An Evaluation Framework for Controlled Natural Languages[⋆]

Tobias Kuhn

Department of Informatics & Institute of Computational Linguistics,
University of Zurich, Switzerland
tkuhn@ifi.uzh.ch
http://www.ifi.uzh.ch/cl/tkuhn

Abstract. This paper presents a general framework called *ontographs* that relies on a graphical notation and enables the tool-independent and reliable evaluation of human understandability of knowledge representation languages. An experiment with 64 participants is presented that applies this framework and compares a controlled natural language to a common formal language. The results show that the controlled natural language is easier to understand, needs less learning time, and is more accepted by its users.

1 Introduction

Controlled natural languages (CNL) have been proposed for the area of knowledge representation, and specifically for the Semantic Web, in order to overcome the problem that common formal languages are often hard to understand for people unfamiliar with formal notations [17, 16]. User studies are the only way to verify whether CNLs are indeed easier to understand than other languages.

I propose here a novel approach for evaluating the understandability of CNLs. My approach relies on a graphical notation that I call *ontographs*. It allows for testing CNLs in a tool-independent way and for comparing them to other formal languages. The ontograph approach has been outlined in the extended abstract of this full paper [12]. Therein, the results of a first experiment are described. Here, I describe a second, more thorough experiment that has been conducted in the meantime. Both experiments and their results are described in more detail in my doctoral thesis [13].

Existing approaches to evaluate CNLs can be subdivided into two categories: task-based and paraphrase-based approaches. I will argue that it is difficult to get reliable results concerning the understandability of CNLs with either approach.

[⋆] This work was funded by the research grant (Forschungskredit) programs 2006 and 2008 of the University of Zurich. I would like to thank Alain Cohn, Norbert E. Fuchs, Kaarel Kaljurand, Marc Lutz, Cerstin Mahlow, Philipp Müller and Michael Piotrowski for their suggestions and corrections. Furthermore, I would like to acknowledge the Institute for Empirical Research in Economics of the University of Zurich for organizing the recruitment of the participants for the experiment.

1.1 Task-based CNL Experiments

In task-based experiments, the participants are given specific tasks to be accomplished by entering CNL statements into a given tool. One such experiment was described by Bernstein and Kaufmann [1], and another one was presented by Funk et al. [7, 6]. In both cases, the participants of the experiment received tasks to add certain knowledge to the knowledge base using a tool that is based on a CNL.

An example taken from [7] is the task "Create a subclass *Journal* of *Periodical*" for which the participants are expected to write a CNL statement in the form of "Journals are a type of Periodicals". To evaluate whether the task is accomplished, the resulting knowledge base can be checked whether it contains this actual piece of information or not. This approach bears some problems if used to test the understandability of a language.

First of all, such experiments mainly test the ability to write statements in the given CNL and not the ability to understand them. A user succeeding in the task shown above does not necessarily understand what the statement means. In order to add "Journal" as a subclass of "Periodical", the user only has to map "subclass" and "type of", but does not have to understand these terms.

Another problem is that it is hard to determine with such experiments how much the CNL contributes to the general usability and understandability, and how much is due to other aspects of the tool. It is also hard to compare CNLs to other formal languages with such studies, because different languages often require different tools. For these reasons, it would be desirable to be able to test the understandability of CNLs in a tool-independent way, which is not possible with task-based approaches. However, such approaches seem to be a good solution for testing the *writability* of CNLs.

1.2 Paraphrase-based CNL Experiments

Paraphrase-based approaches are a way how CNLs can be tested in a tool-independent manner. In contrast to task-based approaches, they aim to evaluate the comprehensibility of a CNL rather than the usability of tools based on CNL.

Hart et al. [10] present such an approach to test their CNL (i.e. the Rabbit language). The authors conducted an experiment where the participants were given one Rabbit statement at a time and had to choose from four paraphrases in natural English, only one of which was correct. The authors give the following example of a Rabbit statement and four options:

Statement: Bob is an instance of an acornfly.
Option 1: Bob is a unique thing that is classified as an acornfly.
Option 2: Bob is sometimes an acornfly.
Option 3: All Bobs are types of acornflies.
Option 4: All acornflies are examples of Bob.

They used artificial words like "acornfly" in order to prevent that the participants classify the statements on the basis of their own background knowledge. Option 1 would be the correct solution in this case. Similar experiments are described by Hallett et al. [8] and Chervak et al. [4]. Again, there are some problems with such approaches.

First of all, since natural language is highly ambiguous, it has to be ensured somehow that the participants understand the natural language paraphrases in the way they are intended, which just takes the same problem to the next level. For the example above, one has to make sure that the participants understand phrases like "is classified as" and "are types of" in the correct way. The problem is even more complicated with words like "unique" and "sometimes". If one cannot be sure that the participants understand the paraphrases then the results do not permit any conclusions about the understandability of the tested language.

Furthermore, since the formal statement and the paraphrases look very similar in many cases (both rely on English), it is yet again hard to determine whether understanding is actually necessary to fulfill the task. The participants might do the right thing without understanding the sentences (e.g. just by following some syntactic patterns), or by misunderstanding both — statement and paraphrase — in the same way. For the example above, a participant might just think that "an instance of" sounds like having the same meaning as "a unique thing that is classified as" without understanding any of the two. Such a person would be able to perform very well on the task above. In this case, the good performance would imply nothing about the understandability of the tested language.

Nevertheless, paraphrase-based approaches also have their advantages. One of them is that they scale very well with respect to the expressivity of the language to be tested. Basically, CNLs built upon any kind of formal logic can in principle be tested within such experiments, once the problems identified above are solved in one way or another.

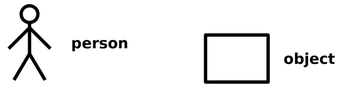
2 The Ontograph Framework

In order to overcome the discussed problems of existing approaches, I propose a novel, diagram-based approach to test the understandability of languages. It relies on a graphical notation called *ontographs* (a contraction of "ontology graphs"). This notation is designed to be very simple and intuitive. The basic idea is to describe simple situations in this graphical notation so that these situation descriptions can be used in human subject experiments as a common basis to test the understandability of different formal languages. This approach allows for designing reliable understandability experiments that are completely tool-independent.

