I2-D15

More on Reasoning and Semantic Wikis

Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Project number:	IST-2004-506779
Project instrument:	EU FP6 Network of Excellence (NoE)
Project thematic priority:	Priority 2: Information Society Technologies (IST)
Document type:	D (deliverable)
Nature of document:	R/P (report and prototype)
Dissemination level:	PU (public)
Document number:	IST506779/Zurich/I2D15/D/PU
Responsible editor:	Norbert E. Fuchs
Reviewers:	Juri Luca De Coi (I2)
Contributing participants:	University of Zurich
Contributing workpackages:	I2
Contractual date of delivery:	15 March 2008
Actual date of delivery:	22 May 2008

Abstract

We report the progress of our work on the Attempto Reasoner RACE and the semantic wiki AceWiki, both of which were already described in detail in deliverable I2-D13. Also, we summarise updates on the ACE tools APE and ACE View, and list cooperations and visits.

Keyword List

Attempto Controlled English, ACE, Attempto reasoner, RACE, semantic wiki, AceWiki

Project co-founded by the European Commission and the Swiss State Secretariat for Education and Research within the Sixth Framework Programme

More on Reasoning and Semantic Wikis

Norbert E. Fuchs, Kaarel Kaljurand, Tobias Kuhn
Department of Informatics
&
Institute of Computational Linguistics
University of Zurich
Email: {fuchs, kalju, tkuhn}@ifi.uzh.ch

22 May 2008

Abstract

We report the progress of our work on the Attempto Reasoner RACE and the semantic wiki AceWiki, both of which were already described in detail in deliverable I2-D13. Also, we summarise updates on the ACE tools APE and ACE View, and list cooperations and visits.

Keyword List

Attempto Controlled English, ACE, Attempto reasoner, RACE, semantic wiki, AceWiki

Contents

1. Introduction

Research on Attempto Controlled English and its tools has progressed as outlined during the final REWERSE Review Meeting in January 2008.

In this deliverable we present progress for the Attempto Reasoner RACE and the semantic wiki AceWiki, summarise updates on the ACE tools APE and ACE View, and list cooperations and visits.

2. Reasoning in ACE

2.1. Introduction

The Attempto Reasoner RACE [Fuchs & Schwertel 2003] supports automatic reasoning in ACE. RACE proves that theorems expressed in ACE are the logical consequence of axioms expressed in ACE, and gives a justification for the proof in ACE. If there is more than one proof, then RACE will find all of them. Variations of the basic proof procedure permit query answering and consistency checking. RACE was already described in some detail in REWERSE deliverable I2-D13 [Fuchs et al. 2007]. We assume that readers are familiar with this description, and will here only report the progress of work since then.

2.2. Controlling Deductions

2.2.1. Deductions form Collective Plurals

RACE uses several parameters to control, i.e. to enable or disable, deductions from collective plurals that can occur as subjects or objects of intransitive, transitive, and ditransitive verbs.

Here is an example. To enable the derivation of

```
A man waits.
```

from

```
Two men wait.
```

or from

```
A man and a woman wait.
```

users have to set the parameter "enable deductions from subject of intransitive verb" with the abbreviation "si".

However as replacing the verb `wait` by the verb `meet` shows, the deduction is not felicitous in all cases, and thus should not be generally enabled.

To disable the derivation of

```
A man lifts a table.
```

from

```
Two men lift a table.
```

users should not set the parameter "enable deductions from subject of transitive verb" with the abbreviation "st".

Again as replacing the verb `lift` by the verb `see` shows, there are cases when the deduction might be desirable, and thus should be enabled.

Altogether the following parameters for deductions from collective plurals are defined:

- enable deductions from collective plural subjects of intransitive verbs (si)
- enable deductions from collective plural subjects of transitive verbs (st)
- enable deductions from collective plural objects of transitive verbs (ot)
- enable deductions from collective plural subjects of ditransitive verbs (sdt)
- enable deductions from direct collective plural objects of ditransitive verbs (dodt)
- enable deductions from indirect collective plural objects of ditransitive verbs (iodt)
- enable deductions from 'there are' constructs with collective plurals (sti)

Users need to decide for each deduction which of these parameters have to be set. There are default settings.

2.2.2. Check for Additional Deductions

If there are any proofs, RACE returns all of them from the first level of the proof tree where proofs are found. The parameter "check for additional proofs" with the abbreviation "ap" allows you to search for additional proofs that could occur deeper in the proof tree.

