Formal Ontology in Information Systems
R. Ferrario and W. Kuhn (Eds.)

doi:10.3233/978-1-61499-660-6-211

Antipattern Comprehension: An Empirical Evaluation

Tie HOU [a,1], Peter CHAPMAN [b] and Andrew BLAKE [a]
[a] *Visual Modelling Group, University of Brighton*
[b] *Edinburgh Napier University*

Abstract. Comprehension of justifications is known to be difficult for even experienced ontology engineers, and much more so for other stakeholders. In this paper, we present two methods for displaying justifications using concept diagrams: using multiple concept diagrams to represent the justification (one diagram for each axiom); and using a merged concept diagram to represent all axioms in the justification. We performed an empirical evaluation of both methods along with a textual representation of the justification using Protégé. The results were that novice users could both more accurately and more quickly identify an incoherence when using merged diagrams than using multiple diagrams or Protégé statements.

Keywords. Evaluation; antipattern; visual

1. Introduction

Debugging and repairing ontologies is important for ontology evaluation [1]. Neuhaus *et al.* identify five high-level characteristics that ontologies should have. Three relate directly to debugging, repair and incoherence: intelligibility, fidelity and craftsmanship. Intelligibility is concerned with human understanding: a justification seeks to enable understanding by providing a minimal set of axioms from which it can be inferred that a concept or property is unsatisfiable. Fidelity is concerned with whether or not domain knowledge is accurately represented: a justification may reveal that expected causal links are not present in ontology. Craftsmanship requires that an ontology is well-built: justifications highlight inconsistencies in axioms, and suggest ways in which the inconsistencies can be rectified. Making the comprehension of justifications of incoherence more accessible is then of paramount importance to engineers.

The ontology engineer will be unable to debug or repair ontologies in an efficient and robust manner without understanding how the incoherence follows from the justification. However, it is known that understanding justifications of incoherence is difficult [2,3]. A variety of automated tools exist to identify incoherence, and provide aid to the engineer, but even with such tools the task is still difficult [4,5]. Our approach to alleviating the burden on engineers is through visualization. In this paper, we examine

[1] Tie Hou; E-mail: t.hou@brighton.ac.uk. The first two authors were supported as part of the EPSRC Grant *Visual Justifications for Ontologies* [EP/M016323/1]

whether diagrammatic justifications can make identification of unsatisfiable concepts or properties less error-prone.

Visualization of complex data can provide a way of making the information more accessible [6]. Visualizations can provide visual affordances [7,8] whereby information can be read-off a diagram that would otherwise have had to be inferred. For example, in Figure 1, we can read-off the diagram that $\mathsf{Lorry} \sqcap \mathsf{Hall} \sqsubseteq \bot$, even though that is *not* one of the axioms that constructed the diagram. For this reason, visualization of ontologies is an active area of research, with a variety of approaches and tools existing (see [9,10] for overviews). However, most of these visualization tools show only the hierarchical structure of an ontology, and do not visualize object or data properties. Even then, the tools are useful to comprehension [11]. Incoherence in an ontology can, however, arise from the interaction between concepts and properties, and thus any effective visualization must represent both. Concept diagrams, the visualization used in this paper, visualize both concepts and object properties.

Using our visualization we address the research question in this paper: *does visualization of incoherent ontologies make them easier to understand?* By providing visualizations of common anti-patterns, based on those in [12], we provide insight into whether or not visual affordances exist for justification visualization, and whether visualization can alleviate the burden on ontology engineers when debugging incoherent ontologies.

The structure of the paper is as follows. In section 2 we introduce the syntax and semantics of concept diagrams, and in section 2.1 we summarise the procedure for merging diagrams together. We discuss the study design in section 3, including a description of the software used to collect results (in section 3.4). We describe the results in section 4 with a particular focus on the error analysis in 4.1. Finally, we conclude and describe the next steps of the research in section 5.

2. Concept Diagrams

This section is intended only as an overview of the syntax and semantics of concept diagrams. For a fuller exposition, see [13,14]. Concept diagrams are extensions of Euler diagrams, designed with the intention of representing ontologies. Like Euler diagrams, they use closed curves to represent sets, in this case concepts, and their arrangement in the plane represents subsumption, disjointness or intersection of those concepts. For example, Figure 1 shows a concept diagram where the curve labelled Lorry is completely contained within the curve labelled MobileBase. This relationship between the curves represents the axiom $\mathsf{Lorry} \sqsubseteq \mathsf{MobileBase}$. Similarly, the curve labelled MobileBase is disjoint from the curve labelled Hall, representing the axiom $\mathsf{MobileBase} \sqcap \mathsf{Hall} \sqsubseteq \bot$.

Boxes represent the universe. Where more than one box is present in a diagram, we are representing partial information. For example, in Figure 2, we have no information as to whether Thunder is disjoint from Superpower. We could draw this situation by having the curve labelled Thunder intersect the curve labelled Superpower inside a single box, but intersections between curves would need to be present wherever there was neither a subsumption nor a disjointness relationship holding between the represented concepts. In other words, the diagram would quickly become visually cluttered. By drawing Thunder and Superpower in different boxes, we are reducing visual clutter but still allowing that the two may have a non-empty intersection.