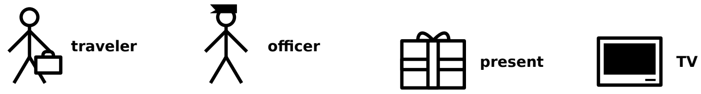
2.1 The Ontograph Notation

Every ontograph diagram consists of a legend that introduces types and relations and of a mini world that describes the actual individuals, their types, and their relations.

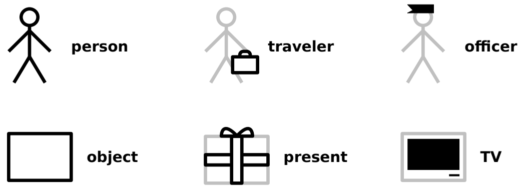
The legend of an ontograph introduces the names and the graphical representations of types and relations. Types are introduced by showing their name beside the symbol that represents the respective type. For example, introducing a type "person" and another type "object" can be done as follows:



Starting from such general types, more specific ones can be defined. For example, "traveler" and "officer" can be defined as specific types of the general type "person", and "present" and "TV" can be defined as specific types of "object":

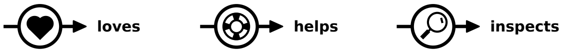


If a legend contains a type like "person" and, at the same time, a specific type like "traveler" then the part of the symbol of the specific type that is copied from the general type is displayed in gray:

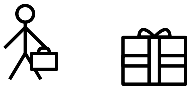


The purpose of this is to specify that only the black part of the symbol represents the respective type. This becomes important for individuals that belong to several types. For example, the suitcase is the deciding criterion in the case of the "traveler" definition (and not e.g. the missing hat).

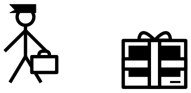
Relations are represented by circles that contain a specific symbol with an arrow going through this circle. As with types, the legend introduces the names of the relations by showing them on the right hand side of their graphical representation. Some examples are the relations "loves", "helps" and "inspects":



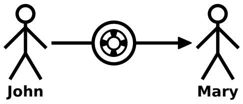
In contrast to the legend that only introduces vocabulary and the respective graphical representations, the mini world describes actual situations. Such situations consist of individuals, the types of the individuals, and the actual relations between them. Every individual is represented by exactly one symbol indicating the types of the individual. For example, an individual that is a traveler and another individual that is a present are represented by a traveler symbol and a present symbol:



An individual that belongs to more than one type is represented by a combined symbol that is obtained by merging the respective symbols. For example



represents an individual that is a traveler and an officer and another individual that is a present and a TV. Individuals can optionally have a name, which is shown in the mini world below the respective symbol. Relation instances are represented by arrows that point from one individual to another (or the same) individual and that have a relation symbol somewhere in the middle. "John helps Mary", for example, would be represented as follows:



There is no explicit notation for negation. The fact that something is not the case is represented implicitly by not saying that it is the case. For example, stating that "John does not help Mary" is done by *not* drawing a *help*-relation from John to Mary. Thus, mini worlds are closed in the sense that everything that is true is shown and everything that is not shown is false.

Mini world and legend are compiled into a complete ontograph. The mini world is represented by a large square containing the mini world elements. The legend is represented by a smaller upright rectangle to the right of the mini world and contains the legend elements. Figure 1 shows an example.

2.2 Properties of the Ontograph Notation

The ontograph notation has some important characteristic properties. First of all, the ontograph notation does not allow for expressing incomplete knowledge. This means that nothing can be left unspecified and that every statement about the mini world is either necessarily true or necessarily false. For example, one can express "John helps Mary", or one can also state "John does not help Mary", but one cannot express that it is unknown whether one or the other is the case. Most other logic languages (e.g. standard first-order logic) do not behave this way. For the ontograph notation, this has the positive effect that no explicit notation for negation is needed.

Another important property of the ontograph notation is that it has no generalization capabilities. Logically speaking, the ontograph notation has no support for any kind of quantification over the individuals. For example, one cannot express something like "every man loves Mary" in a general way. The only way to

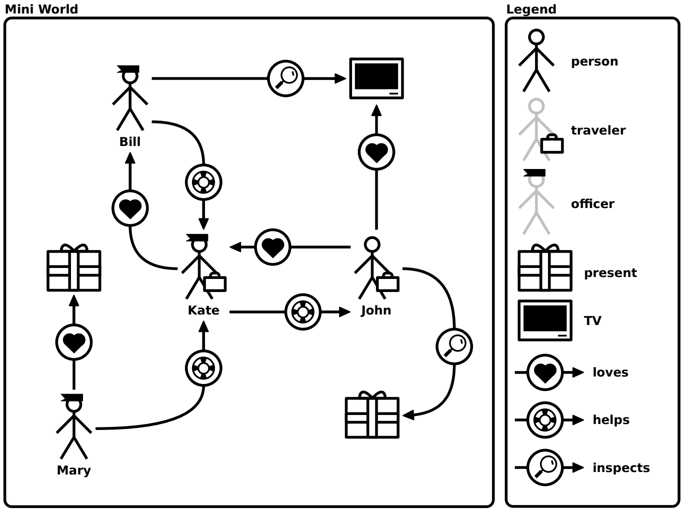


Fig. 1. This is an example of an ontograph. The legend on the right hand side defines the types and relations. The mini world on the left hand side shows the actual individuals, their types, and the relations between them.

express this is to draw a *love*-relation from every individual that is a man to the individual Mary. Thus, every individual and every relation instance has to be represented individually. This has the apparent consequence that the ontograph notation cannot be used to describe situations with an infinite number of individuals and becomes impractical with something around 50 or more individuals and relation instances.

These properties make the ontograph notation a very simple language. They have also the consequence that the ontograph notation is no candidate for becoming a knowledge representation language of its own. A knowledge representation language without support for partial knowledge and generalization would not be very useful.