- check for additional deductions (ap)

Here is an example showing the action of the parameter "ap". The results use the format of the RACE web-interface (see below). Given the ACE axioms

```
There is a red apple.
There is a green apple.
There is an apple-tree.
If there is an apple-tree then there are some apples.
```

and the question

```
Is there an apple?
```

then without setting the parameter "ap" RACE will provide two solutions generated directly from the first two axioms,

```
The following minimal subsets of the axioms answer the query:
Subset 1
     There is a green apple.
Subset 2
     There is a red apple.
```

With the parameter "ap" set, RACE provides one further solution that is generated by modus ponens from the third and fourth axioms.

```
The following minimal subsets of the axioms answer the query:
Subset 1
     There is a green apple.
Subset 2
     There is a red apple.
Subset 3
     There is an apple-tree.
     If there is an apple-tree then there are some apples.
```

2.2.3. Display Auxiliary Axioms

RACE uses internal auxiliary axioms, for instance to relate the plural and the singular of noun phrases, or to operate with natural numbers. A parameter allows you to output the auxiliary axioms that were used in a deduction.

- display auxiliary axioms (aux)

Here is an example. With the parameter "aux" set we can derive from the axiom

```
There are more than 5 green apples and at least 3 red apples.
```

the theorem

```
There is an apple.
```

with the output

```
The following minimal subsets of the axioms entail the theorems:
Subset 1
     There are more than 5 green apples and at least 3 red apples.
     Prolog Axiom cd2: at least M objects |- M, M-1, ..., 1 objects
Subset 2
     There are more than 5 green apples and at least 3 red apples.
```

```
        Prolog Axiom cd3: more than M objects |- M, M-1, ..., 1 objects
```

With the parameter "aux" not set we get the results

```
The following minimal subsets of the axioms entail the theorems:
Subset 1
        There are more than 5 green apples and at least 3 red apples.
```

Notice that the two previous results – that only differed in the auxiliary axioms but not in the ACE axioms being used for the proof – are now compacted to one result.

2.3. Auxiliary Axioms

In I2-D13 we wrote

> RACE is supported by auxiliary axioms expressed in the language of first-order logic or in Prolog. Auxiliary axioms implement domain-independent linguistic and mathematical knowledge that cannot be expressed in ACE since it depends on the DRS representations of ACE texts. Examples are the relation between plurals and singulars and a theory of natural numbers. Auxiliary axioms can also act as meaning postulates for ACE constructs that are under-represented in the DRS, for example generalised quantifiers.

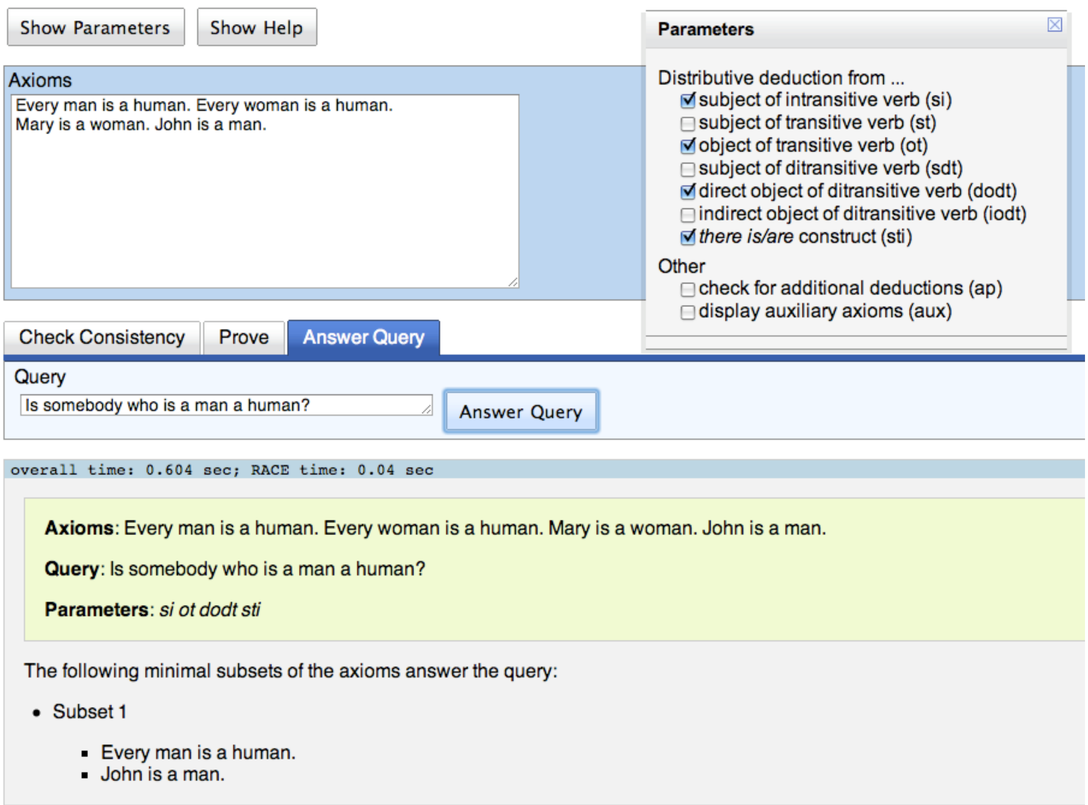
As it turned out, all auxiliary axioms devised so far can be expressed in Prolog. This results in increased efficiency of RACE. While auxiliary axioms expressed in first-order logic are translated into clauses that permanently participate in RACE's forward chaining – whether or not the auxiliary axioms are actually needed – Prolog axioms are only called when needed.

