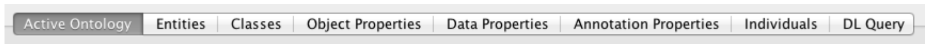


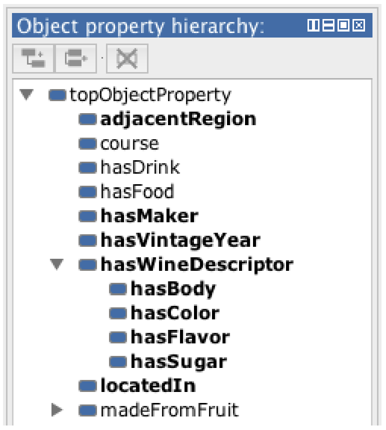
Fig. 1: Default tabs of Protégé 4.3

3.1 Interaction events

The user interface of Protégé 4.3 is implemented using data structures such as hierarchies and lists that are containers of different types of entities. For instance, the hierarchy data structure can contain classes, properties, data and annotations (see Figure 2). Similarly, lists expand the description of a particular element (see Figure 3) and are used to describe classes, properties, data properties, annotations and individuals. Therefore, in Protégé4US the functionalities of hierarchies and lists, and the events triggered by user interaction are analogous regardless of the type of entity they contain. As a result, the interaction events described in Table 1 apply to all hierarchies and lists, irrespective of the type of entity upon which the event is generated.

3.2 Authoring events

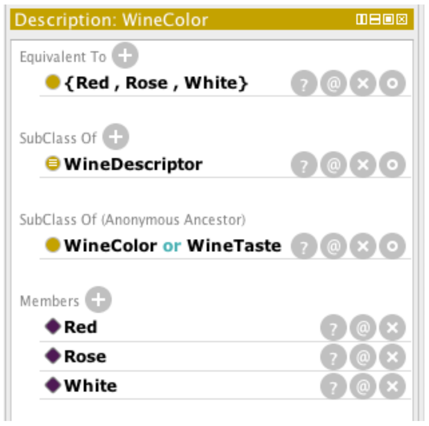
Authoring events are those that modify the state of the ontology. Table 2 does not only describe the CRUD events (create, read, update and delete) that Protégé4US collects, but also gives specific details about the modification of entities in terms of *how*, amongst others, ranges, domains or restrictions are set.



Object property hierarchy:

- topObjectProperty
  - adjacentRegion
  - course
  - hasDrink
  - hasFood
  - hasMaker
  - hasVintageYear
  - hasWineDescriptor
    - hasBody
    - hasColor
    - hasFlavor
    - hasSugar
  - locatedIn
  - madeFromFruit

Fig. 2: The hierarchy structure of Protégé 4.3: property hierarchy



Description: WineColor

Equivalent To
{Red , Rose , White}

SubClass Of
WineDescriptor

SubClass Of (Anonymous Ancestor)
WineColor or WineTaste

Members
Red
Rose
White

Fig. 3: The list structure of Protégé 4.3

Table 1: Interaction events in Protégé4US

Event	Description
Annotations	Hovering and selecting
Lists of entities	Entering/exiting the list, hovering and selecting entities
Links	Hovering and selecting
Hierarchies	Entering/exiting the hierarchy, hovering and selecting entities. Expanding and collapsing entity hierarchies. Hovering tooltips
Individuals	Hovering and selecting
Search result panel	Hovering, selecting and clicking on search results

Table 2: Authoring events in Protégé4US

Event	Description
Add entities	Add child/sibling classes, properties, individuals or annotations
Class conversion	Converting a class into a defined class or a primitive class
Class hierarchy wizard	A hierarchy is added into the current ontology
Characteristics of properties	Check/uncheck the characteristics of properties to make them functional, inverse functional, transitive, symmetric, asymmetric, reflexive or irreflexive
Characteristics of classes	Create a defined class or an inverse of a existing class
Closure	Add a closure
Delete	Delete an entity or a restriction
Duplicate entity	When an entity with the same characteristics as another is created
Entity dragged	Drag & drop an entity in a hierarchy
Entity edited	Create universal, existential and cardinality restrictions to classes. Add domain and ranges to properties
Rename entity	Invoke refactoring to rename an entity

3.3 Environment commands in Protégé4US

The events shown in Table 3 can be understood as the features added by ontology authoring IDEs on top of the basic functionalities required to build an OWL ontology (see Table 2). These include the events generated by the interaction with buttons that trigger actions, the invocation of the reasoner and explanation handler, and the file saving events.