2.3 Ontograph Experiments

Ontographs are designed to be used in experiments testing the understandability of formal languages. They could, in principle, also be used to test the writability of languages by asking the participants to describe given situations. However, the latter approach has not yet been investigated.

In order to test the understandability of a language, an ontograph and several statements (written in the language to be tested) are shown to the participants of an experiment, who have to decide which of the statements are true and which are false with respect to the situation depicted by the ontograph.

Another important property of ontographs is that they use a graphical notation that is syntactically very different from textual languages like CNLs. This makes it virtually impossible to distinguish true and false statements of a given textual language with respect to a given ontograph just by looking at the syntax. If participants manage to systematically classify the statements correctly as true or false then it can be concluded with a high degree of certainty that the participants understood the statements and the ontograph.

This point gets even clearer by applying a direct connection to model theory [3]. From a model-theoretic point of view, one can say that ontographs are a language to describe first-order models. The statements that are shown to the participants of an experiment would then be very simple first-order theories. From this point of view, the task of the participants is to decide whether certain theories have the shown ontograph as a model or not. We can say that participants understand a statement if they can correctly and systematically classify different ontographs as being a model of the statement or as not being a model thereof. This conclusion can be drawn because the truth of first-order theories can be defined solely on the basis of their models. Thus, being able to identify the ontographs that are models of a theory represented by a statement means that the respective person understands the statement correctly.

Admittedly, this model-theoretic interpretation of the term "understanding" is relatively narrow and ignores important problems like symbol grounding [9]. Such kinds of problems are not covered by the ontograph approach. Nevertheless, the ontograph framework allows us to draw stronger conclusions about the understandability of a language than other existing approaches.

2.4 Limitations and Related Approaches

The introduced ontograph approach, of course, also has its limitations. The most important one is probably the fact that only relatively simple forms of logic can be represented. Basically, ontographs cover first-order logic without functions and with the restriction to unary and binary predicates.

I do not see a simple solution at this point how predicates taking three arguments, for example, could be represented in an intuitive way. It would be even harder to represent more sophisticated forms of logic, e.g. modal or temporal logic. For such cases, it might be necessary to come back to task-based and paraphrase-based approaches to evaluate the understandability of languages. The core of such sophisticated forms of logic, however, could still be tested with the ontograph approach.

There are several existing approaches of using graphical notations to represent logical statements, for example Peirce's *existential graphs* [14] and Sowa's *conceptual graphs* [18]. However, such languages are fundamentally different from ontographs in the sense that they aim at representing general logical statements and in the sense that they are not designed to be intuitively understandable but have to be learned.

The combination of intuitive pictures and statements in natural language can also be found in books for language learners, e.g. "English through pictures"

[15]. As in the ontograph framework, pictures are used as a language that is understood without explanation.

The idea of "textual model checking" presented by Bos [2] is similar to the ontograph approach in some respects. Like in the ontograph approach, there is the task of classifying given statements as true or false with respect to a given situation. In contrast to the approach presented here, the task is to be performed by computer programs and not by humans, and it is about testing these computer programs rather than the language.

3 Experiment Design

The presented ontograph framework has been applied to test whether Attempto Controlled English (ACE) [5], which is a controlled subset of English, is easier to understand than a comparable common formal language. The experiment to be described was performed on 64 participants. Further design decisions are described below in more detail.

3.1 Comparable Language

The most important design decision for the experiment is the choice of the language to which ACE is compared. For this experiment, the Manchester OWL Syntax, a usability-oriented syntax of the ontology language OWL, has been chosen. The inventors of the Manchester OWL Syntax introduce it as follows [11]:

> The syntax, which is known as the Manchester OWL Syntax, was developed in response to a demand from a wide range of users, who do not have a Description Logic background, for a "less logician like" syntax. [...] This means that it is quick and easy to read and write.

As this quotation shows, the Manchester OWL Syntax is designed for good usability and good understandability and thus seems to be an appropriate choice for this experiment. However, the Manchester OWL Syntax requires the statements to be grouped by their main ontological entity (the one in subject position so to speak). This is a reasonable approach for the definition of complete ontologies, but it makes it impossible to state short and independent statements that could be used for a direct comparison to ACE in an experimental setting. For this reason, a modified version of the Manchester OWL Syntax has been defined specifically for this experiment. The resulting language, which I will call "Manchester-like language" or "MLL", uses the same or very similar keywords but allows us to state short and independent statements.

3.2 Learning Time

Obviously, the understanding of a language highly depends on the amount of time spent for learning the language. This means that one has to define a certain time frame when evaluating the understandability of languages. Some languages might be the best choice if there is only little learning time; other languages might be less understandable in this situation but are more suitable in the long run.

So far, little is known about how the understandability of CNLs compares to the understandability of common formal languages. CNLs are designed to be understandable with no learning and the results of the first ontograph experiment [12] show that this is indeed the case. Since other formal languages like the Manchester OWL Syntax are not designed to be understandable with no learning at all, it would not be appropriate to compare ACE to such a language in a zero learning time scenario.

For this reason, I chose a learning time of about 20 minutes. This seems to be a reasonable first step away from the zero learning time scenario. The effect of longer learning times remains open to be studied in the future.

3.3 Ontographs and Statements

Four series of ontographs have been created that cover certain types of statements: The first series contains only individuals and types without relations; the statements of the second series contain relations with different kinds of simple universal quantifications; the third series contains domain, range, and number restrictions; the fourth series, finally, consists basically only of relations.

For each of the four series, three ontographs have been created. For each ontograph, 20 statement pairs have been defined in a way that each pair consists of an ACE statement and a semantically equivalent MLL statement. Some of the statement pairs are true with respect to their ontograph and the others are false.

Table 1 shows examples of statements in their representations in ACE and MLL. It also shows how the statements are divided into four series. All statements together with their ontograph diagrams are available in my doctoral thesis [13] and online[1].