2.4. RACE Web-Service and Web-Interface

RACE can be called via a SOAP web-service that is described in detail on our web-site (http://attempto.ifi.uzh.ch/site/docs/race_webservice.html).

RACE can also be accessed via a web-client (http://attempto.ifi.uzh.ch/race) that is described in detail in the document (http://attempto.ifi.uzh.ch/site/docs/race_webclient_help.html).

Here is a typical screen-shot of the RACE web-client.



2.5. Why Not?

Succeeding proofs of RACE show the reason why they succeed by listing, for instance, the axioms needed to answer a question. In brief: RACE answers the question "why?".

For failing proofs, RACE also answers the question "why not?" by listing those parts of the theorem or query that could not be proved. Here is an example.

```
overall time: 0.603 sec; RACE time: 0.01 sec
```

Axioms: Every man is a human. Every woman is a human. Mary is a woman. John is a man.

Theorems: Does Mary own a cat?

Parameters: *si ot dodt sti*

Theorems do not follow from axioms.

The following parts of the theorems/query could not be proved:

- transitive verb: own
- common noun: cat

As we see, the transitive verb "own" and the common noun "cat" of the query "Does Mary own a cat?" could not be proved because no relevant information was found in the axioms.

Currently, the "why not?" information is created by checking each logical atom of the clauses derived from the ACE theorem, respectively ACE query, against all atoms that were proved during the proof. Using atoms instead of the ACE text itself, means that the "why not?" output can in some cases not readily be associated with the ACE text of the theorem, respectively query. Here is an example that shows this discrepancy.

```
overall time: 0.704 sec; RACE time: 0.01 sec
```

Axioms: Every man is a human. Every woman is a human. Mary is a woman. John is a man.

Theorems: John and Mary see two men.

Parameters: *si ot dodt sti*

Theorems do not follow from axioms.

The following parts of the theorems/query could not be proved:

- common noun: (equal 2) na
- has_part(_G22241, sk1)
- transitive verb: see
- common noun: (equal 2) man
- has_part(_G22241, sk2)

The transitive verb "see", the common noun "(equal 2) man" – derived from "two men" – and the atoms "(equal2) na", "has_part(_G22241, sk1)" and "has_part(_G22241, sk2)" – derived from the conjunctive plural "John and Mary" – could not be proved.

This last output may confound users since its relation to the text of the theorem "John and Mary see two men." is not obvious.

We are working on a solution for "why not?" that will report the concrete ACE words or ACE phrases of the theorem, respectively query.

3. AceWiki

3.1. Introduction

Since ontologies are often defined within communities, semantic wikis could be used for their collaborative creation and management. Unfortunately, most of the existing semantic wikis do not support expressive ontology languages in a general way. They do not allow the users to add complex axioms like "every landlocked country borders no sea". Furthermore, the existing semantic wikis are often hard to understand for people who are not familiar with the technical terms of logic and ontologies.