Table 3: Events on the Protégé4US environment

Event	**Description**
Button clicked	When clicking on the interface buttons to add or remove child/sibling classes, properties, individuals and annotations
Class hierarchy wizard	When the wizard is invoked through *Tools* $\rightarrow$ *Create class hierarchy...*
Explaining	Invoking the explanation handler to clarify an inference or an inconsistency
History navigation	When the back and forth arrows are clicked, and when the *undo* or *redo* commands are invoked
Reasoning	Events are generated when a reasoner is invoked, when reasoning is finished and when the reasoner is stopped
Search panel	When the user types a query in the search box
Usage panel	When the checkboxes of the usage panel are clicked in order to establish the visualisation criteria: all, disjoints, differents and superclasses
Workspace events	Ontology saved, loaded and set as the active ontology

4 Study

4.1 Participants

Fifteen participants (ten male) in the age range between 22–47 and a median age of 32 took part in our study. They had a background in computer science and worked both in academia or industry. We used snowball sampling to contact possible candidates and invited to participate those who reported to be knowledgeable about OWL and Protégé, which is corroborated by the self-reported assessment about these selection criteria in Figures 4 and 5. Users completed a questionnaire containing 5-point Likert scales in order to answer "Assess your expertise with OWL" and "Assess your expertise with Protégé", where 1 indicated 'Novice' and 5 was for 'Expert'.

4.2 Experimental design

The goal of the tasks was to make an ontology of potatoes that could drive a 'potato finder' application. Participants were told to carry out the following three tasks with an intended incremental difficulty level:

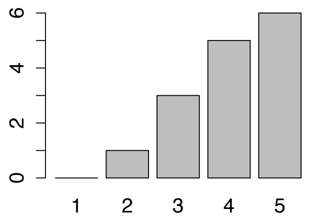


Fig. 4: Reported OWL expertise: 1-novice, 5-expert; X-axis: expertise, Y-axis: frequency

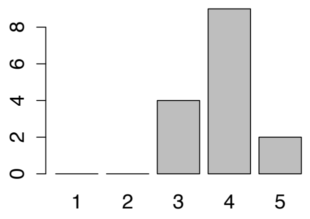


Fig. 5: Reported Protégé expertise: 1-novice, 5-expert; X-axis: expertise, Y-axis: frequency

1. Classify the potatoes by cropping times.
2. Import a file containing descriptions of potato yields. Represent the yields of each kind of potato and classify by combinations of yield and cropping time.
3. Add in a representation of culinary role (ie. preferred way of cooking). Build at least two classes that combine the three axis (culinary role, yield and cropping time).

Participants were also provided with a printed table with the necessary information —cropping time, yield or skin colour— to build the potato ontology. They were also told not to start from scratch, but they should adopt the persona of a 'jobbing' ontologist given an OWL ontology to extend and maintain. Therefore, participants were provided with an OWL file containing 13 subclasses of various potato varieties and another one with a small hierarchy of cropping times for potatoes, which also removed the burden of heavy editing.

Protégé4US was deployed onto a Windows 7 laptop and used to carry out the above-mentioned tasks. There was not a fixed time to complete the tasks although participants were free to stop their participation at any time. Participants filled out a post-test questionnaire about the perceived difficulty of each task using a 5-point Likert scale where 1 indicated 'easy' and 5 'difficult'.

5 Results

5.1 Performance metrics

The median values for completion times indicate that it took 11.04 minutes for Task 1 ($m = 6.9, M = 18.05$), 11.83 minutes for Task 2 ($m = 7.36, M = 36.53$) and 17.87 minutes for Task 3 ($m = 10.13, M = 28.62$). These differences in completion time are significant as suggested by the Kruskal-Wallis test, $\chi^2 = 10.04, p < 0.01$. Figure 6 shows the completion times of each participant per task. All participants completed their task except for P7 and P15[2], who were

[2] Participants are coded using the Pi notation, where i denotes the participant number $1 \leq i \leq 15$.

not able to complete Task 2 and 3. While P7 gave up participating in the study, P15 finished all the tasks, but did not achieve the goals we established. The perceived difficulty of each task can be seen in Figure 7, where the median value is 1 for Task 1, 2 for Task 2 and 2.5 for Task 3. A Kruskal-Wallis test indicates that these differences are significant, $\chi^2 = 10.79, p < 0.005$. This means that we succeeded in defining tasks with an increased level of difficulty.