3.4 Participants

Another important design decision is the choice of the participants. Such studies are mostly performed with students because they are flexible and usually close to the research facilities of the universities. In my case, there are even more reasons why students are a good choice. Students are used to think systematically and logically but they are usually not familiar with formal logical notations (unless this lies in their field of study). In this way, they resemble domain experts who have to formalize their knowledge and who should profit from languages like ACE.

The requirements for the participants have been defined as follows: They had to be students or graduates with no higher education in computer science or logic.

[1] `http://attempto.ifi.uzh.ch/site/docs/ontograph/`

Table 1. This table shows examples of statements in their representations in ACE and MLL. These statements are divided into four series.

Series	ACE	MLL
1	Mary is a traveler.	Mary **HasType** traveler
	Bill is not a golfer.	Bill **HasType** **not** golfer
	Mary is an officer or is a golfer.	Mary **HasType** officer **or** golfer
	Sue is an officer and is a traveler.	Sue **HasType** officer **and** traveler
	Every man is a golfer.	man **SubTypeOf** golfer
	No golfer is a woman.	golfer **DisjointWith** woman
	Every woman is an officer and every officer is a woman.	woman **EquivalentTo** officer
	Every traveler who is not a woman is a golfer.	traveler **and** (**not** woman) **SubTypeOf** golfer
	Every man is a golfer or is a traveler.	man **SubTypeOf** golfer **or** traveler
	Nobody who is a man or who is a golfer is an officer and is a traveler.	man **or** golfer **SubTypeOf** **not** (officer **and** traveler)
2	Lisa sees Mary.	Lisa sees Mary
	Mary does not see Tom.	Mary **not** sees Tom
	Tom buys a picture.	Tom **HasType** buys **some** picture
	Mary sees no man.	Mary **HasType** **not** (sees **some** man)
	John buys something that is not a present.	John **HasType** buys **some** (**not** present)
	John sees nothing but men.	John **HasType** sees **only** man
	Every man buys a present.	man **SubTypeOf** buys **some** present
	Everything that buys a present is a man.	buys **some** present **SubTypeOf** man
	Every man buys nothing but presents.	man **SubTypeOf** buys **only** present
	Everything that buys nothing but pictures is a woman.	buys **only** picture **SubTypeOf** woman
3	Everything that inspects something is an officer	inspects **HasDomain** officer
	Everything that is inspected by something is a letter.	inspects **HasRange** letter
	Everything that inspects something is a golfer or is an officer.	inspects **HasDomain** golfer **or** officer
	Everything that is seen by something is an officer or is a picture.	sees **HasRange** officer **or** picture
	Lisa inspects at least 2 letters.	Lisa **HasType** inspects **min** 2 letter
	Lisa helps at most 1 person.	Lisa **HasType** helps **max** 1 person
	Every officer helps at least 2 persons.	officer **SubTypeOf** helps **min** 2 person
	Everything that sees at least 2 pictures is an officer.	sees **min** 2 picture **SubTypeOf** officer
	Every person inspects at most 1 letter.	person **SubTypeOf** inspects **max** 1 letter
	Everything that is an officer or that is a golfer sees at most 1 picture.	officer **or** golfer **SubTypeOf** sees **max** 1 picture
4	If X helps Y then Y helps X.	helps **IsSymmetric**
	If X sees Y then Y does not see X.	sees **IsAsymmetric**
	If X sees somebody who sees Y then X sees Y.	sees **IsTransitive**
	If X admires Y then X sees Y.	admires **SubRelationOf** sees
	If X inspects Y then X helps Y.	inspects **SubRelationOf** helps
	If X helps Y then Y admires X.	helps **SubRelationOf** **inverse** admires
	If X loves Y then X does not admire Y.	loves **DisjointWith** admires
	If X sees Y then Y does not love X.	sees **DisjointWith** **inverse** love
	If X admires Y then X sees Y. If X sees Y then X admires Y.	admires **EquivalentTo** sees
	If X inspects Y then Y sees X. If Y sees X then X inspects Y.	inspects **EquivalentTo** **inverse** sees

Furthermore, at least intermediate level skills in written German and English were required, because the experiment itself was explained and performed in German, and English was needed to understand the ACE sentences.

64 students have been recruited who fulfill these requirements and exhibit a broad variety of fields of study. The students were on average 22 years old and 42% of them were female and 58% were male.

In order to enable a good comparison between the two languages, each participant was tested on ACE and on MLL. However, since participants cannot be expected to concentrate for much longer than one hour, only one of the four ontograph series could be tested per participant.

In order to rule out learning effects, half of the participants received the ACE task first and then the MLL task while the other half received the tasks in the reverse way.

3.5 Procedure

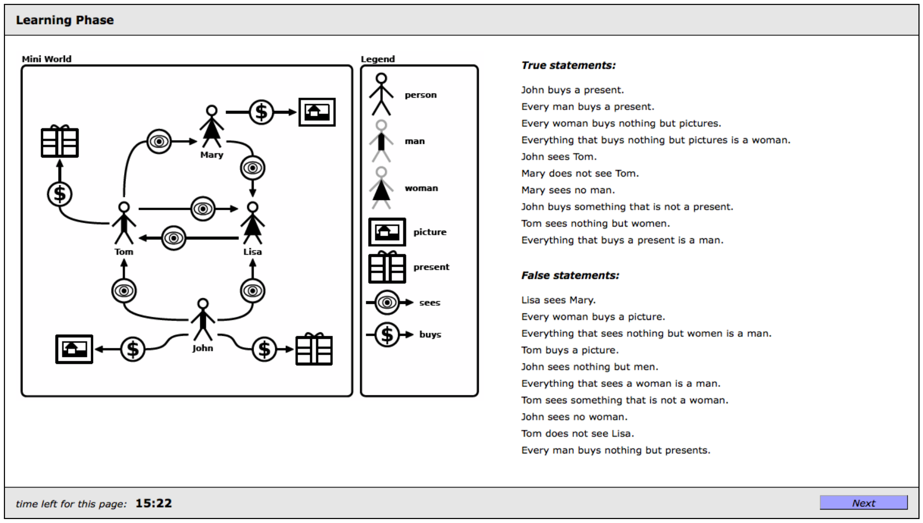
The experiment was conducted in a computer room with a computer for each participant. The main part of the experiment was performed on the computer screen. Additionally, the participants received different printed sheets during the experiment. The overall procedure consisted of six stages:

1. Instructions with Control Questions
2. Learning Phase 1
3. Testing Phase 1
4. Learning Phase 2
5. Testing Phase 2
6. Questionnaire

For the instruction phase, the participants received a printed instruction sheet that explained the experiment procedure, the payout[2], and the ontograph notation. The reverse side of the instruction sheet contained control questions for the participants to answer, which allowed us to check whether the participants understood the instructions correctly. The participants had to return the filled-out instruction sheets to the experimenter, who checked whether all questions were answered correctly. In the case of false answers, the experimenter explained the respective issue to the participant.