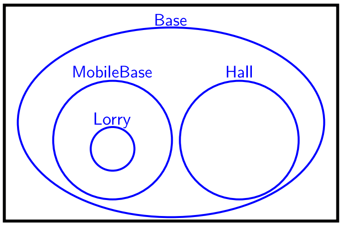

Figure 1. A concept diagram

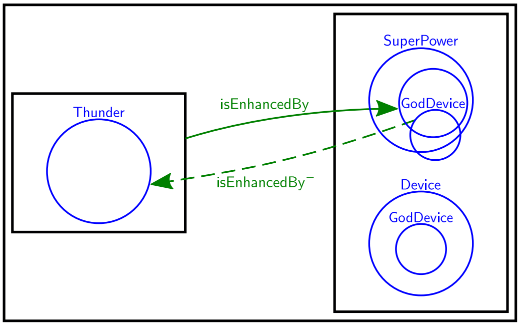

Figure 2. Arrows and multiple boxes

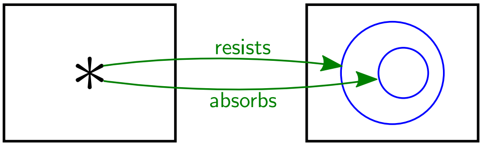

Figure 3. The use of asterisks

Axioms involving object properties are represented using arrows, unlabelled curves, and possibly asterisks. A solid arrow represents total information, whereas a dashed arrow represents partial information. For example, in Figure 2 we see a solid arrow labelled isEnhancedBy, which is sourced on the left-hand box, and targeted on the unlabelled curve completely contained within SuperPower. The meaning of this arrangement is that, between them, any individuals in the left-hand box are related, via isEnhancedBy, to all and only the individuals in the unlabelled curve, all of which are inside SuperPower. In other words, the range of isEnhancedBy is SuperPower, or $\top \sqsubseteq \forall$ isEnhancedBy.SuperPower. By contrast, a dashed arrow represents partial information. Consider the dashed arrow labelled isEnhancedBy^{-} in Figure 2. It is sourced on GodDevice, and targets Thunder. The meaning is as follows: between them, the individuals in GodDevice are related, via isEnhancedBy^{-}, to some individual in Thunder. Crucially, however, other individuals may be related, via isEnhancedBy^{-}, to individuals in Thunder too. In other words, this arrangement represents the axiom Thunder $\sqsubseteq \exists$ isEnhancedBy.GodDevice.

Asterisks are used to encode sub-property information. For example, in Figure 3, we see two arrows labelled absorbs and resists, both sourced on an asterisk, whose targets form a subset-superset pair. The meaning of this is that for every individual, if it is related to something via absorbs, then it is related to the same thing via resists. In other words, absorbs is a sub-property of resists.

2.1. *Merging Diagrams*

The power of concept diagrams is that they can represent more than one axiom at a time. For example, in Figure 1 there are several axioms which can be read off the diagram. By contrast, Figure 3 represents a single axiom. Given a set of axioms it is a simple matter to produce a diagram for each. It is more complicated to then take the resulting set of diagrams and merge them into a single diagram. The full details of this process is outside the scope of this paper; here will give the intuition as to how diagrams are to be merged. For more details, see [13].

Consider merging the four diagrams that make up d_1 in Figure 4. Each sub-diagram of d_1 represents either an axiom of the form A $\sqsubseteq$ B or an axiom of the form A $\sqcap$ B $\sqsubseteq \bot$. Take the top two diagrams (representing MobileBase $\sqsubseteq$ Base and Hall $\sqsubseteq$ Base). Since Base appears in both diagrams, we can combine the curves inside Base. At this stage of the process, we do not know anything about the relationship between MobileBase and Hall, and so we must allow that they can either intersect, one could subsume the other, they could be equivalent, or they could be disjoint. The appropriate arrangement of curves to represent this situation (known as Venn-2) can be seen in top sub-diagram of d_2. In the second stage of merging, we use the information that MobileBase $\sqcap$ Hall $\sqsubseteq \bot$ to separate the curves MobileBase and Hall, giving d_3. Finally, we use the axiom Lorry $\sqsubseteq$ MobileBase to merge the final two sub-diagrams, giving d_4.

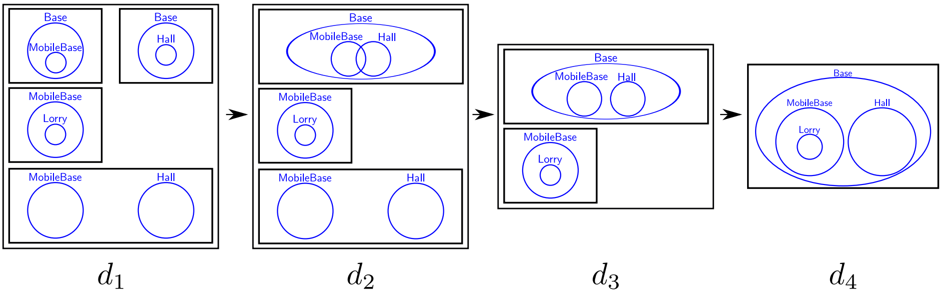

Figure 4. An example of merging

In essence, the merging procedure works the same when arrows (properties) are present in the axioms: we combine the curves where appropriate. If a curve has no relationship with any others (for example the curve Bear in d_1 in Figure 5) then we create less cluttered diagrams by keeping that curve in its own box. In that example, we *could* combine the curve Bear with those curves in the other box. However, since we have no information about the subsumption or disjointness relationships between Bear and Cave and Aeroboat, we would need to allow all possibilities. In other words, the curve Bear would have to split every region in the diagram in two, as shown in d_2 in Figure 5.