AceWiki[1] tries to solve both problems by using controlled natural language. Ordinary people who have no background in logic should be able to understand, modify, and extend the formal content of a wiki.

Many existing semantic wikis are classical wikis enriched with semantic annotations. The goal is not to manage stand-alone ontologies, but rather to give some kind of formal backbone to the wiki articles. We follow a different approach – similar e.g. to the *myOntology* project [Siorpaes & Hepp 2007] – by providing a wiki that is dedicated to building and maintaining ontologies. In contrast to myOntology, we do not restrict ourselves to lightweight (i.e. relatively inexpressive) ontologies. The use of controlled natural language allows us to express also complex axioms in a natural way. The picture below shows a screenshot of the AceWiki interface.

In our usage scenario, a community of domain experts uses AceWiki to create and maintain a formal knowledge base in a collaborative manner. There are two exemplary wiki instances – one about geography and the other about protein interactions – that demonstrate how AceWiki could be used to represent knowledge of such communities.

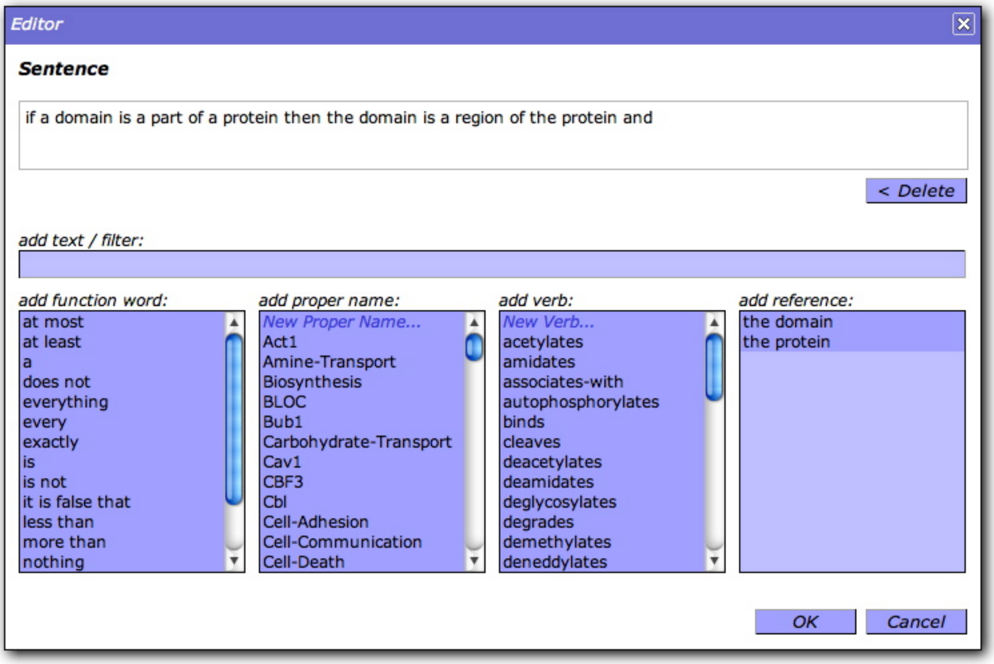
3.2. Design

The goal of AceWiki is to show that semantic wikis can be more natural and at the same time more expressive than existing semantic wikis.

Naturalness is achieved by representing the formal statements in ACE. Since ACE is a subset of natural English, every English speaker can immediately read and understand the content of the wiki. In order to enable easy creation of ACE sentences, AceWiki provides a predictive editor that shows step-by-step the words that are syntactically possible at a given

[1] See [Kuhn 2007], [Kuhn 2008], and http://attempto.ifi.uzh.ch/acewiki

position in the sentence. The picture below shows a screenshot of the predictive editor of AceWiki. Furthermore, the AceWiki interface does not use technical terms like “ontological element”, “property”, or “subclass” but uses instead terms like “word”, “transitive verb”, or “hierarchy” which should be much more familiar to people with no background in logic.



AceWiki makes use of the high expressivity of ACE that goes beyond OWL and SWRL. We do not like the idea of cutting down the expressivity just for the sake of reasoning performance. Even if some statements become so complex that it is almost impossible to do reasoning with them, it is better to have them formalized than just left out. We do not lose anything, since we are free to ignore those complex statements for certain reasoning tasks.