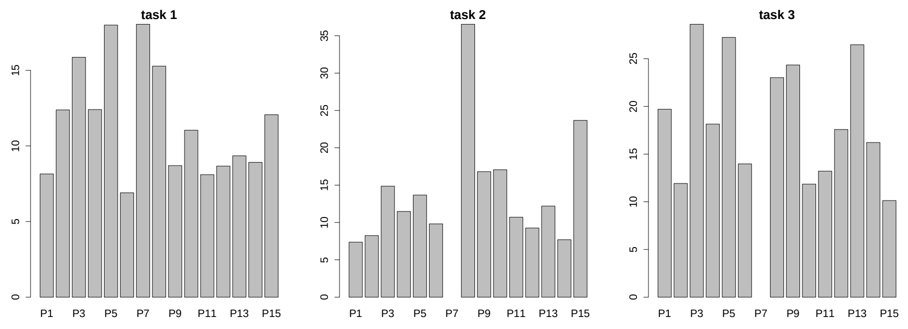


Fig. 6: Completion times for tasks; X-axis: participants; Y-axis: time in minutes

We found a moderate negative correlation between self-reported expertise with Protégé and the task completion time for Task 1, Spearman's $\rho = -0.51, p < 0.05$. This suggests that the more expert a participant was with Protégé, the faster Task 1 was completed. No correlation is found between expertise (either with Protégé or OWL) with the remaining task completion times. There is no correlation between the time taken to complete a task and its perceived difficulty, which suggests that the time taken was more an indicator of the number of authoring actions required to complete a task than an indicator of cognitive difficulty.

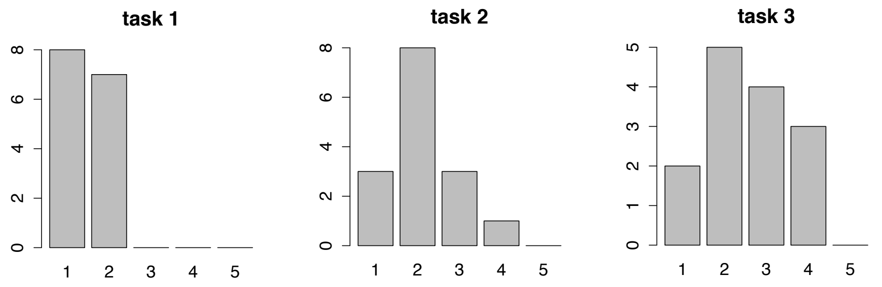


Fig. 7: Perceived difficulty of tasks: 1-easy, 5-difficult; X-axis: difficulty; Y-axis: frequency

5.2 Usage metrics

Figure 8 shows the number of events triggered by each participant at each Protégé tab (see Figure 1). It can be observed that there are roughly 2 types of users when it comes to tab usage. On the one hand, those who stick to one tab: P1, P2, P14 and P15 use the *Classes* tab, while P3, P4, P5, P6, P9, P11 and P12 use mostly the *Entities* tab. On the other hand, P7, P10 and P13 use mostly two tabs, one of which is the *Classes* tab.