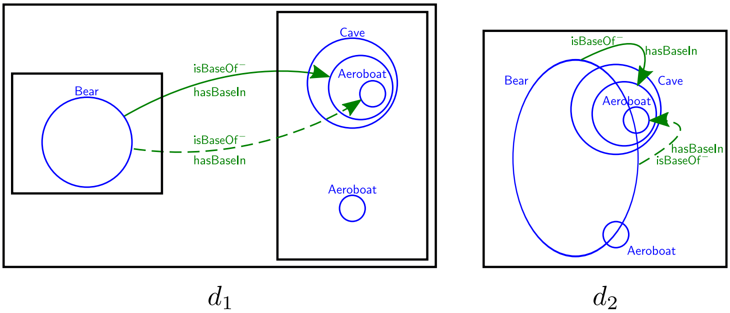

Figure 5. Choosing not to merge

3. Study Design

In this section, we describe our experiment on interpreting and understanding antipatterns using three different representations: merged concept diagrams, multiple concept diagrams and Protégé statements. We propose that incoherence can be easily identified in the merged representation by the separate positions of curves with the same labels. Thus we hypothesise that diagrammatic representations, especially the merged diagrams, are more effective for detecting errors in ontologies than textual representations.

To test the hypothesis a between group design was adopted consisting of three participant groups: merged, multiple and Protégé. Each participant group was presented with a set of 14 different tasks. Importantly, for any given task, the same information was conveyed to each participant group. For example, Figure 6 (in Section 3.2) illustrates one task for each group. Participants were required to identify a number of facts from each task. Performance measurements were made consisting of whether a task was completed correctly and the time taken to complete all the tasks. Consequently, differences in performance between each participant group could be analysed.[2]

3.1. Antipatterns

A set of antipattern categorizations was extracted from the online TONES ontology repository, focusing only on logical antipatterns, rather than those which required domain-specific knowledge [12]. In order that no domain specific knowledge would be needed in order to take part, we adopted a scenario of superheroes and their villains for the antipatterns. Superheroes might consist of alien heroes, humans who have superpowers, gods, etc., whereas villains may be equipped with high-tech devices, evil minds, etc. Table 1 lists examples of antipatterns used in this study.

Table 1. Lists of antipattern examples

(Q1) $\top \sqsubseteq \forall$ isEnhancedBy.SuperPower Thunder $\sqsubseteq \exists$ isEnhancedBy.GodDevice GodDevice $\sqsubseteq$ Device SuperPower $\sqcap$ Device $\sqsubseteq \bot$	(Q2) Psychic $\sqsubseteq \exists$ isEnemyOf.SecretHero $\exists$ isEnemyOf.$\top \sqsubseteq$ HighTechVillain Psychic $\sqsubseteq$ Villain Villain $\sqcap$ HighTechVillain $\sqsubseteq \bot$
(Q6) Team $\sqsubseteq \exists$ isAffilWith.Division Division $\sqsubseteq \exists$ isAffilWith.Bureau Transitive: isAffilWith $\exists$ isAffilWith.Bureau $\sqsubseteq$ Agency Team $\sqcap$ Agency $\sqsubseteq \bot$	(Q7) GodRace $\sqsubseteq$ SecretTeam Iceman $\sqsubseteq$ Costumed Costumed $\sqsubseteq$ GodRace SecretTeam $\sqsubseteq \leq 4$ hasMember.Thing Costumed $\sqsubseteq = 5$ hasMember.Thing
(Q8) MultiPower $\sqsubseteq$ Female SuperSenses $\sqsubseteq$ MultiPower Female $\sqsubseteq$ Hero Hero $\sqcap$ MultiPower $\sqsubseteq \bot$	(Q11) $\forall$ steals.Wood $\sqsubseteq$ Others Others $\sqsubseteq$ Villain $\exists$ steals.$\top \sqsubseteq$ Villain Villain $\sqcap$ Wood $\sqsubseteq \bot$
(Q12) $\top \sqsubseteq \forall$ absorbs.Fire $\top \sqsubseteq \forall$ resists.Heat Heat $\sqsubseteq$ Energy Fire $\sqsubseteq$ Matter absorbs $\sqsubseteq$ resists Energy $\sqcap$ Matter $\sqsubseteq \bot$	(Q13) isBaseOf $\equiv$ hasBaseIn$^-$ Bear $\sqsubseteq \forall$ hasBaseIn.Cave Aeroboat $\sqsubseteq \exists$ isBaseOf.Bear Cave $\sqcap$ Aeroboat $\sqsubseteq \bot$

[2] Tasks and raw data used in the study can be found at https://sites.google.com/site/visual4onto/file-cabinet.

3.2. Task

The task set contained eight different antipatterns, of which three (one containing concepts only, one including properties, and one including a property with a cardinality restriction) were repeated. Finally, three (one containing concepts only, and two including properties) correct justifications were added, so that we could determine whether any of the treatments would lead to false positives, i.e. whether participants could correctly identify that no incoherence was present. The answer choices for each task were concepts or properties taken from its axioms, plus a "None of the above" option. There was at least one correct answer for each task, with some tasks having more than one correct answer. There was a 3-minute time limit, which was set based on results of the pilot study.

