Proceedings Paper:
Denaux, R, Thakker, DA, Dimitrova, V et al. (1 more author) (2012) Interactive semantic feedback for intuitive ontology authoring. In: Formal Ontology in Information Systems. Proceedings of the Seventh International Conference (FOIS 2012). Frontiers in Artificial Intelligence and Applications. 7th International Conference on Formal Ontology in Information Systems(FOIS 2012), 24-27 Jul 2012, Graz, Austria. IOS Press , 160 - 173. ISBN 978-1-61499-083-3

Interactive Semantic Feedback for Intuitive Ontology Authoring

Ronald Denaux, Dhaval Thakker, Vania Dimitrova and Anthony G. Cohn
School of Computing, University of Leeds,
Woodhouse Lane, Leeds, LS2 9JT, UK
e-mail: {r.denaux,d.thakker,v.dimitrova,a.g.cohn}@leeds.ac.uk

Abstract. The complexity of ontology authoring and the difficulty to master the use of existing ontology authoring tools, put significant constraints on the involvement of both domain experts and knowledge engineers in ontology authoring. This often requires substantial effort for fixing ontologies defects (e.g. inconsistency, unsatisfiability, missing or unintended implications, redundancy, isolated entities). The paper argues that ontology authoring tools should provide *immediate semantic feedback upon entering ontological constructs*. We present a framework to analyse input axioms and provide meaningful feedback at a semantic level. The framework has been used to augment an existing Controlled Natural Language-based ontology authoring tool – ROO. An experimental study with ROO has been conducted to examine users' reactions to the semantic feedback and the effect on their ontology authoring behaviour. The study strongly supported responsive intuitive ontology authoring tools, and identified future directions to extend and integrate semantic feedback.

1. Introduction

Formal ontologies form the basis for the Semantic Web as they structure data for comprehensive and transportable machine understanding [17]. Ontology creation is a challenging task requiring specialism in knowledge engineering, logical background and domain expertise. The active involvement of domain experts is crucial for effective domain modelling but is hard to achieve because domain experts usually lack logical modelling skills. The active involvement of domain experts in ontology authoring becomes even more important now, as there is a growing interest in linked data [8] and a push for iterative, collaborative ontology development that favours reusability [6,9]. In an iterative, collaborative development style, it is important to be aware of the logical implications while contributing or expanding existing facts in order to avoid inadvertently introducing logical defects in ontologies.

There is thus a growing need for ontology authoring tools that provide intuitive interfaces for entering ontology constructs. Recently, this has lead to a stream of research about using Controlled Natural Languages (CNL) for ontology authoring. This research has shown the benefits of using CNL-based tools [19,13,3] that offer intuitive interfaces and abstract the complexities of logical languages such as OWL. There is evidence that domain experts can understand the logical semantics of most CNL-sentences (and thus also of individual OWL axioms) [15,5]. However, our previous research [3], also indicated that authors often do not understand the logical implications of sets of axioms when jointly entered into an ontology. Recent experiences developing ontologies in the context of two European projects [1] show that, as a result of this lack of logical understanding, knowledge engineers have to spend a considerable amount of time validating, debugging and improving the resulting ontologies. There is thus a great opportunity to increase the efficiency and effectiveness of the ontology authoring process by providing interactive, semantic feedback that helps ontology authors to consider relevant logical consequences of the entered facts.

This paper presents a framework which defines how to *understand* input axioms in order to provide appropriate *semantic feedback*. We do this by defining the possible interpretations of input axioms and specify the semantic feedback that can be associated with each interpretation (Section 2). This paper also describes the implementation of the semantic feedback and its application to extend a CNL-based ontology editor (Section 3). Section 5 presents an experimental study which examines users' opinions on the semantic feedback and its impact on ontology authoring behaviour. Finally, we position our work within the relevant literature and, based on the findings from the study, we discuss the broader implications to ontology authoring methods and tools.

2. Semantic Feedback for Input Axioms

2.1. Preliminaries

We assume that an ontology $\mathcal{O}$ is built in a monotonic description logic(DL) language $\mathcal{L}$ [2]. An axiom α is an assertion in $\mathcal{L}$ which is a well-formed formula.

$\mathcal{O}$ is a finite set of axioms in $\mathcal{L}$. C denotes a concept; R denotes a role, a denotes an individual, and Θ denotes a set of concepts. $\top$ denotes the top concept, i.e. every individual is a member of $\top$, $\bot$ denotes the bottom concept, i.e. no individual is a member of $\bot$, and U denotes the top role. $TBox$ represents all axioms in $\mathcal{O}$ which relate concepts to each other. $RBox$ represents all axioms in $\mathcal{O}$ which define role hierarchies and characteristics. $ABox$ represents all axioms which make assertions about individuals.

Λ will be used to denote any set of axioms in $\mathcal{L}$, and a corresponding superscript can be used to assign an additional label to the set. We will use $\models$ to denote that an ontology entails an axiom (i.e. $\mathcal{O} \models \alpha$). $\mathcal{O}^*$ is the deductive closure of $\mathcal{O}$, i.e. the set of axioms in $\mathcal{L}$ that can be entailed by the axioms in $\mathcal{O}$. When an axiom α is entailed from $\mathcal{O}$, we denote with $\mathcal{J}(\alpha, \mathcal{O})$ a **justification** comprising a minimum set of axioms in $\mathcal{O}$ that is sufficient for α to hold. Further definitions and examples of justification are given in [11].