3.3. Reasoning in AceWiki

We have started to integrate the OWL reasoner Pellet[2] into AceWiki. Since ACE sentences can be beyond the expressivity of OWL, the reasoner cannot consider all sentences. In order to make this clear to the users, each sentence is tagged as blue (inside of OWL) or red (outside of OWL):

- ▷ Every protein interacts-with a protein.
- ▷ No protein interacts-with every protein.
- ▷ Every protein-complex contains at least 2 proteins.

In this way, it is easy to explain to the users that only the blue statements are considered when the reasoner is used. We plan to provide an interface that allows skilled users to export the formal content of the wiki and to use it within an external reasoner or rule-engine. Thus, even though the red statements cannot be interpreted by the built-in reasoner they can still be useful.

Consistency checking plays a crucial role because any other reasoning task requires a consistent ontology in order to return useful results. Most other semantic wikis do not have this problem since their languages are simply not expressive enough to ever run into inconsistency.

In order to ensure that the ontology is always consistent, AceWiki checks every new sentence – immediately after its creation – whether it is consistent with the current ontology. Otherwise, the sentence is not included in the ontology:

[2] http://pellet.owldl.com/

▷ Every city is a part of exactly 1 country.
▷ Zurich is a city that is a part of Switzerland.
▷ Zurich is a part of Italy.

After the user created the last sentence of this example, AceWiki detected that it contradicts the current ontology. The sentence is included in the wiki article but the red font indicates that it is not included in the ontology. The user can remove this sentence again, or keep it and try to reassert it later when the rest of the ontology has changed.

For this approach, it is very important to perform incremental reasoning which Pellet supports only partially at the moment. For that reason, AceWiki does not scale very well. We expect that future reasoners will be able to run much faster in such incremental scenarios.

Not only asserted but also inferred knowledge can be represented in ACE. At the moment, AceWiki can show inferred class hierarchies and class memberships. Furthermore, AceWiki supports inline-queries, i.e. questions within wiki articles. Questions are formulated in ACE and evaluated by the reasoner:

▷ What borders Spain?
- Andorra
- Atlantic-Ocean
- France
- Mediterranean-Sea
- Portugal

Thus, ACE can be used not only as an ontology- and rule-language, but also as a query-language.

3.4. Evaluation

In order to evaluate AceWiki, we set up a small-scale end-user experiment. The hypothesis to be tested was whether average people (i.e. people who are not familiar with ontologies and logic) are able to learn how to deal with AceWiki within a short amount of time and without the help of an expert. Note that the experiment was performed on an earlier version of AceWiki.

3.4.1. Experiment Design

The experiment was performed through the internet and it had a very simple design. The prerequisites for participation were only basic English skills and access to a computer with a broadband internet connection. We recruited 20 participants.

The participants received an instruction sheet which they read before they started with the experiment. These instructions explained the procedure and the task, but they did not explain how to interact with the AceWiki interface.

After reading the instructions, the participants were ready to start with the experiment. In the end, they filled out a questionnaire which asked for their background and their experiences with AceWiki.

The task for the participants was to visit AceWiki and to add knowledge to it. They were free to choose what kind of knowledge to add, as long as they followed three restrictions:

- The participants should add only knowledge that is true or at least true in most cases.
- The knowledge should be general, i.e. verifiable by others.
- The participants were allowed and encouraged to change or even delete the contributions of other participants if they found them violating one of the first two restrictions.

Furthermore, the participants were encouraged to add a couple of complex sentences starting with “every”, “no”, or “if”.

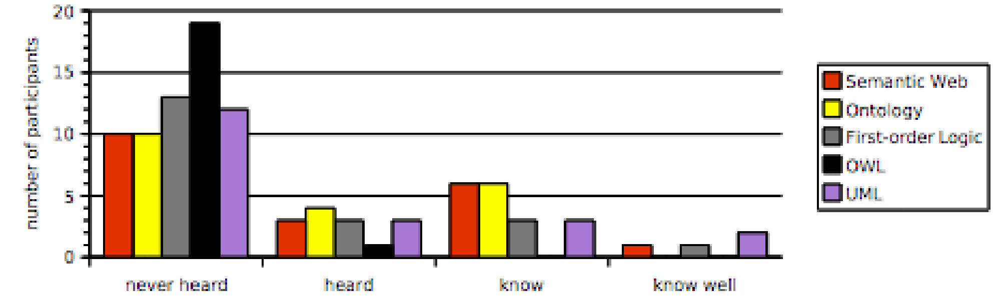
Altogether, each participant should spend between 30 minutes and two hours (possibly split into several intervals) within a time-frame of 14 days.