Each task contained a set of axioms represented in each of the treatments. Participants were required to identify the emptiness of selected concepts and properties. Figure 6 shows a task, as used in the training material. Note that each participant group was only shown one treatment throughout. Since it can be inferred that Anthro $\sqcap$ Anthro $\sqsubseteq \bot$, it must be the case that Anthro is an unsatisfiable concept. Similarly, both FishLike and BirdLike are necessarily unsatisfiable concepts.

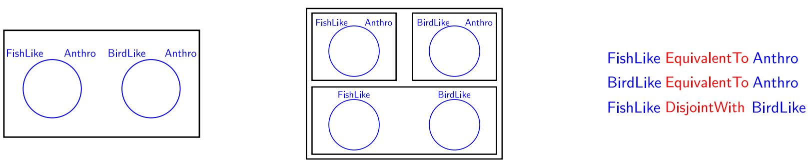

Figure 6. A task question example of an antipattern for merged, multiple, Protégé groups respectively

Starting with the Protégé presentation of the axioms, we generated a multiple diagram representation, and then merged this multiple diagram (using the process outlined in section 2.1). For two axioms, the direct translations into diagrams are visually complex. When such an axiom was needed, we generated diagrams from equivalent axioms. For example, instead of representing $\forall$ steals.Wood $\sqsubseteq$ Others, we interpreted its equivalent alternative $\neg\exists$ steals.$\neg$Wood $\sqsubseteq$ Others as shown in Figure 7 where a gray area (called shading) asserts that nothing is in that area. In the case of transitivity of properties, we simply annotated the diagram, as shown in Figure 8.

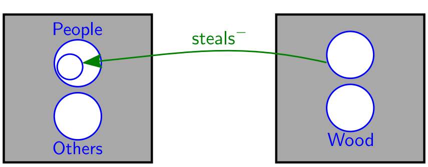

Figure 7. Indirect interpretation

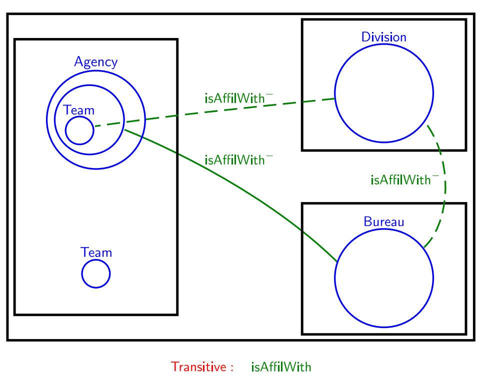

Figure 8. Transitive property

3.3. Drawing Conventions

In order to ensure consistency between representations, we applied the following guidelines when drawing diagrams and presenting Protégé statements: (1) Curve and arrow labels were placed as close as possible to the object. (2) Labels had the same font style (sans serif) and size. (3) The stroke width for boxes was set to 3 pixels. (4) The stroke width for curves and arrows was set to 2 pixels. (5) Blue (RGB(0,0,255)) was used for concept names. (6) Green (RGB(0,128,0)) was used for property names. (7) Red (RGB(255,0,0)) was used for Protégé keywords e.g. SubClassOf, only, etc. (8) Diagrams or statements were on the left side of the screen, answer choices were on the right. (9) Answer choices were displayed in the same location for each task with identical spacing.

3.4. Study Software

The main part of the study was conducted using software that recorded the elapsed time for each task by tracking clicks between submission of answers. The software presented 14 tasks randomly ordered for each participant one at a time. It was not permitted to attempt the next task until at least one option was chosen for the current task. Since questions for each task in a group were the same, the question did not appear on the screen. Instead, it was told to participants by the experiment facilitator at the beginning that the task question was: *Which of the following has to be true for this diagram/this set of diagrams/this set of statements?*

Once a question was answered or 3-minutes elapsed without a participant response (a *time-out*), the software moved to a 'Pause' page which allowed participants to take a break for as long as they wished before continuing to the next task. A 'Finish' page indicated that participants had attempted all tasks. Figure 9 shows an example of a task whose correct answer is F for the merged group within the software.

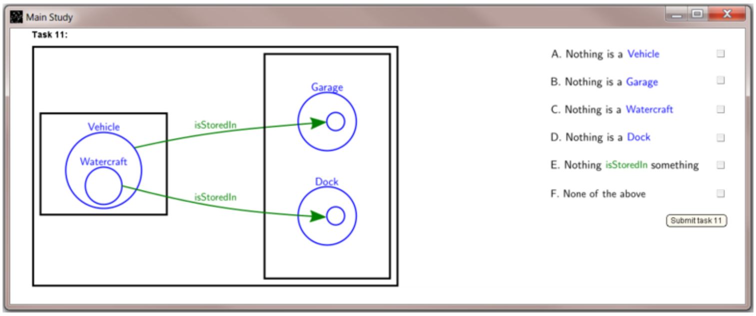

Figure 9. An example of the task displayed in the software

3.5. Study Execution

Participants. A total of 67 students, both undergraduate and postgraduate, were invited to participate in the study. Of these students, 11 were involved in the pilot study and a further 56 for the main study. Students were from both University of Brighton and Sussex Downs College and were studying a variety of courses, none of whom had any previous experience with ontologies. Four of these students were not included in the final analysis of the study as they were unable to provide answers to the majority of questions. Consequently, the study consisted of 63 participants (35 male and 28 female, aged 18 to 41) that were randomly divided into three groups, 21 participants per group.