[1]ImREAL (`http://www.imreal-project.eu`) and Dicode (`http://www.dicode-project.eu`). Both of these experiences motivate the work presented in this paper.

[2]In this paper we assume $\mathcal{L}$ is $\mathcal{SROIQ}$ since OWL2 is based on this DL, although our approach is suitable for other DL languages, such as $\mathcal{ALC}$.

2.2. Ontology Defects

When adding an axiom α to $\mathcal{O}$, certain defects may be introduced:

- $\mathcal{O}$ is **inconsistent** when it includes a set of axioms from $ABox$ which contradict with axioms in $TBox$ or $RBox$.
- $\mathcal{O}$ includes a concept C that is **unsatisfiable**, i.e. $\mathcal{O} \models C \equiv \bot$.
- $\mathcal{O}$ includes an axiom α that is **redundant**, i.e. $\mathcal{O} \setminus \{\alpha\} \models \alpha$.
- $\mathcal{O}$ includes **isolated entities**, i.e. $\mathcal{O}$ can have an isolated concept C that only occurs in the axiom $C \sqsubseteq \top$, or an isolated role R that only occurs in the axiom $R \sqsubseteq U$, or an isolated individual occurring only in the axiom $\top(a)$.

Most of the above defects can be detected *explicitly* using DL reasoners and related tools.[3] Inconsistency is considered a defect because it makes further reasoning about the ontology impossible (an inconsistent ontology entails everything). It is possible to reason with ontologies that contain unsatisfiable concepts. However, we consider this a defect, as concepts are usually intended to be satisfiable. Redundant axioms and isolated entities are not necessarily wrong, but they make the ontology cluttered and less concise. Hence, axiom redundancy and entity isolation are considered as bad practice, which should be avoided.

There can also be subjective defects caused by the ontology author's limited understanding of the semantics of the ontology language. Such defects are *impossible to detect* using DL reasoners. Only the authors themselves can discover these defects, e.g. by noticing **unintended inferences** (i.e. inferences that the author considers to be false) or **missing inferences** (i.e. inferences that the author expects to be inferred). Therefore, such defects may be pointed *implicitly* to the ontology author by listing the implications of new axioms.

2.3. Axiom Categories and Semantic Feedback

We now define a semantic analysis strategy that, given a consistent ontology $\mathcal{O}$ and an axiom α in the ontology language $\mathcal{L}$, diagnoses the impact of adding α to $\mathcal{O}$ taking into account the various ontology defects presented above. This will allow *interpreting* the ontology author's input and generating corresponding semantic feedback. The feedback is intended to *inform* authors about potential issues without overwhelming them with too much detail.[4] Because of this, we only show one justification even if there are multiple justifications for an entailment.[5]

Axioms added to an ontology can be either known or novel: α is **known** by $\mathcal{O}$ when $\alpha \in \mathcal{O}^*$, otherwise α is **novel**. Known axioms can be split into two categories:

(A) Already Asserted Axiom
Definition and Detection: $\alpha \in \mathcal{O}$
Feedback: α is already in $\mathcal{O}$.
Defect warning: Adding α to the ontology $\mathcal{O}$ is not needed.

[3]See for example: HermiT (`http://hermit-reasoner.com`), Pellet (`http://clarkparsia.com/pellet`) and [7]

[4]Thus the feedback may not provide all the information necessary to *resolve* the issues.

[5]Feedback that takes into account multiple justifications is left as future work.

(R) Inferred Axiom
Definition and Detection: $\alpha \in \mathcal{O}^* \setminus \mathcal{O}$
Feedback: α is redundant as it can be inferred from $\mathcal{O}$. One possible set of axioms in $\mathcal{O}$ that implies α is the justification $\Lambda^{\text{justification}} = \mathcal{J}(\alpha, \mathcal{O})$.
Defect warning: Adding α to $\mathcal{O}$ causes redundancy. Check the axioms in $\Lambda^{\text{justification}}$.

Adding a novel axiom to an ontology will always lead to an infinite number of further implications. We define a subset of these new inferences Δ_α to represent the finite **set of new relevant implications** brought by adding α to $\mathcal{O}$, i.e.

$\Delta_\alpha = \mathcal{E}_{\mathcal{O}+\alpha} - \mathcal{E}_{\mathcal{O}}$, where $\mathcal{E}_{\mathcal{P}}$ is the set of "relevant axioms" entailed by an ontology $\mathcal{P}$. Relevant axioms are those of the form $A \sqsubseteq B$, $\top \sqsubseteq B$, $A \sqsubseteq \bot$ and $A(a)$ such that A and B are concept expressions[6] that appear in some axiom in $\mathcal{O} \cup \{\alpha\}$ and a is a named individual in $\mathcal{O} \cup \{\alpha\}$. Note that Δ_α is finite, because the set of concept expressions and individuals appearing in $\mathcal{O} \cup \{\alpha\}$ is finite.

A subset of Δ_α that is of particular interest is the set of axioms of the form $C \sqsubseteq \bot$, where C is a named concept, as this set helps us to identify the set of new unsatisfiable concepts: $\Theta^{\text{newUnsatisfiable}} = \{C \mid (C \sqsubseteq \bot) \in \Delta_\alpha\}$.