For the first learning phase, the participants received a language description sheet of the first language (either ACE or MLL). This language description sheet only explained the subset of the language that is used for the respective series. For this reason, each series had its own instruction sheets for both languages. During the learning phase, the participants had to read the language description sheet. Furthermore, an ontograph (the same as on the instruction sheet) was shown on the screen together with 10 true statements marked as “true” and 10 false statements marked as “false” in the respective language. Figure 2 shows a screenshot of the experiment screen during the learning phase.

[2] Apart from a fixed fee, the participants received an additional small amount of money for each correctly classified statement and half of it for “don’t know” answers.



True statements:

John buys a present.
Every man buys a present.
Every woman buys nothing but pictures.
Everything that buys nothing but pictures is a woman.
John sees Tom.
Mary does not see Tom.
Mary sees no man.
John buys something that is not a present.
Tom sees nothing but women.
Everything that buys a present is a man.

False statements:

Lisa sees Mary.
Every woman buys a picture.
Everything that sees nothing but women is a man.
Tom buys a picture.
John sees nothing but men.
Everything that sees a woman is a man.
Tom sees something that is not a woman.
John sees no woman.
Tom does not see Lisa.
Every man buys nothing but presents.

Fig. 2. This is the screen shown during the learning phase of the experiment.

During the testing phase, a different ontograph was shown on the screen. Furthermore, 10 statements in the respective language were shown on the screen together with radio buttons that allowed the participants to choose between "true", "false" and "don't know". Figure 3 shows how the experiment screen of the testing phase looked like. During the testing phase, the participants could keep the language description sheet that they got for the learning phase. Thus, they did not need to know the language description by heart but they could read parts of it again during the testing phase if necessary.

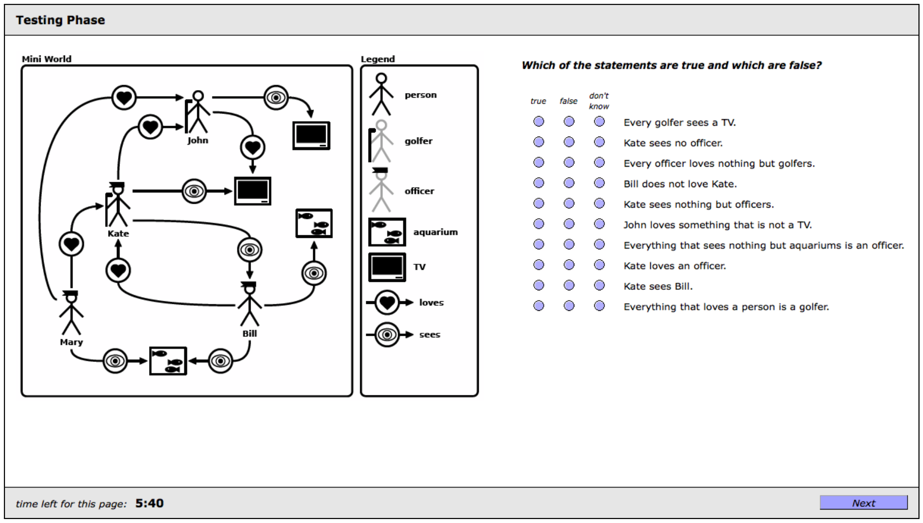
For the steps 4 and 5, the procedure of the steps 2 and 3 was repeated for the second language (i.e. ACE if the first language was MLL and vice versa) with the same ontograph for the learning phase but a new one for the testing.

Finally, the participants received a questionnaire form inquiring about their background and their experiences during the experiment. The experiment was finished when the participants turned in the completed questionnaire form.

The learning phases had a time limit of 16 minutes each, and the time limit for the testing phases was 6 minutes. The participants were forced to proceed when the time limit ran out but they could proceed earlier. In this way, it can not only be investigated how understandable the languages are but also how much time the participants needed to learn them.

3.6 Language Description Sheets

The proper design of the language description sheets is crucial for this experiment. If the participants perform better in one language than in the other, it might be that the respective language was merely described better than the



Which of the statements are true and which are false?

true	*false*	*don't know*	
○	○	○	Every golfer sees a TV.
○	○	○	Kate sees no officer.
○	○	○	Every officer loves nothing but golfers.
○	○	○	Bill does not love Kate.
○	○	○	Kate sees nothing but officers.
○	○	○	John loves something that is not a TV.
○	○	○	Everything that sees nothing but aquariums is an officer.
○	○	○	Kate loves an officer.
○	○	○	Kate sees Bill.
○	○	○	Everything that loves a person is a golfer.

Fig. 3. This is the screen shown during the testing phase of the experiment.

other. Thus, the language description sheets have to be written very carefully to be sure that they are not misunderstood and are optimal for learning the respective language under the given time restrictions. Especially the description sheets for MLL are critical. In contrast to ACE, MLL is not designed to be understood without training. For this reason, a special effort has been made to ensure the quality of the MLL description sheets. This quality assurance effort involved several steps.

First of all, the four series were designed in a way that at most seven MLL keywords are used per series. Since each series has its own language description sheets, not more than seven keywords have to be described by the same sheet. This should make it easier to understand the needed subset of the language.