Procedure. The study was performed in a usability lab free from interruptions and distractions. It consisted of five phases. *Introduction:* The participant was given a brief introduction to the study and asked for his/her informed consent to take part. At this stage, the participant was informed that he/she could withdraw from the study at any time. *Paper-based training:* The participant was presented with a paper-based explanation of the representation and three examples on how to determine whether a concept or property was unsatisfiable, and he/she could keep the explanation sheets with him/her during the study. *Computer-based training:* The participant was given training on how to use the computer software, including performing tasks on two examples (1 containing only concepts and 1 including properties). This phase was an attempt to minimise any learning effects associated with understanding both the software and tasks. The experiment facilitator checked the participant's answers after the training. Where a participant made a mistake, the facilitator explained how to identify the correct answer. Only when the facilitator was happy with the participant's understanding was the next phase started. *Main study:* The participant completed the main tasks on the software and necessary data were collected. *Debrief:* The participant was thanked and given a £6 café voucher for expenses, and told they had access to the results of the study on request.

4. Results and Analysis

4.1. Error Analysis

We recorded two primary variables for each task: the time taken to perform the task, and whether or not the task was completed successfully. Since we view accuracy as more important than speed when interrogating justifications, we first analyse the errors, after which we analyse the timing data.

Each participant was presented with 79 checkboxes across the 14 questions in the main study. They could either check, or not check, each one. Similarly, the correct response for a given checkbox was that it should be checked, or not checked. There are thus 4 different combinations of participant response and correct answer:

1. **Participant correctly checked a checkbox.** The correct response was to check a checkbox, and the participant did that.
2. **Participant incorrectly checked a checkbox.** The correct response was to not check a checkbox, but the participant checked it.
3. **Participant correctly did not check a checkbox.** The correct response was to not check a checkbox, and the participant did that.
4. **Participant incorrectly did not check a checkbox.** The correct response was to check a checkbox, but the participant did not check it.

Responses of type 2 and 4 above were recorded as errors, and those of type 1 and 3 were recorded as non-errors. If there was a time-out for a particular question, this was not recorded as either. We can thus give the overall error rates for each treatment, shown in

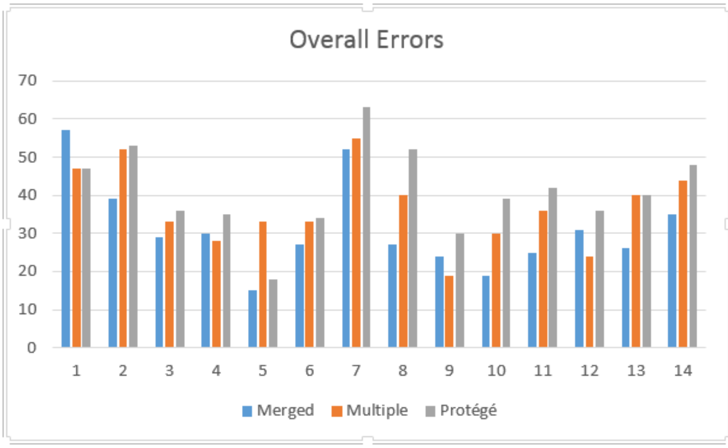

Figure 10. Total errors by group

Table 2. Total error rates by treatment

Treatment	Errors (rate)	Non-errors (rate)
Merged	436(26.6%)	1205(73.4%)
Multiple	514(31.5%)	1116(68.5%)
Protégé	573(34.8%)	1074(65.2%)

Table 3. Error rates for checked boxes

Treatment	Errors (rate)	Non-errors (rate)
Merged	252(52.8%)	225(47.2%)
Multiple	308(65.1%)	165(34.9%)
Protégé	325(67.8%)	154(32.2%)

Table 4. Errors rates for unchecked boxes

Treatment	Errors (rate)	Non-errors (rate)
Merged	184(15.8%)	980(84.2%)
Multiple	206(17.8%)	951(82.2%)
Protégé	248(21.2%)	920(78.8%)

Table 5. Error rates for concepts

Treatment	Errors (rate)	Non-errors (rate)
Merged	259(23.1%)	863(76.9%)
Multiple	300(26.9%)	814(73.1%)
Protégé	355(31.5%)	771(68.5%)

Table 2, and the split by group can be seen in Figure 10. Performing a χ^2-test revealed significant differences ($p < 0.001$) between the expected and actual error rates. Performing pairwise tests revealed a total order on the errors accrued by the three groups. The merged group accrued significantly fewer errors than the multiple group (χ^2 statistic 9.78 with 1 degree of freedom, $p < 0.01$) who in turn accrued significantly fewer errors than the Protégé group (3.919, 1 df, $p < 0.05$). Overall, then, we can say that using diagrams to identify incoherence leads to significantly fewer errors than using Protégé axioms.

We can perform further analyses to find out *where* and *what kind* of errors participants were making. If we look only at those checkboxes which participants checked, we have the error rates as shown in Table 3. Again, performing a χ^2 test revealed significant differences ($p < 0.001$) between the treatments, and pairwise tests revealed that there was a significant difference between the errors accrued by the merged group and the other two groups ($p < 0.001$ for merged vs. multiple and $p < 0.0001$ for merged vs. Protégé), but not between the multiple and Protégé groups ($p = 0.3717$). Broadly speaking, if a participant checked a checkbox, they were correct in doing so half of the time when using the merged diagrams, but only a third of the time using either of the other treatments.