Adding a new axiom can thus also make the ontology inconsistent or create an unsatisfiable concept:[7]

(I) Novel Axiom Leading to Inconsistency
Definition: $\alpha \notin \mathcal{O}^*$ and $\mathcal{O} \cup \{\alpha\}$ is inconsistent.
Detection: $\mathcal{O} \cup \{\alpha\}$ is inconsistent.
Feedback: α is novel to $\mathcal{O}$ but directly contradicts $\neg\alpha$, which follows, for example, from the existing facts $\Lambda^{\text{justification}} = \mathcal{J}(\neg\alpha, \mathcal{O})$.
Defect warning: Check the axioms in $\Lambda^{\text{justification}}$.

(N) Novel Axiom without new Relevant Implications
Definition: $\alpha \notin \mathcal{O}^*$, $\mathcal{O} \cup \{\alpha\}$ is consistent and $\Delta_\alpha = \emptyset$
Detection: $\Delta_\alpha = \emptyset$
Feedback: α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ does not bring new relevant implications.
Defect warning: If any entailments were expected, α should be reviewed or $\mathcal{O}$ may have to be extended.

(N+) Novel Axiom with new Relevant Implications
Definition: $\alpha \notin \mathcal{O}^*$, $\mathcal{O} \cup \{\alpha\}$ is consistent, $\Delta_\alpha \neq \emptyset$ and $\Theta^{\text{newUnsatisfiable}} = \emptyset$
Detection: $\Delta_\alpha \neq \emptyset$ and $\Theta^{\text{newUnsatisfiable}} = \emptyset$
Feedback: α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ brings the set of new relevant implications Λ_α.
Defect warning: There should be no no missing or unexpected implications in Λ_α.

[6] We use concept expressions here instead of only named concepts. Using only named concepts is easier to compute as there are fewer axioms to be generated. However, new entailed axioms often involve concept expressions and should be reported to the ontology author to maximise the chances of finding uninteded entailments.

[7] Note that the notation $\neg\alpha$ used in the feedback for axiom category **I** is not standard in description logics. We use it to denote either (i) the *opposite of an assertion*: if α is of the form $C(a)$, then $\neg\alpha$ is $\neg C(a)$) or (ii) *a counter example of a T-Box (or R-Box) axiom*: if α is of the form $A \sqsubseteq B$, then $\neg\alpha$ is an axiom of the form $(\neg B \sqcap A)(a)$, where a is an individual in the signature of the reference ontology (an ontology may contain more than one counter example)

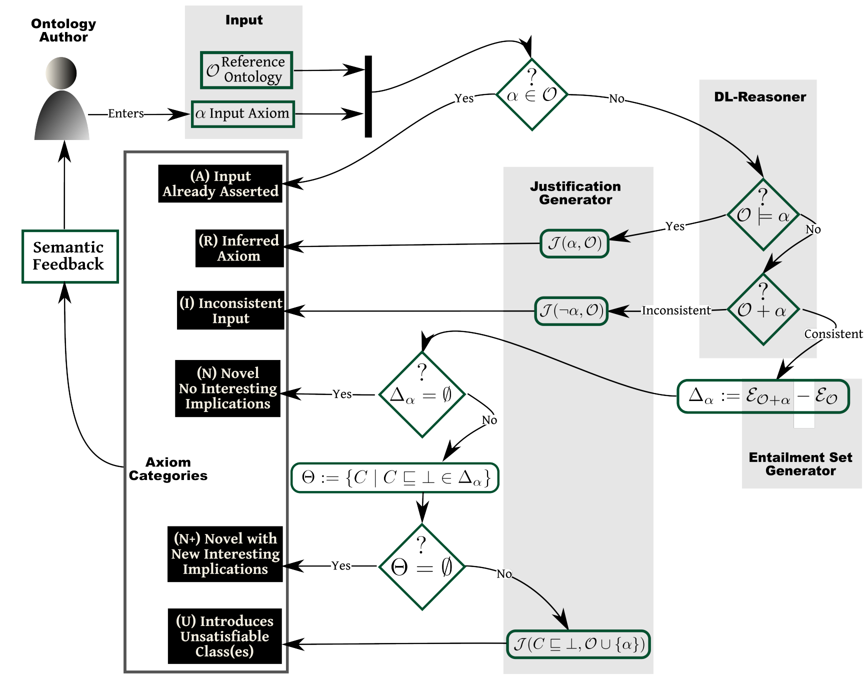

Figure 1. Workflow to categorise axioms and generate semantic feedback.

(U) Axiom Introducing Unsatisfiable Concept
Definition: $\alpha \notin \mathcal{O}^*$, $\mathcal{O} \cup \{\alpha\}$ is consistent and $\Theta^{\text{newUnsatisfiable}} \neq \emptyset$ **Detection:** $\Theta^{\text{newUnsatisfiable}} \neq \emptyset$ **Feedback:** α is novel to $\mathcal{O}$. Adding α to $\mathcal{O}$ makes the concepts $\Theta^{\text{newUnsatisfiable}}$ unsatisfiable. For each concept $C \in \Theta^{\text{newUnsatisfiable}}$, a set of existing axioms that make C unsatisfiable is the justification $\Lambda^{\text{justification}} = \mathcal{J}(C \sqsubseteq \bot, \mathcal{O} \cup \{\alpha\})$. **Defect warning:** Check the axioms in $\Lambda^{\text{justification}}$.