3.4.2. Results

The basis for the evaluation of the experiment were the questionnaire and the log files from the server.

Most of the 20 participants were students or graduates. Two participants had a computer science background, but they were not experts in the fields of Semantic Web or logic. The table below shows the exact distribution.

participants in total	20	*(100%)*
students in computer science	1	*(5%)*
graduates in computer science	1	*(5%)*
other students	8	*(40%)*
other graduates	8	*(40%)*
no academic background	2	*(10%)*



In the questionnaire, the participants were asked how familiar they are with different technical terms. The picture above shows the result. The chart shows the familiarity of the participants with the terms *Semantic Web*, *ontology*, *first-order logic*, *OWL*, and *UML*. This data was retrieved from five questions of the questionnaire. For each of the terms the question was "How familiar are you with this term?" and there were four choices: "I have never heard it", "I have heard it but I don't really know what it means", "I know more or less its meaning", "I know this term (very) well". The term *OWL* was almost completely unknown. The majority of the participants have never heard the terms *first-order logic* or *UML*. Only in the case of the terms *Semantic Web* and *ontology*, we have a substantial minority knowing those terms. The results show that the participants had no considerable background in Semantic Web technologies or similar fields.

The picture above shows two examples of AceWiki articles how they came out of the experiment. Those screenshots intuitively show that the participants understood the ideas of AceWiki.

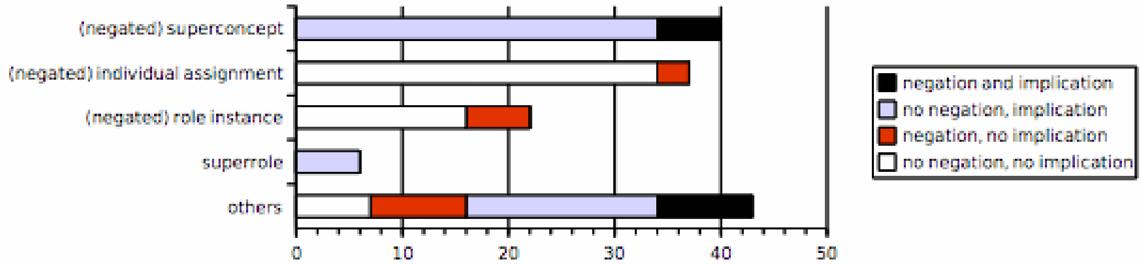
In order to measure how well the participants managed to work with AceWiki, we evaluated the sentences that they created or modified. For each of these sentences, we checked manually whether they represent a correct and sensible fact of the real world under the ACE semantics. The results are shown in the table below.

	total	average	median	min	max
S	186	9.3	7.0	1	31
S^+	148	7.4	6.0	1	22
S^-	38	1.9	1.0	0	9
S^+/S	0.796	0.854	0.912	0.5	1.0

The overall number of sentences S is 186. 148 of them are considered correct and sensible (S^+), whereas the remaining 38 are not (S^-). We do not count the sentences that have been created and later removed by the same participant. If someone modified a sentence that was created by himself then we count only the last version. If a sentence was modified by another participant then the respective versions of the sentence count for each of the participants. Thus, this table shows the achievements of the individuals, not of the community.

Let us first explain how we judged whether a sentence is correct and sensible. The main criteria was that the sentence is a true statement (under the ACE semantics) of the real world using the common interpretations of the natural language words. If this is not the case, e.g. for “every musician is a man”, then the sentence counted for S^-. In the case of 24% of the incorrect sentences, words were used in the wrong category, for example “every London is a city” where “London” was introduced as a common noun instead of a proper name. Another 24% of the incorrect sentences are statements like “a city is a landscape-element” which is interpreted in ACE as having only existential quantification: “there is a city that is a landscape-element”. Even though this is a correct statement about the real world, the user probably wanted to say “every city is a landscape-element”. For that reason, such sentences were considered incorrect. (The remaining 52% of the incorrect sentences do not show specific patterns for further categorization.) On the other hand, sentences like “every country is a part of a continent” were considered correct, even though it depends on the interpretation of “part of” and “continent”. One could say that Russia is not part of a continent, but only overlaps with Europe and Asia. But in this case, there is no reason to believe that the participant wanted to say something different than what the ACE semantics defines.