Table 6. Error rates for properties

Treatment	Errors (rate)	Non-errors (rate)
Merged	112(49.1%)	116(50.9%)
Multiple	113(49.8%)	114(50.2%)
Protégé	119(54.3%)	100(45.7%)

Table 7. Error rates for coherent ontologies

Treatment	Errors (rate)	Non-errors (rate)
Merged	65(22.3%)	226(77.7%)
Multiple	101(34.9%)	188(65.1%)
Protégé	89(30.5%)	203(69.5%)

If, instead, we ask whether there is a difference when participants *do not* check a checkbox, we have the results given in Table 4. Performing a χ^2-test revealed significant differences ($p < 0.01$) between the treatments, and pairwise tests revealed there was a significant difference between the errors accrued between the merged group and the Protégé group ($p < 0.001$) and between the multiple group and Protégé group ($p < 0.05$) but there was no difference between the the merged and multiple groups ($p = 0.1982$). In other words, when not checking a checkbox, both diagrammatic groups performed significantly better than the Protégé group.

We also perform analyses depending on the source of the incoherence. We know that, when a set of axioms is incoherent, it could be that a named concept must be empty, or that a property must be empty. Participants were offered both options (whenever an axiom set contained properties), and accordingly checkboxes either related to concepts or properties. We can thus ask whether any treatment made either kind of information more apparent. Firstly, we look at concepts (note that we are now looking at both checked and unchecked boxes). The error rates can be seen in Table 5. The pairwise χ^2-tests revealed a total ordering on groups. The merged group accrued significantly fewer errors than the multiple group ($p < 0.05$), who in turn accrued significantly fewer errors than the Protégé group ($p < 0.05$). Thus, we can say that using diagrams, rather than Protégé, makes recognising unsatisfiable concepts easier.

Secondly, we look at properties. Again, we are now looking at both checked and unchecked boxes, and the rates can be seen in Table 6. The χ^2-tests revealed no significant differences between any of the treatments ($p = 0.4885$). Of note is that the error rates for empty properties over empty concepts are much higher. Regardless of treatment, identifying that a property must be unsatisfiable is a difficult task: no treatment performs better than guessing.

Finally, in some tasks the axiom sets were coherent; for those tasks "None of the above" was the single checkbox which should have been selected. The error rates for the "None of the above" checkboxes (whether checked or not) are in Table 7. The χ^2-test revealed a significant difference ($p < 0.01$) between the treatments, and the pairwise tests show that using merged diagrams leads to significantly fewer errors than using multiple diagrams ($p < 0.001$) or Protégé statements ($p < 0.05$). However, there was no significant difference between the errors accrued by the groups using multiple diagrams or Protégé statements ($p = 0.2510$). In other words, not only does using merged diagrams help identify incoherences, but it helps identifying where there is *not* an incoherence.

4.2. Time Analysis

The mean time taken for participants to answer each question is 54.54 seconds for the merged group, 71.53 seconds for the multiple group, and 65.57 seconds for the Protégé group. In order to fit a model to these data, and to determine significance, we normalised the data by taking the logarithm of time. An ANOVA test was performed

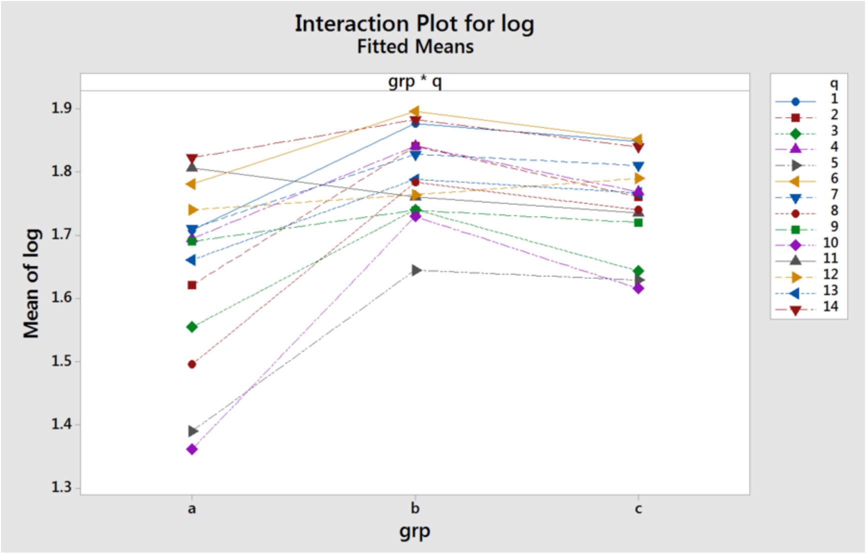

Figure 11. Interaction between question and group for time: (a) is the merged group, (b) is the multiple group, and (c) is the Protégé group.

($F_{(2,13)} = 4.20, p = 0.020$), revealing significant differences (at 95%) between the time taken by the groups. By performing Tukey tests in order to determine significance (at 99%), we find that there is a significant difference between the merged and multiple groups, and merged and Protégé groups, but not between the multiple and Protégé groups. Furthermore, when removing time-outs from the data set, these results still hold.

Certain questions took longer than others. We see the interaction between question number and group in Figure 11. Of particular interest is that only question 11 took longer when using merged diagrams than with the other treatments. The three questions which include only concepts (i.e. no properties) are 5, 8 and 10, where participants performed the task much faster when using merged diagrams.