3. Implementation

We implemented the workflow for axiom categorisation depicted in Figure 1 by reusing and extending existing functionality provided by the **OWLAPI**. First, the **OWLAPI** defines an interface for **DL reasoner**s that provides various reasoning services such as entailment and consistency checking (we use HermiT as the reasoner). The OWLAPI is also used to perform the basic operations on axioms and ontologies, like determining whether the input axiom is asserted in the reference ontology. Another interface and a partial implementation [14,11] (which we extended) is for **Justification Generation**. Finally, we extended an **Inferred Axiom Generator**, which combines individuals and concept expressions to find axioms that are entailed by the reference and the merged ontology. This implementation is written in Scala[8] and is available at `http://sf.net/projects/entendre`.

[8] `http://www.scala-lang.org/`

3.1. Adding Semantic Feedback to an Ontology Authoring Tool

We have integrated the semantic feedback functionality in ROO(Rabbit to OWL Ontology authoring) [3]. ROO allows authors to edit ontologies using the Rabbit controlled natural language (a restricted subset of English that can be converted into OWL)[3]. Authors add knowledge in ROO by writing Rabbit sentences using a *Rabbit editor*. ROO provides syntactic feedback to help users compose valid Rabbit sentences. Once the sentence is parsed correctly, the author can accept the sentence, which is converted to OWL axioms and added to the ontology.

We have extended the Rabbit editor in ROO by adding *semantic feedback*. Before an ontology author adds a Rabbit sentence, it is converted to an OWL axiom, analysed as shown in Figure 1 and appropriate semantic feedback is generated.[9] Textual explanations are provided following the definitions in Section 2.3 and using pre-defined templates. Hereafter, we refer to this updated version of ROO. The semantic feedback provided by ROO is illustrated below.[10]

Rabbit **Input:** *Every Student Accommodation is a kind of Accommodation.*

Axiom category: N [Novel Axiom without new Relevant Implications]

ROO **Feedback:** This assertion is **novel**: it has not been added to the ontology yet. However, **no new interesting facts** could be inferred based on this input. This generally means that the terms you are using have been loosely defined in the ontology. It is also possible that there are some implications that could not be found.

Advice Add the sentence and maybe add more new definitions if you want the ontology to be able to make more inferences.

Rabbit **Input:** *Every Teaching Hospital is a kind of Hospital.*

Axiom category: N+ [Novel Axiom with new Relevant Implications]

ROO **Feedback:** This assertion is **novel**: it has not been added to the ontology yet. This input **implies** 6 new relevant facts. Have a look at the list of implications to make sure you agree with the implications. If you do not agree, it may be that you are using the wrong terminology.

Check the new implications:

- *Every Teaching Hospital has footprint a Footprint.*
- *Organisation and Teaching Hospital are mutually exclusive.*
- *Training Centre and Teaching Hospital are mutually exclusive.*
- *Every Teaching Hospital is a kind of Topographic Object.*
- *Every Teaching Hospital is a kind of Place.*
- *Teaching Hospital and University (Institution) are mutually exclusive.*

Rabbit **Input:** *Every University has part at least one Point of Interest.*

Axiom category: R [Inferred Axiom]

ROO **Feedback:** This statement is **redundant**. Although the sentence itself is not included in the ontology, this sentence can be inferred from the following sentences that have been said to be true.

Note that existing Rabbit statements imply that *Every University has part at least one Point of Interest*:

- *Every University has part a Building that has purpose Education of University Students.*
- *Every Building is a kind of POI.*
- *POI and Point of Interest are equivalent.*

[9] ROO uses the feedback to *advise* on a course of action, not to *enforce* it.
[10] We omit Asserted axioms due to space constraints and because the feedback is trivial.

Rabbit **Input:** *Every Student Union is contained within a University(Institution).*
Axiom category: U [Axiom Introducing Unsatisfiable Concepts]
ROO **Feedback:** This sentence makes concept `Student Union` **unsatisfiable!** This means that nothing can be a `Student Union` anymore. **Advice** You should not add an unsatisfiable concept to an ontology because this concept becomes practically unusable. This is especially true if you make a concept unsatisfiable and that concept was defined by somebody else, as you are probably not using the concept in the way it was intended. **Check some existing sentences contradicting input**: • *Organisation and POI are mutually exclusive.* • *Every University (Institution) is a kind of Organisation.* • *The relationship contains must have subject POI* • *The relationship is contained within is the inverse of contains.*

Rabbit **Input:** *Edge contains a Swimming Pool.*
Axiom category: I [Axiom Leading to Inconsistency]
ROO **Feedback:** This sentence makes the ontology *inconsistent!* This means that this sentence contradicts what has been said in other sentences in the ontology. **Advice**: You should never enter a sentence that makes the ontology inconsistent because: • it is very hard for computers to reason about inconsistent ontologies • You are likely using a term (concept, relation or instance) in a way that was not intended by the people who defined the ontology. You should probably find an alternative term that you can use instead. See also the provided list of sentences that contradict this sentence. **Check some existing sentences that contradict input**: • *Edge is contained within UoL Campus.* • *UoL Campus does not contain a Swimming Pool.* • *The relationship contains is transitive.*