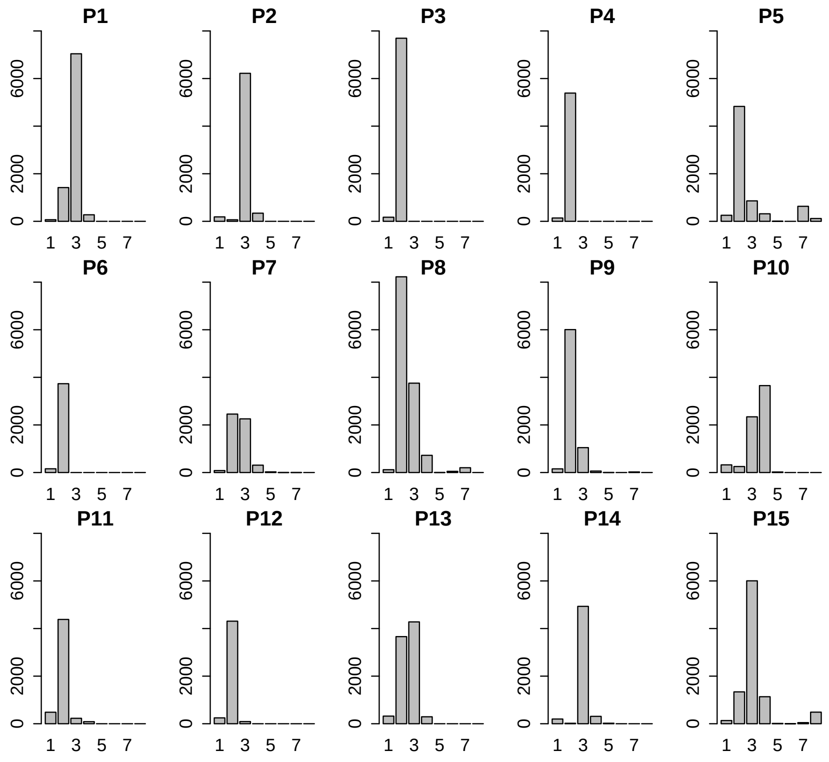


Fig. 8: Tab usage per user in Protégé 4.3; X-axis: 1: Active Ontology; 2: Entities; 3: Classes; 4: Object Properties; 5: Data Properties; 6: Annotation properties; 7: Individuals; 8: DL Query; Y-axis: number of events occurring in each tab

We selected a sample of the total events as they may be indicative of typical behaviours based on the findings of our previous study [6]. *Undoing*, *deleting*, *navigating back* and *renaming* are indicative of users trying to repair mistakes made. The invocation of the *reasoner* and *explanation handler* is often a way of testing the current ontology for the former and the user trying to know more about the source of problems for the latter. *Expanding* any hierarchy suggests that users are navigating the class or property hierarchy trying to find an entity. Table 4 shows the number of events triggered by each participant.

Table 4: Sample of the events triggered by each participant

Participant	**Undo**	**Back**	**Delete**	**Save**	**Rename**	**Reason**	**Explanation**	**Expand hierarchy**
P1	0	0	7	11	0	12	2	80
P2	1	0	5	33	0	12	0	90
P3	0	0	2	1	0	14	0	858
P4	0	0	4	15	0	9	0	82
P5	0	0	1	4	12	16	0	105
P6	0	0	1	1	0	2	0	74
P7	0	5	3	0	0	2	0	403
P8	2	0	3	15	3	36	1	316
P9	0	0	1	15	3	14	4	409
P10	0	0	2	11	0	0	1	80
P11	0	0	2	19	0	12	1	97
P12	7	0	2	5	2	0	0	69
P13	2	0	11	5	1	15	0	208
P14	0	0	0	1	0	4	0	279
P15	0	0	9	2	2	24	10	66

Considering the number of participants we can relax the α value and thus consider marginally significant those correlations with a *p value* between 0.05–0.1. When computing Spearman's correlation we find a strong correlation between the number of times a hierarchy has been expanded and the time taken to complete Task 3 —$\rho = 0.68, p < 0.01$— and a moderate correlation in Task 1, $\rho = 0.46, p = 0.08$. According to the strategies we enumerate in Section 1, this may be indicative of users having some trouble in that they are navigating in the hierarchy and exploring an ontology with which they are not familiar. Similarly, there is a moderate correlation in Task 2 between the time taken to complete the task and the number of times the reasoner was invoked, $\rho = 0.46, p = 0.08$. Again, this is supported by some of the insights we obtained in the interviews (see Section 1 when summarising the outcomes of [6]): these data suggest that users who did not achieve their goal straight away ran the reasoner time and again to test their latest update until the goal was achieved. Moderate correlations were found between the times entities are renamed and task completion time, $\rho = 0.53, p = 0.05$. This may indicate that renaming entities was a factor that delayed users completing their tasks.

5.3 Activity patterns

Since the sequence of events and their timestamps are collected, we are able to reconstruct the interaction and plot time diagrams such as the one shown in Figure 9.