The results show that almost 80% of the created sentences were correct, which is – we think – a very good result. Furthermore, every participant created at least one correct sentence. Another interesting result is that the ratio of correct sentences was in the worst case 50%. Thus, no participant created more wrong sentences than correct ones. Altogether, we can conclude that all of the participants managed to deal with AceWiki.



The participants were encouraged to create not only simple sentences, but also some complicated ones. We can now find out whether they managed to do so. Picture above shows the most frequent sentence patterns and reveals the occurrence of negation (i.e. “does not”, “is not”, “no”, or “it is false that”) and implication (i.e. “if ... then”, “every”, or “no”). Only the correct sentences are considered here.

The two most frequent sentence patterns were superconcept statements (positive e.g. “every canal is a waterbody” or negated e.g. “it is false that every animal is a mammal”) and individual assignments (positive e.g. “Zurich is a city” or negated e.g. “Bob-Dylan is not a woman”). Also quite frequent were role instances (positive e.g. “Limmat flows-through Zurich” or negated e.g. “it is false that Winston-Churchill is a prime-minister of Denmark”) and superrole statements (e.g. “if something X protects something Y then X shelters Y”). All the examples are sentences that the participants created during the experiment.

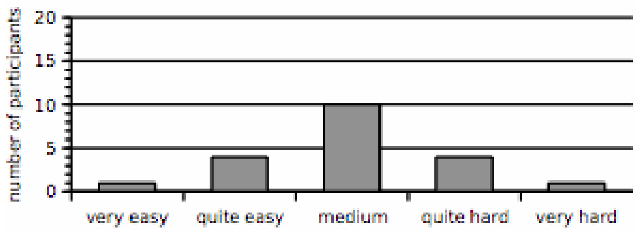
It is remarkable that there is a long tail of other sentence patterns and that 61% of the sentences contained a negation or an implication or both. This shows that the participants made use of the high expressivity of ACE.

In order to evaluate the performance of the participants, we have to take the time dimension into account. The following table shows the time values (in minutes) that we retrieved from the log files.

	total	average	median	min	max
t	931.2	46.6	39.1	7.7	132.5
t_f		11.0	8.0	2.5	33.8
t/S^+	6.3	8.2	7.2	2.9	24.7

The first line shows the overall time t of the participants. This is the time they spent on AceWiki, not counting the time for reading the instructions and for filling out the questionnaire. The second line shows the time t_f needed for creating the first correct sentence. The final line contains the time per correct sentence and shows how well the participants performed. Thus, they needed on average 11.0 minutes to create their first correct sentence, and overall the time per correct sentence was 8.2 minutes.

Those results do not look very spectacular at first sight, but we have to recall the situation of the participants. When we start counting the minutes, the participants see AceWiki for the very first time. The instructions contained no explanation whatsoever of the AceWiki interface. In order to get familiar with this unknown interface, the participants started to navigate around, searched for terms, and explored the predictive editor. Some of them added new words without adding a sentence yet, and some added a sentence but removed it again. And then, after only eleven minutes, on average, the participants managed to create their first correct sentence. Over the complete duration, they created a correct sentence approximately every eight minutes, and – as shown above – most of those sentences were quite complicated. We think that these are very good results under the given circumstances, and they show that AceWiki has indeed a shallow learning curve.



Finally, we can take a look at the participants' feedback after the experiment. In the questionnaire we asked how easy or how difficult the handling of AceWiki was. The picture above shows the result. The responses are distributed symmetrically and have a peak at “medium”. On the one hand, this is a good result since only 25% of the users found it hard to use AceWiki. We have to consider that the participants experienced only the costs of formal knowledge representation, but not the benefits (since reasoning features were missing). Furthermore, knowledge representation is inherently a difficult task. Probably, it will never be possible to make this very easy for everybody. On the other hand, the results show that there is certainly room for improvement.

3.5. Conclusions

The AceWiki prototype shows how ontologies can be managed in a natural way within a wiki. It demonstrates how semantic wikis using controlled natural language can be expressive and easy to use at the same time. Our previous evaluation showed that AceWiki is indeed easy to learn. We explained how AceWiki ensures – in a very simple way – the consistency of the ontology which is the basis for other integrated reasoning services.