4. Experimental Study

An experimental study with ROO was conducted to examine users' reactions to semantic feedback and whether this feedback affected users' ontology authoring behaviour. The following *research questions* were addressed: *How did users characterise the semantic feedback provided by ROO?, Did users find the semantic feedback helpful and for what? Did users understand the logical aspects indicated in the semantic feedback provided by ROO?*

Domain and Ontology ($\mathcal{O}$). The study followed a task-based approach which involved using ROO to add new axioms to an ontology $\mathcal{O}$. Points of interest (POI) was chosen as the domain because of its increasing importance, broad application, and familiarity to people. $\mathcal{O}$[11] was created by reusing the W3C POI data model[12] and Ordnance Survey's Buildings and Places ontology[13] using ROO by entering Rabbit sentences (see [3]). $\mathcal{O}$ described main points of interest relevant to Leeds

[11] available at `http://www.comp.leeds.ac.uk/confluence/Entendre-Study`
[12] http://www.w3.org/2010/POI/
[13] http://www.ordnancesurvey.co.uk/oswebsite/ontology/

University, including buildings and places related to accommodation, eating and drinking, health services, and transport.

Participants. The study involved 10 participants recruited on a volunteering basis; 6 were from the School of Computing, Leeds University, and 4 were from outside. All participants had general IT background and used computers as part of their everyday practice. They were grouped into: **Group 1 KE-novices** (5 users who had no logical background and had never been involved in ontology construction tasks) and **Group 2 KE-experts** (5 users who had logical background or had built ontologies as part of their research).

Procedure and Materials. All evaluation materials and data from the experimental study are available online.[14] Each participant had an individual session observed by an experimenter (from the first three authors). Each session comprised of two steps. In **Step 1** [*5 to 10 minutes*] the participants were given a list of classes, instances, and relationships from $\mathcal{O}$ to examine in ROO. In **Step 2** [*60–90 minutes*] the participants were asked to enter new facts, formulated as Rabbit sentences([3]). There were 15 sentences in 3 batches; a batch included examples from each of the axiom categories : N, N+, R, I, and U (defined in Section 2.3). After entering a sentence in ROO, the participants were asked to press the *Semantic Feedback* tab, read the provided feedback, answer a series of questions about their opinion on the feedback, and indicate whether they would add or discard the sentence.

Data. The collected data included the participants' answers related to each sentence and the observers' notes. The analysis is presented below.

4.1. Participants' opinions about semantic feedback.

Participants stated their opinion about feedback by selecting characteristics from a given list: informative, relevant, trustworthy, reassuring, confusing, overwhelming and misleading. Table 1 presents a summary of all sentences, the five axiom categories are compared in figure 2, and the two user groups – in figure 3.

	Info	**Relev**	**Trust**	**Reass**	**Conf**	**Over**	**Misl**	**Help**	**NotSure**	**NotHelp**
Overall	78%	56%	38%	16%	10%	10%	1%	91%	8%	1%

Table 1. Summary of the participants' opinion on semantic feedback.

Overall, the feedback was found **informative** and **relevant** by all participants. KE-experts found the feedback **reassuring** mainly because their *assumption* about what impact an axiom would have on the ontology was confirmed. KE-novices found feedback reassuring in few cases (9%), mainly for *novel with new relevant implications* axioms, as the feedback helped them decide to add the axioms. In more than one third of the cases users found the semantic feedback **trustworthy**. KE-novices trusted the feedback more often, and saw it as a *crutch* to give support when they were unsure. KE-experts in many cases preferred to double-check everything themselves, although they did find the feedback very informative.

[14] http://www.comp.leeds.ac.uk/confluence/Entendre-Study/

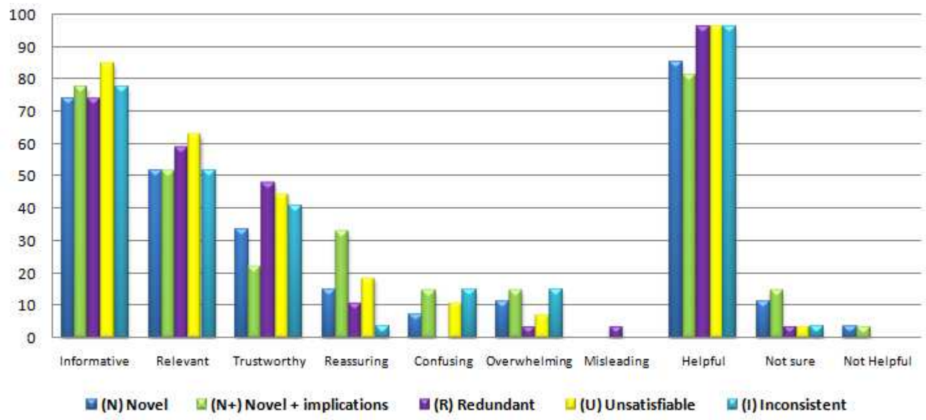

Figure 2. Participants' opinions on feedback distributed over axiom categories, together with participants' ranking of feedback as 'Helpful', 'Not sure' or 'Not helpful'. The values are percentages based on all messages from each axiom category.