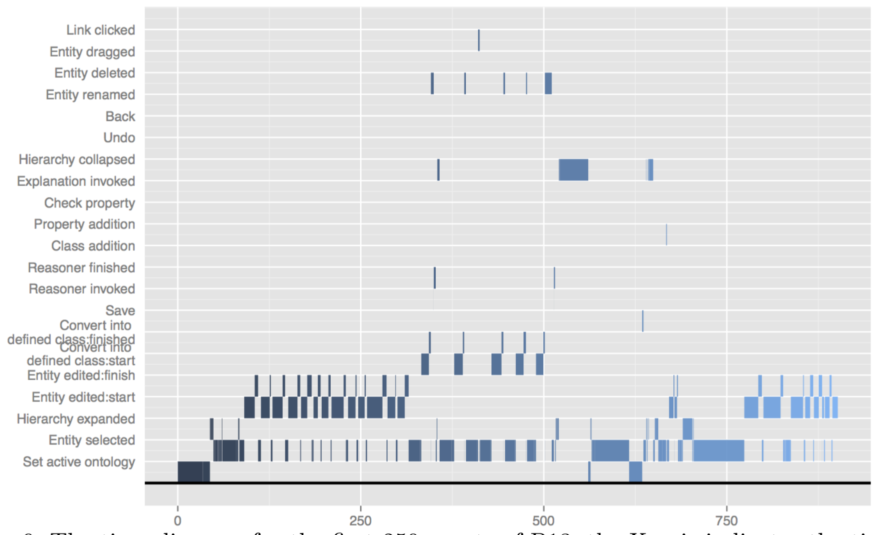


Fig. 9: The time diagram for the first 250 events of P13: the X-axis indicates the time elapsed in seconds, while the Y-axis shows the events triggered by the user.

The blocks between events denote the time spent between events in which no other event (from the ones we sample) was triggered. These diagrams give visual hints on the strategies employed by participants to achieve their goals. A preliminary visual analysis of 15 time diagrams allows us to uncover some activity patterns and regularities that may be shaped as a decision tree:

- The reasoner is invoked after,
  (a) A class is converted into a defined class.
  (b) An entity's modification is finished.
  (c) The ontology is saved.
  (d) An entity is selected.
- An ontology is saved after,
  (a) An entity's modification is finished.
- A tree is expanded after,
  (a) Reasoner finishes.
  (b) A tree is expanded.
  (c) Entity edition has been invoked.
- An entity is selected after,
  (a) Another entity is selected.

In order to corroborate the information derived from the analysis of user interaction visualisations, future work will delve into analysing these data statistically. Consequently, we would be able to anticipate user behaviour and make Protégé adaptive. By implementing adaptive features we can address some of the limitations of current tools we list in Section 1. For instance, if we confirm that users invoke the reasoner after saving the ontology, the reasoner could be run as a background process right after saving, which would save time for the user.

6 Conclusion

We present Protégé4US, an instrument to carry out user tests of ontology authoring tasks. As a proof of concept we run a study in which 15 experts complete three tasks of different difficulty. The results show that it is feasible to collect objective and reliable performance metrics such as task completion time. Within the context established by these tasks we are able to: 1. identify two types of users based on how they use the tabs of Protégé; 2. find correlates between interaction events and performance metrics that corroborates our initial insights [6]: a higher number of times the reasoner is invoked indicates trouble and thus, longer completion times; 3. visualise emerging activity patterns: e.g. an ontology is saved before invoking the reasoner and after modifying an entity. This suggests that our instrument has an enormous potential to expand our knowledge about the ontology authoring process, identify its pitfalls, propose design guidelines and develop intelligent authoring tools that anticipate user actions in order to support ontology authoring in the future.

7 Acknowledgements

The source code of Protégé 4.3 was retrieved from
http://smi-protege.stanford.edu/repos/protege/protege4/protege-base/trunk.
Protégé4US and the datasets generated in this study can be downloaded from
http://owl.cs.manchester.ac.uk/whatif/.
This research has been funded by the EPSRC project: *WhatIf: Answering "What if..." questions for Ontology Authoring*. EPSRC reference EP/J014176/1.

References

[1] J. Cardoso, Jorge: The Semantic Web Vision: Where Are We? IEEE Intelligent Systems 22(5), 84–88 (2007)
[2] M. Dzbor, E. Motta, C.B Aranda, J.M Gomez-Perez, O. Goerlitz and H. Holger: Developing ontologies in OWL: An observational study. OWL: Experiences and Directions Workshop (2006)
[3] A. Khalili and Sören Auer: User interfaces for semantic authoring of textual content: A systematic literature review. Web Semantics: Science, Services and Agents on the World Wide Web 22, 1–18 (2013)
[4] P. Lambrix, M. Habbouche, M. Pérez: Evaluation of ontology development tools for bioinformatics. Bioinformatics 19(12), 1564–1571 (2003)
[5] T. Tudorache, J. Vendetti and N.F. Noy: Web-Protege: A Lightweight OWL Ontology Editor for the Web. Proceedings of the 5th OWLED Workshop on OWL (2008)
[6] M. Vigo, C. Jay and R. Stevens: Design Insights for the Next Wave Ontology Authoring Tools. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI' 14, 1555–1558 (2014)
[7] H. Wang, T. Tudorache, D. Dou, N.F. Noy and M.A. Musen: Analysis of User Editing Patterns in Ontology Development Projects. On the Move to Meaningful Internet Systems: OTM 2013 Conferences. LNCS 8185, 470–487 (2013)