In a second step, the different MLL description sheets were given to three persons, who comply with the restrictions of the experiment but who did not participate in it. These three persons read the sheets and gave me feedback about what they did not understand and what could be improved.

As a third step, I performed a test run with 9 participants to receive final feedback about the understandability and usefulness of the language description sheets. After the test run, the participants received the sheets again and they were told to highlight everything that was difficult to understand. Only very few things were highlighted (altogether two highlightings in the MLL description, one in the ACE description, and none in the general instructions) and according to this I made a couple of small last changes for the main experiment.

Altogether, the language description sheets were compiled very carefully and it is very unlikely that a different description of MLL would radically increase its understandability.

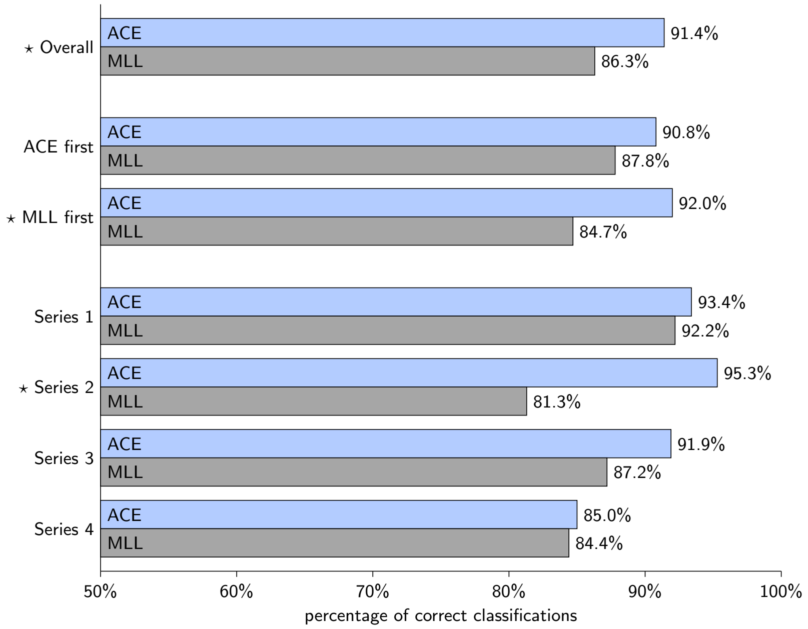


Fig. 4. This chart shows the percentage of correct classifications. The base line is 50% that can be achieved by mere guessing. "Don't know"-classifications and cases where the time limit ran out count as 0.5 correct classifications. Significant differences are marked by "⋆" (see Table 2 for details).

4 Results

The results of the experiment allow for comparing ACE and MLL on the basis of the classification results, the time required by the participants, and the answers they gave in the questionnaire. It also has to be evaluated whether the ontograph framework altogether worked out or not.

4.1 General Classification Scores

Figure 4 shows the average percentages of correct classifications per testing phase. "Don't know" answers and the cases where the time limit ran out are counted as 0.5 correct classifications. 50% is the baseline because an average of five correct classifications out of ten can be achieved by mere guessing (or, for that matter, by choosing always "don't know" or by letting the time limit run out).

91.4% of the statements were classified correctly in the case of ACE and 86.3% in the case of MLL. Thus, out of the ten statements of a testing phase, ACE was on average 0.5 points better than MLL. This is a considerable and statistically significant difference (the details of the used statistical test to compare the two samples are explained later on). One has to consider that these values are already close to the ceiling in the form of the perfect score of 10, which might have reduced the actual effect.

The results of the participants who received ACE first and then MLL can now be compared with the ones who received MLL first. As expected, both languages were understood better when they were the second language. This can be explained by the fact that the participants were more familiar with the procedure, the task, and the ontograph notation. However, even in the case when ACE was the first language and MLL the second one, ACE was understood better (but in this case not within statistical significance).

Looking at the results from the perspective of the different series, one can see that ACE was better in all cases but only the series 2 and 3 exhibit a clear dominance of ACE (and this dominance is significant only for series 2). According to these results, one could say that languages like MLL are equally easy to understand for very simple statements as the ones in series 1 and for statements about relations as they appear in series 4. In the case of series 1, the reason might be that these statements are so simple that they can be understood even in a rather complicated language. In the case of series 4, the reason is probably that Description Logic based languages like MLL can express these statements without variables whereas ACE needs variables, which are somehow borderline cases in terms of naturalness.

In summary, the results show that — while both languages are understood reasonably well — ACE is easier to understand than MLL.

4.2 Time

As a next step, we can look at the time values. For simplicity reasons and since the learning process was presumably not restricted to the learning phase but continued during the testing phase, the time needed for both phases will together be called the *learning time*.

Figure 5 shows the learning times of the participants. They could spend at most 22 minutes: 16 minutes for the learning phase and 6 minutes for the testing phase. The results show that the participants needed much less time for ACE than for MLL. In the case of ACE less than 14 minutes were needed, whereas in the case of MLL the participants needed more than 18 minutes. Thus, MLL required 29% more time to be learned, compared to ACE.

Note that these results can be evaluated only together with the results described above concerning the classification scores. The learning time can only be evaluated together with the degree of understanding it entails. The smaller amount of learning time for ACE can be explained simply by the fact that the language description sheets for ACE contained less text than the ones for MLL. But together with the results described above that show that ACE was understood better and the fact that the language description sheets have been written very carefully, it can be concluded that ACE required less learning time while leading to a higher degree of understanding.