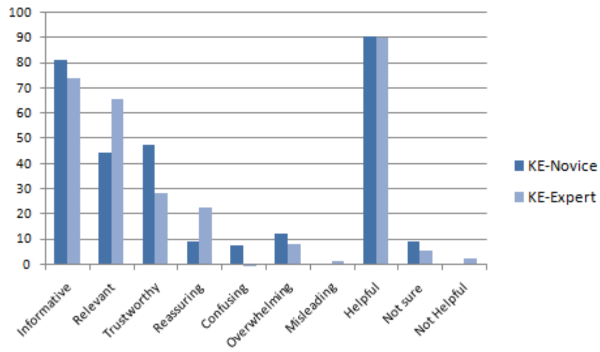

Figure 3.: Summary of the opinions on feedback for the two groups. The values are percentages based on all messages from each axiom category for the corresponding group.

There was only one case of **misleading** feedback indicated by a KE-expert. The KE-expert pointed out that although it was possible to infer one of the axioms representing cardinality constraints from the (R) category of the existing ontology, feedback should not encourage the user to discard it, as it would still be valuable to state cardinality constraints explicitly. We also analysed the cases when participants found feedback **confusing** or **overwhelming** and notice that confusion seems to decrease with time: the first batch of 5 sentences includes most misleading and confusing cases, while the last batch has only one occurrence. The analysis indicated that the confusing and misleading feedback was associated with certain sentences. These sentences could be grouped in: **Confusing terminology:** KE-novices found feedback about axiom category unsatisfiable (U) hard to follow. They understood that the sentence should not be entered but were confused about what else to do. We plan to work on the usability aspects of the semantic feedback and will consider improving the feedback terminology. **Too abstract:** This concerned mainly feedback containing justifications (R, N+, I and U); users from both groups felt that certain abstract concepts (e.g. Footprint) made it more difficult to understand the feedback message and suggested that appropriate filtering should be done. **Oversimplified:** This was pointed by KE-experts who felt that advice to not enter sentences which made the ontology inconsistent was inappropriate (the sentences were considered valuable, and should have been entered; instead existing sentences should be edited). **Insufficient:** Two KE-experts pointed out that in the cases when the novel(N) axioms could make existing axioms redundant (e.g. entering *Every student bus route is a kind of bus route* made the existing axiom *Every student bus route is a kind of transport route* redundant but this was not detected) the feedback was insufficient. ROO can be improved to consider this form of redundancy.

4.2. Helpfulness of Semantic Feedback

For every feedback message, participants were asked to indicate whether the feedback was helpful and to clarify why. The results, summarised in table 1, are very encouraging, as 91% of feedback messages were acknowledged to be helpful. Further analysing (figure 2) results by considering the axiom categories, there are notable differences between both groups.

Group 1 - KE-novices - considered feedback as: (a) **Providing new information**, which they did not know (e.g. *'Tells me that a new fact may have impact on the ontology(category N+)'*, *'Informed me about the ontology and the links between the concepts (category R)'*; (b) **Preventing ontology defects**, e.g. *'Told me about inconsistency, I would have not checked otherwise(category I)'*, *'The feedback explained why the sentence should not be included (category U)'*); and (c) **Providing hints on what to do next**, which was mostly for novel facts without relevant implication, as feedback pointed out that further connections should be entered (e.g. *'I have the hint that something may be missing (category N)'*.

Group 2 -KE-experts - found the feedback helpful for: (a) **Developing ontology awareness**, the users found additional information about the ontology provided with the contradicting sentences (category I), implications (category N+), or sentences which make an axiom redundant (category R) useful to gain awareness of the ontology *'(getting the right information at the right time)'*; (b) **Providing warnings**, when something may be overlooked, as one participant commented *'Helps keeping the ontology foolproof (category I).'*; (c) **Providing assurance**, when the KE-experts knows what may happen, (e.g. *'Gives me assurance that I was right in the first place (category R)'*); and (d) **Facilitating decision making**, when further action is needed, e.g. information about contradicting sentences was considered helpful (e.g. *'Directed me what to change from the ontology (category I)'*).

Participants were not sure of the helpfulness of feedback when: (a) it was confusing or misleading (see previous section); (b) did not provide much new information (which was pointed by KE-experts); (c) did not provide enough information what to do next. The two occurrences of *Not helpful* were on novel axioms category and came from the same KE-expert who commented that, apart from telling that a sentences was novel, the feedback was not much useful.

4.3. Understanding of Logical Aspects and Impact on User behaviour

For every sentence to be entered, participants were asked three questions to test their understanding of the logical implications relating to the sentence. To avoid asking questions that followed trivially from the feedback given, the questions used rephrasings and slightly different terminology as that used in the feedback. For example, for category N+, we asked whether "the ontology *already knew* the fact that 'X"', where X was one of the presented new entailments. Some questions also inverted the information given in the feedback: for category I, we asked whether the opposite of the input sentence was already known.

We classified the score for each participant's answers to indicate the level of awareness about logical implications: *confusion*, *neutral* or *understanding*. Overall, both groups showed a high level of understanding: 69% for **Group1(KE-novices)** and 86% for **Group 2(KE-experts)**. There are notable differences between both groups (see Table 2). Particularly surprising was that **Group 2(KE-experts)** showed signs of confusion when answering some of the questions. We note however that, in the case of axioms leading to inconsistency (I) this apparent confusion matches with this group's opinion that the advice was oversimplified.