4.3. Summary of Results

There are two parts to our results: time and error data. For time data, participants using merged diagrams performed the tasks significantly faster than those using multiple diagrams or Protégé statements. For error data, we can say that:

- **Overall:** merged diagrams perform better than multiple diagrams, which perform better than Protégé sentences.
- **Checking a box:** merged diagrams perform better than both multiple diagrams and Protégé sentences, with no difference between the latter two.
- **Not checking a box:** both merged and multiple diagrams perform better than Protégé sentences, with no difference between the former two.
- **Identifying concepts:** merged diagrams performed better than multiple diagrams which in turn performed better than Protégé sentences.
- **Identifying properties:** there were no significant differences between any of the treatments.

- **Identifying lack of incoherence:** merged diagrams performed better than multiple diagrams and Protégé statements, with no difference between the latter two.

In conclusion, we can say that the participants using merged diagrams never performed worse than those using either of the other two treatments, and participants using Protégé statements never performed better than those using diagrams. We can give an affirmative answer to our research question in section 1: visualization of incoherent ontologies makes them easier to understand. In addition, we conclude that a visualization that combines information is more effective than one where axioms are visualized individually.

4.4. Analysis and Discussion

Given the literature on free-rides [7], it is perhaps no surprise that the group using merged diagrams performed significantly better than those using multiple diagrams. When information can be read off a merged diagram, little cognitive effort is needed. For example, where the same labelled curve appears in two different places within a merged diagram, each of which must contain all the individuals in that concept, it is clear that the concept must be necessarily empty. This effect appears in both the time taken to perform, and the errors made when performing, the task. Indeed, when only curves are present in the diagram (questions 5, 8 and 10) it is clear from Figures 10 and 11 that merged diagrams markedly outperform multiple diagrams.

Our results on unsatisfiable concepts (Table 5) suggest that having information in diagrammatic, rather than sentential, form allows people to more easily infer that concepts are unsatisfiable. This effect can be seen both in the time taken to perform, and the errors made when performing, the task. In other words, participants appeared to find it easy to merge information themselves when the axioms were presented as diagrams rather than as sentences. By contrast, there is no improvement in identifying unsatisfiable *properties* through using diagrams over sentences. It could be that people do not have an adequate concept definition of properties and their meaning (either sentential or diagrammatic) to reason about them. Further investigation, just focusing on properties, would be needed to be able to explain this observation.

This study was not designed to determine how clutter affects comprehension. Visual clutter (intuitively, the more “messy” something looks) has been shown to adversely affect understanding in Euler diagrams [15], which are the foundations of concept diagrams. Even allowing for the extra syntax of concept diagrams, some diagrams are more cluttered than others. We note that the two of the more complicated merged diagrams (shown in Figures 2 and 13) gave error rates that were higher than for multiple diagrams, and the cluttered diagram for question 14 (shown on the left in Figure 14) had the longest response time. In certain circumstances, then, it *may* be better to give several, uncluttered diagrams, rather than a merged but cluttered diagram, to aid comprehension. However, a further study focusing on clutter is needed to be able to determine, in a robust manner, whether clutter *does* affect comprehension.

Question 7 (the merged diagram for which is shown in Figure 12[3]) had the highest errors across all questions. Within the merged group, the participants could identify that there was a problem with the property, but they did not then infer a problem with the

[3] Cardinality restrictions on property assertions are not part of the syntax described in [13]. We thank Michael Compton and Gem Stapleton for their contribution of arrow annotations.

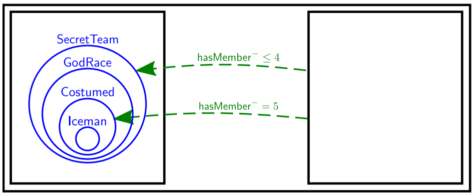

Figure 12. Problematic question.

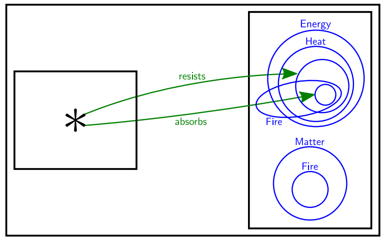

Figure 13. A highly cluttered diagram.

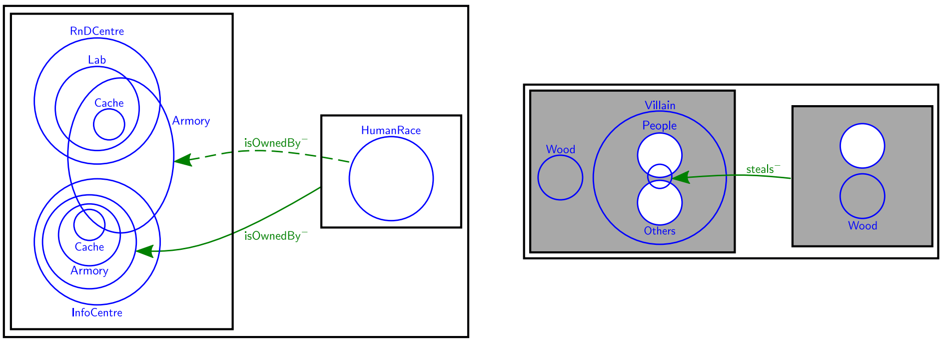

Figure 14. Diagrams with long time responses

two concepts Costumed and Iceman. For the other groups, there was no one reason why people erred. This observation supports our assertion that participants had trouble, in general, with properties. However, it gives further insight in that the merged diagrams forced people to converge on the same wrong (or, more accurately, partial) answer. With this in mind, it could be that further training, or using more expert users, would reduce the error rate when using merged diagrams.