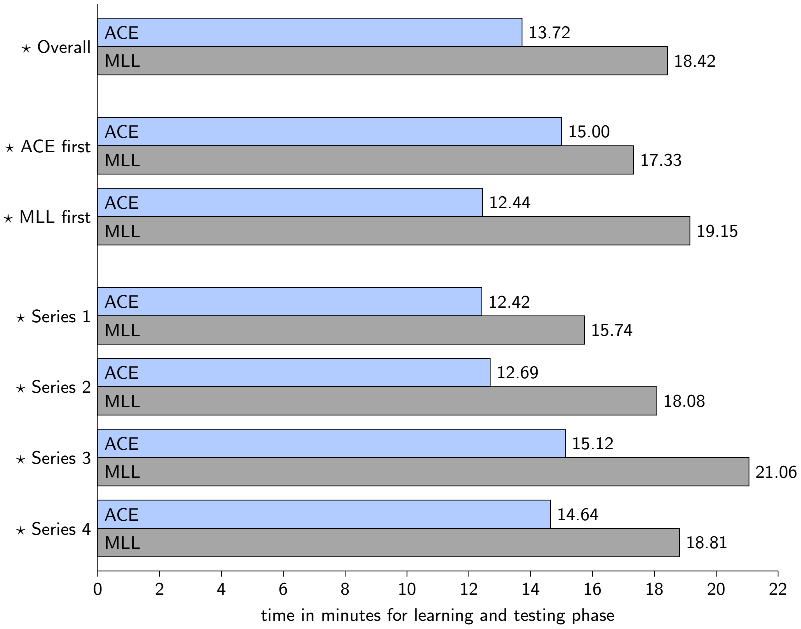


Fig. 5. This chart shows the average time needed for learning and testing phase. Significant differences are marked by "⋆" (see Table 2 for details).

Again, we can split the results according to the participants who received ACE first and those who received MLL first. The results show the expected effect: ACE and MLL required less time as second language. However, ACE required less time than MLL no matter if it was the first language or the second. Thus, even in the cases where ACE was the first language and the participants had no previous experience with the procedure and MLL was the second language and the participants could use the experiences they made before, even in such cases ACE required less time.

Looking at the different series, we can see that this effect spreads over all four series. MLL required on average between 3 and 6 minutes more learning time than ACE.

The better time values of ACE compared to MLL are statistically significant for the whole sample and also for all presented subsamples.

4.3 Perceived Understandability

As a third dimension, we can look at the "perceived understandability", i.e. how the participants perceived the understandability of the languages. The questionnaire that the participants filled out after the experiment contained two questions that asked the participants how understandable they found ACE and

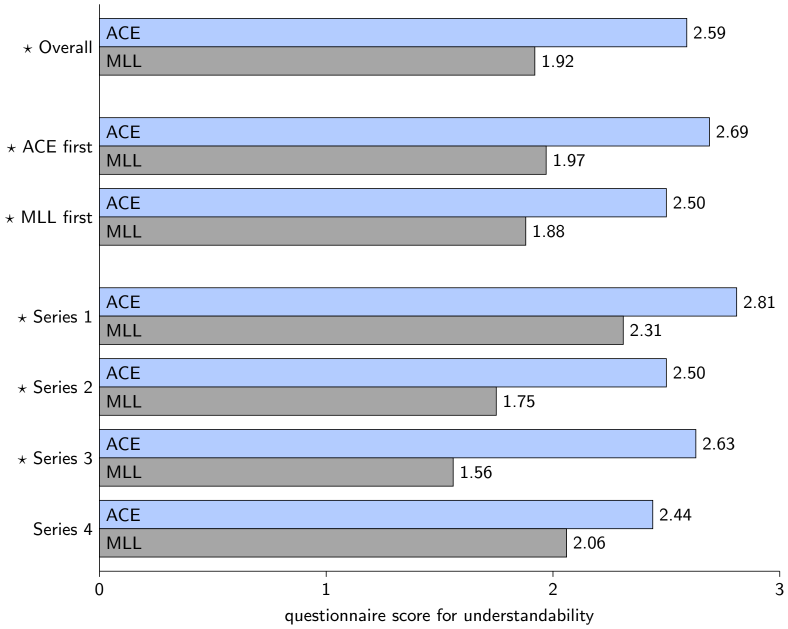


Fig. 6. This chart shows the average scores for perceived understandability derived from the questionnaire. 0 means "very hard to understand", 1 means "hard to understand", 2 means "easy to understand", and 3 means "very easy to understand". Significant differences are marked by "⋆" (see Table 2 for details).

MLL, respectively. They could choose from four options: "very hard to understand" (value 0), "hard to understand" (1), "easy to understand" (2) and "very easy to understand" (3). The perceived understandability does not necessarily have to coincide with the actual understandability and can be a very valuable measure for the acceptance of a language and the confidence of its users.

Figure 6 shows the scores for perceived understandability derived from the questionnaire. Overall, ACE got much better scores than MLL. MLL was close but below "easy to understand" scoring 1.92, whereas ACE was closer to "very easy to understand" than to "easy to understand" scoring 2.59.

By dividing the results into those who received ACE first and those who received MLL first, we see that both languages scored better when ACE was the first language. I do not have a convincing explanation for this and it might just be a statistical artifact.

Looking at the perceived understandability scores from the perspective of the different series, we can see that ACE clearly received better scores in all four series. It is interesting that this also holds for the series 1 and 4 where ACE was not much better than MLL in terms of actual understanding, as shown before. Thus, even though the actual understanding of the statements of these

Table 2. This table shows the *p*-values of Wilcoxon signed-rank tests. The null hypothesis is that the given values are not different for ACE and for MLL. This null hypothesis can be rejected in 16 of the 21 cases on a 95% confidence level and these cases are marked by "$\star$".

	classification		questionnaire
	score	time	score
complete sample	0.003421 $\star$	1.493×10^{-10} $\star$	3.240×10^{-7} $\star$
ACE first	0.2140	0.006640 $\star$	7.343×10^{-5} $\star$
MLL first	0.005893 $\star$	3.260×10^{-9} $\star$	0.001850 $\star$
Series 1	0.5859	0.01309 $\star$	0.02148 $\star$
Series 2	0.003052 $\star$	0.002624 $\star$	0.02197 $\star$
Series 3	0.1250	9.155×10^{-5} $\star$	0.0004883 $\star$
Series 4	0.6335	0.002686 $\star$	0.1855

series does not show a clear difference, the acceptance and confidence of the participants seems to be higher in the case of ACE.