	Group1	KE-novices		**Group2**	KE-experts	
	Confusion	**Neutral**	**Understanding**	**Confusion**	**Neutral**	**Understanding**
N	0%	15%	85%	0%	7%	93%
N+	0%	15%	85%	7%	0%	93%
R	8%	38%	54%	0%	0%	100%
I	8%	31%	62%	21%	14%	65%
U	15%	8%	77%	7%	14%	79%
Overall	6%	25%	69%	7%	7%	86%

Table 2. Participants' understanding of logical aspects in feedback.

We also reviewed participant responses for measuring the impact of semantic feedback on their behaviour. The experiment study included a question on what actions the participants would take in response to the semantic feedback. The possible answers were: they will (a) add the sentence to the ontology, (b) discard the sentence or try to find an alternative sentence, (c) seek further clarification and (d) do not know. We analysed the answers (user actions) and compared them against the advice from the feedback. The results are very encouraging as **Group 1(KE-novice users)** accepted 96% of advice compared to the 92% for the **Group 2(KE-experts)**, i.e., participants agreed with the advice and followed the action suggested by the semantic feedback message. In the few cases that the advice would not be followed, KE-novices often indicated that they would seek further advice, which indicates that they had been made aware of the problem at hand, but did not have enough information to resolve the situation.

5. Related Work

Research on *ontology debugging* has focused on pinpointing the cause of some defects (inconsistency, unsatisfiability and redundancy) and creating repair plans by using one or all of the justifications for the defects [11,7]. These tools focus on aiding knowledge engineers to find ways to fix ontologies that are already defective. By contrast, the work presented in this paper focuses on *preventing the ontology from becoming inconsistent in the first place*. Also, because we focus on providing interactive feedback, we only provide enough feedback to make ontology authors aware of ontology defects [15].

Recent work focuses on entailment justifications – how to make them more concise [11], predicting their cognitive complexity [10] and analysing the justificatory structure of ontologies [1] – because automatically fixing defective ontologies can be difficult or impossible. Work on justifications is complementary to this work as it helps ontology authors to discover and resolve defects by aiding them to inspect justifications but it first *requires authors to be aware of (i) the possible defects they are introducing and (ii) the relevant inferences they need to inspect.*

Current ontology editors such as Protégé help authors to become aware of some relevant entailments: when browsing a concept `C`, one can see entailments relating to `C`. However, the feedback of current editors does not show dynamically (with the exception of early work on OntoTrack [16]) how adding or removing axioms affect these relevant entailments. E.g. adding an axiom about `C` may cause unexpected entailments that do not involve concept `C`; such entailments will not be visible. Our proposed approach provides a way to focus the author's attention on the relevant entailments.

Work on *Ontology Integration and Revision* aims to help knowledge engineers discover the impact of merging of – typically large numbers of automatically created – assertions into existing ontologies[12,18]. These approaches are geared towards knowledge engineers who already are aware of the issues at hand. By contrast, this work focuses on providing feedback that is suitable for novice ontology authors and that applies to the integration of single axioms.

Other related work includes SuperModel [2], a tool that aims to give authors a better understanding of the OWL semantics by allowing the exploration of generated models. The key limitation of this approach is that the model exploration is a separate activity to ontology authoring and it is not clear whether the generated models provide realistic or relevant examples. This limitation is overcome by the approach presented here.

6. Discussion and Conclusions

The presented work addresses a pressing need for providing intuitive means to aid ontology authoring. Current tools offer little feedback to prevent the introduction of ontology defects and focus instead on the diagnosis and resolution of existing problems. The research presented here takes the next step to intuitive ontology authoring – embedding *intelligence* in the ontology authoring tools to understand the user's actions and respond accordingly. The study helped us examine the feasibility of this approach regarding users with different KE skills and enabled us to draw broader implications for ontology authoring.

Responsive ontology authoring tools. The study strongly indicated support for the philosophy that *authoring tools can act as active listeners* that offer im-

[15]Finding repair plans can be computationally expensive and is only useful if the ontology authors understand the problem at hand, thus we view this as an additional step.

mediate, interactive, and intuitive feedback at the time a new axiom is to be added. It showed: (a) it is possible to develop such tools (ROO is just an example – the feedback features could be embedded in any ontology authoring tool); and (b) users are enthusiastic about such tools. All KE-experts in the study reacted extremely positively to the *embedded* feedback and commented that it would potentially save them substantial time and effort to maintain the ontology. All KE-novices were also pleased to see immediate response to their actions and followed advice in 96% of the cases. Some users commented that feedback helped them to consider what was required when authoring an ontology, and even suggested that ROO would be useful to assist people learn about ontology authoring. This can be addressed in further studies with ROO(e.g. using it as a learning tool in BSc or MSc courses on knowledge engineering).