The merged and multiple diagrams for question 11 (the merged shown on the right in Figure 14) both had long response times. Only these diagrams contain shading. It is reasonable to expect a learning effect is in evidence: when only shown shading once, it is natural that participants take longer to comprehend the diagram. It could be that a more effective pattern, not using shading, is needed to display information of this type.

5. Conclusions and Further Work

We sought to answer the question “does visualization of incoherent ontologies make them easier to understand?” In order to do this, we identified common antipatterns from the literature and encoded them using three treatments: the commonly used Protégé statements, multiple concept diagrams, and merged concept diagrams. We have found that, within the limitations of our study and when using merged diagrams, we can give an affirmative answer to the question.

Various observations were made throughout the execution and analysis of the data, which suggest a number of ways to extend the work. We observed that identifying unsatisfiable properties was particularly difficult (when compared with identifying unsatisfiable concepts) across all treatments. Further theoretical and empirical work is needed to identify what makes this task difficult. Furthermore, we observed that more cluttered diagrams appeared to be harder to interpret than less cluttered diagrams. Since this was not the focus of our study, we have performed no analysis of this apparent effect, but it would be fruitful to investigate to help answer the question "when does information become so complex that visualization is no longer beneficial?"

The implementation of the results is another area for exploitation. Creating the multiple diagrams is a simple matter. However, the process of merging diagrams is one which is much more difficult to do automatically. To maximise the impact of our findings, we seek to create a tool that could produce visual justifications using merged diagrams that could be a plug-in for, say, Protégé.

Finally, it is not enough to just *know* that a set of axioms produces an incoherence, we would wish that ontology engineers can effectively debug the ontology. The natural question that arises is thus "is debugging ontologies easier and more reliable when using visualization?"

References

[1] F. Neuhaus, A. Vizedom, K. Baclawski, M. Bennett, M. Dean, M. Denny, M. Grüninger, A. Hashemi, T. Longstreth, and L. Obrst. Towards Ontology Evaluation across the Life Cycle. *Applied Ontology*, 8(3):179–194, 2013.

[2] M. Horridge, S. Bail, B. Parsia, and U. Sattler. The Cognitive Complexity of OWL Justifications. In *The Semantic Web–ISWC 2011*, volume 7031 of *LNCS*, pages 241–256. Springer, 2011.

[3] A. Kalyanpur, B. Parsia, E. Sirin, and J. Hendler. Debugging Unsatisfiable Classes in OWL Ontologies. *Web Semantics*, 3(4):268–293, 2005.

[4] Q. Ji, Z. Gao, Z. Huang, and M. Zhu. Measuring effectiveness of ontology debugging systems. *Knowledge-Based Systems*, 71:169–186, 2014.

[5] H. Stuckenschmidt. Debugging OWL Ontologies - A Reality Check. In *EON-SWSC-2008*, volume 359 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[6] J.-D. Fekete, J. Van Wijk, J. Stasko, and C. North. The value of information visualization. In *Information Visualization*, volume 4950 of *LNCS*, pages 1–18. Springer, 2008.

[7] A. Shimojima. Inferential and Expressive Capacities of Graphical Representations: Survey and Some Generalizations. In *Diagrammatic Representation and Inference*, volume 2980 of *LNAI*, pages 18–21. Springer, 2004.

[8] P. Coppin, J. Burton, and S. Hockema. An Attention Based Theory to Explore Affordances of Textual and Diagrammatic Proofs. In *Diagrammatic Representation and Inference*, volume 6170 of *LNAI*, pages 271–278. Springer, 2010.

[9] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology Visualization Methods—A Survey. *ACM Computing Surveys (CSUR)*, 39(4):10, 2007.

[10] M. Lanzenberger, J. Sampson, and M. Rester. Ontology visualization: Tools and techniques for visual representation of semi-structured meta-data. *J. of Universal Computer Science*, 16(7):1036–1054, 2010.

[11] B. Fu, N. F. Noy, and M.-A. Storey. Indented Tree or Graph? A Usability Study of Ontology Visualization Techniques in the Context of Class Mapping Evaluation. In *The Semantic Web–ISWC 2011*, volume 8218 of *LNCS*, pages 117–134, 2013.

[12] C. Roussey, O. Corcho, and L. M. Vilches-Blázquez. A Catalogue of OWL Ontology Antipatterns. In *Knowledge Capture*, pages 205–206. ACM, 2009.

[13] G. Stapleton, J. Howse, K. Taylor, A. Delaney, J. Burton, and P. Chapman. Towards diagrammatic ontology patterns. In *4th Workshop on Ontology and Semantic Web Patterns*, volume 1188 of *CEUR Workshop Proceedings*, 2013.

[14] J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing Ontologies: A Case Study. In *International Semantic Web Conference*, number 7031 in LNCS, pages 257–272. Springer, 2011.

[15] M. Alqadah, G. Stapleton, J. Howse, and P. Chapman. Evaluating the Impact of Clutter in Euler Diagrams. In *Diagrammatic Representation and Inference*, volume 8578 of *LNAI*, pages 108–122. Springer, 2014.