4.4 Significance

The charts with the experiment results indicate in which cases the difference between ACE and MLL is statistically significant. This was done by using the Wilcoxon signed-rank test [19], which is a non-parametric statistical method for testing the difference between measurements of a paired sample. In contrast to Student's *t*-test, this test does not rely on the assumption that the statistical population corresponds to a standard normal distribution. This relieves us from investigating whether standard normal distribution can be assumed for the given situation.

Table 2 shows the obtained *p*-values for the three dimensions of our comparison (i.e. classification score, time, and questionnaire score). For the complete sample, the values are well within the 95% confidence level for all three dimensions. They are even within the 99% level.

4.5 Framework Evaluation

Finally, it can be evaluated whether the ontograph framework altogether worked out or not.

Figure 7 shows the results of two questions of the questionnaire asking the participants about how understandable they found the ontograph notation and the overall instructions. Both values are between "easy to understand" and "very easy to understand". This shows that the ontographs were well accepted by the participants and that it is possible to explain the procedure of such experiments in an understandable way.

Furthermore, the results of the experiment show that the ontographs were indeed very well understood by the participants. For both languages, the overall

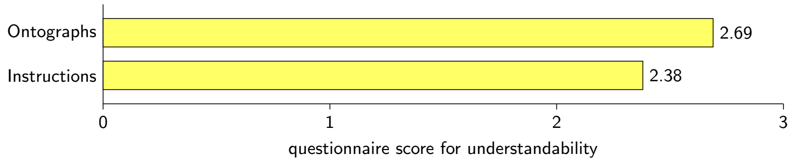


Fig. 7. This chart shows the average understandability scores for the ontograph notation and the instructions, derived from the questionnaire. 0 means "very hard to understand", 1 means "hard to understand", 2 means "easy to understand", and 3 means "very easy to understand".

percentage of correct classifications exceeded 85%. Such good results are only possible if the ontographs and the instructions are understood.

5 Conclusions

The results of the two experiments show that the ontograph framework worked out very well and is suitable for testing the understandability of languages. I could show that ACE is understood significantly better than the comparable language MLL. Furthermore, ACE required much less time to be learned and was perceived as more understandable by the participants.

MLL is directly derived from the Manchester OWL Syntax in a way that leaves its properties concerning understandability intact. For this reason, the conclusions of the experiment can be directly applied to the Manchester OWL Syntax, which is the state of the art approach on how to represent ontological knowledge in a user-friendly manner. Thus, it could be shown that CNLs like ACE can do better in terms of understandability than the current state of the art.

Altogether, the results suggest that CNLs should be used instead of languages like the Manchester OWL Syntax in situations where people have to deal with knowledge representations after little or no training.

References

1. Abraham Bernstein and Esther Kaufmann. GINO — a guided input natural language ontology editor. In *The Semantic Web — ISWC 2006, Proceedings of the 5th International Semantic Web Conference*, number 4273 in Lecture Notes in Computer Science, pages 144–157. Springer, November 2006.
2. Johan Bos. Let's not Argue about Semantics. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 2835–2840. European Language Resources Association (ELRA), 2008.
3. Chen Chung Chang and H. Jerome Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, Amsterdam, 1973.

4. Steve Chervak, Colin G. Drury, and James P. Ouellette. Field evaluation of simplified english for aircraft workcards. In *Proceedings of the 10th FAA/AAM Meeting on Human Factors in Aviation Maintenance and Inspection*, 1996.
5. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for knowledge representation. In *Reasoning Web — 4th International Summer School 2008*, number 5224 in Lecture Notes in Computer Science, pages 104–124. Springer, 2008.
6. Adam Funk, Brian Davis, Valentin Tablan, Kalina Bontcheva, and Hamish Cunningham. Controlled language IE components version 2. SEKT Project Deliverable D2.2.2, University of Sheffield, UK, 2007.
7. Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. CLOnE: Controlled language for ontology editing. In *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC 2007 + ASWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*. Springer, 2007.
8. Catalina Hallett, Donia Scott, and Richard Power. Composing questions through conceptual authoring. *Computational Linguistics*, 33(1):105–133, 2007.
9. Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1–3):335–346, June 1990.
10. Glen Hart, Martina Johnson, and Catherine Dolbear. Rabbit: Developing a controlled natural language for authoring ontologies. In *Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*, number 5021 in Lecture Notes in Computer Science, pages 348–360. Springer, 2008.
11. Matthew Horridge, Nick Drummond, John Goodwin, Alan L. Rector, Robert Stevens, and Hai Wang. The Manchester OWL syntax. In *Proceedings of the OWLED '06 Workshop on OWL: Experiences and Directions*, volume 216 of *CEUR Workshop Proceedings*. CEUR-WS, 2006.
12. Tobias Kuhn. How to evaluate controlled natural languages. In *Pre-Proceedings of the Workshop on Controlled Natural Language (CNL 2009)*, volume 448 of *CEUR Workshop Proceedings*. CEUR-WS, April 2009.
13. Tobias Kuhn. *Controlled English for Knowledge Representation*. Doctoral thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, Switzerland, to appear.
14. Charles S. Peirce. Existential graphs. In *Collected Papers of Charles Sanders Peirce, Volume 4: The Simplest Mathematics*. Harvard University Press, 1932.
15. I. A. Richards and Christine M. Gibson. *English through Pictures*. Washington Square Press, 1945.
16. Rolf Schwitter, Kaarel Kaljurand, Anne Cregan, Catherine Dolbear, and Glen Hart. A comparison of three controlled natural languages for OWL 1.1. In *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions*, volume 496 of *CEUR Workshop Proceedings*. CEUR-WS, 2008.
17. Rolf Schwitter and Marc Tilbrook. Controlled Natural Language meets the Semantic Web. In *Proceedings of the Australasian Language Technology Workshop 2004*, volume 2 of *ALTA Electronic Proceedings*, pages 55–62. Australasian Language Technology Association, 2004.
18. John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole, 2000.
19. Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, December 1945.