Easier to understand than to be understood. The study shows that it is not sufficient only to understand the users but also to *make users understand what is conveyed with the feedback*. During the study we asked participants questions testing their understanding of logical implications. KE-novices understood the logical implications described in the feedback in 69% of the cases, while KE-experts in 80%. Although there was noise in few questions, it was clear that feedback could cause *confusion* (10% of all cases), might be *overwhelming* (10%), and, in one case, was *misleading*. The analysis of these results points to further work required to improve the effectiveness of the provided feedback. The confusing terminology of the feedback messages(e.g. unsatisfiable class) can be improved and made more intuitive. Avoiding concepts or axioms which are *too abstract* requires further research to define and infer *abstractness* (e.g. being in the upper level of the ontology). Furthermore, some *strategy to filter and order* the list of inferred or contradicting axioms would be needed (e.g. following dialogue approaches to maintain focus or visualisation approaches to show context). Some participants suggested a *'traffic light'* approach, using visual signals to show when attention is needed, a problem is about to occur, or further axioms would be required. This could be beneficial for all types, but the level of detail would have to vary (as pointed out above). Some users pointed out that the additional information about the ontology (implications and contradictions) could be presented in a *bite-size* way, starting with the most relevant and moving to less relevant.

Future work. Our immediate future work is to improve the semantic feedback (as pointed above), as well as to merge the feedback mechanism to also take into account syntactic problems of the CNL input. This will provide a systematic approach to generate feedback when entering ontological constructs in CNL (or another ontology authoring language). We have integrated such support in the previous version of ROO[3], the effectiveness of which was confirmed with an evaluation study [4]. By taking into account the syntactic understanding, this approach would be suitable to applications which *interact* with a user based on a reference ontology. We intend to use this in ROO by adding a *dialogue agent* that can support the knowledge elicitation and ontology authoring process. It should be noted that the current implementation relies heavily on ontology reasoners and is dependent on their scalability (currently, working well only for small or medium size ontologies). Modularisation strategies, widely used to address scalability, may also be helpful to handle scalability problems with the axiom analysis, e.g. providing reasoning based on a relevant subset of the whole ontology.

Acknowledgements The PhD studies of the first author are funded by the Boothman, Reynolds & Smithells scholarship at the University of Leeds and the Ordnance Survey. The work is partially supported by the European Union Seventh Framework Programme under grant agreement no ICT 257831 (ImREAL project) and grant agreement no ICT 257184 (DICODE project). We thank the participants in the experimental study.

References

[1] S. Bail, M. Horridge, B. Parsia, and U. Sattler. The justificatory structure of the NCBO bioportal ontologies. *10th International Semantic Web Conference*, pages 67–82, Oct. 2011.

[2] J. Bauer, U. Sattler, and B. Parsia. Explaining by Example: Model Exploration for Ontology Comprehension. *Description Logics*, 2009.

[3] R. Denaux, C. Dolbear, G. Hart, V. Dimitrova, and A. G. Cohn. Supporting domain experts to construct conceptual ontologies: A holistic approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, Feb. 2011.

[4] V. Dimitrova, R. Denaux, G. Hart, C. Dolbear, I. Holt, and A. G. Cohn. Involving Domain Experts in Authoring OWL Ontologies. *In Proceedings of the 7th International Semantic Web Conference, ISWC*, pages 1–16, 2008.

[5] P. Engelbrecht, G. Hart, and C. Dolbear. Talking rabbit: a user evaluation of sentence production. *Proceedings of the 2009 conference on Controlled natural language*, pages 56–64, June 2010.

[6] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research*, 31(1), 2008.

[7] S. Grimm and J. Wissmann. Elimination of Redundancy in Ontologies. In *8th Extended Semantic Web Conferenence*, volume 6643 of *Lecture Notes in Computer Science*, pages 260–274. Springer, 2011.

[8] J. Hendler. Why Semantic Web will never work, 2011.

[9] C. W. Holsapple and K. D. Joshi. A collaborative approach to ontology design. *Communications of the ACM*, 45(2):42–47, Feb. 2002.

[10] M. Horridge, S. Bail, B. Parsia, and U. Sattler. The Cognitive Complexity of OWL Justifications. *International Semantic Web Conference*, 2011.

[11] M. Horridge, B. Parsia, and U. Sattler. Justification oriented proofs in OWL. *9th international semantic web conference on The semantic web*, pages 354–369, Nov. 2010.

[12] E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga. Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. *Proceedings of the 6th European Semantic Web Conference*, 5554:173–187, May 2009.

[13] K. Kaljurand and N. Fuchs. Bidirectional mapping between owl dl and attempto controlled english. *Principles and Practice of Semantic Web Reasoning*, pages 179–189, 2006.

[14] A. Kalyanpur, B. Parsia, E. Sirin, and J. Hendler. Debugging unsatisfiable classes in OWL ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4):268–293, Dec. 2005.

[15] T. Kuhn. An Evaluation Framework for Controlled Natural Languages. In N. E. Fuchs, editor, *Proceedings of the Workshop on Controlled Natural Language (CNL 2009)*, volume 5972 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2010.

[16] T. Liebig and O. Noppens. OntoTrack: A semantic approach for ontology authoring. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):116–131, 2005.

[17] A. Maedche and S. Staab. Ontology learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, Mar. 2001.

[18] N. Nikitina, B. Glimm, and S. Rudolph. Wheat and Chaff – Practically Feasible Interactive Ontology Revision. *10th International Semantic Web Conference*, 7031:487–503, 2011.

[19] R. Schwitter. Creating and Querying Formal Ontologies via Controlled Natural Language. *Applied Artificial Intelligence*, 24(1):1–26, 2010.