4. Updates

4.1. APE

As of 6 May, 2008, the Attempto Parsing Engine (APE) is available as open source under the GNU Lesser Public Licence (LGPL), and can be downloaded from the Attempto web-page (attempto.ifi.uzh.ch/site/downloads/).

From the same web-page users can download a large lexicon of content words under the GNU Public Licence (GPL).

4.2. ACE View

Since the previous deliverable, ACE View has undergone major improvements.

- The "entailments view" has been fully implemented to show the ACE sentences that are logically entailed by the ACE text. Each entailment can be "explained" by listing sentences that cause the entailment.
- The "questions and answers view" shows all the questions that are part of the ACE text and whenever the reasoner is run and the set of entailments is generated, ACE View will act on it, i.e. will verbalise all entailments and will fill the answer tables with fresh answers. The questions are of the form "Which individuals belong to a given class?" and "Which classes are sub/super classes of a given class?", and are answered using the DL-Query technology, i.e. each question corresponds to a DL-Query (an OWL class expression), each answer is a list of named individuals/classes that either instantiate the class expression or are in a sub/super class relationship with it. These named entities are presented by ACE View simply as lists of ACE words.
- In general, the ACE View plug-in now fits better into the Protégé 4 framework – each "view" is a Protégé 4 view aware of the Protégé events, e.g. if an entity is selected in one of the Protégé 4 views (by clicking on the subclass-hierarchy, or by searching) then the ACE views automatically focus on this entity, e.g. one can filter the (possibly many) entailments to show only those related to a certain content word.

ACE View is currently available in binary form on request.

There is a screencast: http://attempto.ifi.uzh.ch/site/docs/screencast_aceview.mov

We are working towards making ACE View available under an open-source licence.

5. Cooperations and Visits

We are pleased to report that we are cooperating with several research groups who use, or plan to use, Attempto Controlled English. Here is a summary:

- Adam Wyner (University of Liverpool) visited the Attempto group on 22 April 2008 to discuss the application of ACE to argumentation.
- Rick Shiffman and Michael Krauthammer (Yale University) visited the Attempto group on 5 May 2008 to discuss details of using ACE for medical guidelines. This activity is part of the project ERGO for which Norbert E. Fuchs acts as consultant.
- The cooperation with REWERSE A2 concerning the use of ACE as query language for GoPubMed is making progress. A2 sent a set of GoPubMed queries that the Attempto group translated systematically into ACE.

6. Future Research

Future research of the Attempto group will include the tasks

- consistent and non-redundant knowledge assimilation
- extending RACE by abduction

that were listed in our original workplan.

Concerning knowledge assimilation, AceWiki automatically checks that newly added ACE sentences are consistent with the existing ones, while APE and ACE View can do a consistency check on demand.

7. References

[Fuchs & Schwertel 2003] N. E. Fuchs, U. Schwertel. Reasoning in Attempto Controlled English, in: F. Bry, N. Henze and J. Maluszynski (eds.): *Principles and Practice of Semantic Web Reasoning*, International Workshop PPSWR 2003, Mumbai, India, December 2003. Lecture Notes in Computer Science 2901, Springer Verlag, 2003.

[Fuchs et al. 2006] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Deliverable I2-D9. Attempto Controlled English 5: Language Extensions and Tools I. Technical report, REWERSE, 2006. http://rewerse.net/deliverables.html.

[Fuchs et al. 2007] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Deliverable I2-D13. Reasoning, Rules, and Semantic Wikis. Technical report, REWERSE, 2007. http://rewerse.net/deliverables.html.

[Kuhn 2007] Tobias Kuhn. AceWiki: A Natural and Expressive Semantic Wiki. Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges, CEUR Workshop Proceedings, 2008.

[Kuhn 2008] Tobias Kuhn. AceWiki: Collaborative Ontology Management in Controlled Natural Language. Proceedings of the 3rd Semantic Wiki Workshop, CEUR Workshop Proceedings, 2008.

[Siorpaes & Hepp 2007] Katharina Siorpaes, Martin Hepp. myOntology: The Marriage of Ontology Engineering and Collective Intelligence. Proc. Bridging the Gap between Semantic Web and Web 2.0, 2007.

[Sirin et al. 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A Practical OWL-DL Reasoner. Web Semantics: Science, Services and Agents on the World Wide Web, 5(2):51–53, 2007.