I2-D7

Attempto Controlled English and the Semantic Web

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/I2D5/D/PU
Responsible editor:	Norbert E. Fuchs
Contributing participants:	University of Zurich
Contributing workpackages:	I2, A2
Contractual date of delivery:	April 11, 2006
Actual date of delivery:	April 21, 2006

Abstract

This report presents three tracks of research on Attempto Controlled English (ACE). First, we show how ACE can be translated into OWL DL and how OWL DL can be verbalized in ACE. Second, we describe work done in cooperation with the working group A2 *Bioinformatics* on using ACE as ontology language for protein interactions. Third, we summarise changes to ACE and its associated tools.

Keyword List

Attempto Controlled English, ACE, OWL DL, ontology

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

Attempto Controlled English and the Semantic Web

Norbert E. Fuchs, Kaarel Kaljurand, Tobias Kuhn, Gerold Schneider

Department of Informatics

&

Institute of Computational Linguistics

University of Zurich

Email: {fuchs, kalju, tkuhn, gschneid} @iifi.unizh.ch

Loic Royer, Michael Schröder

Biotechnological Center

TU Dresden

Email: {loic.royer, ms} @biotec.tu-dresden.de

April 21, 2006

Abstract

This report presents three tracks of research on Attempto Controlled English (ACE). First, we show how ACE can be translated into OWL DL and how OWL DL can be verbalized in ACE. Second, we describe work done in cooperation with the working group A2 *Bioinformatics* on using ACE as ontology language for protein interactions. Third, we summarise changes to ACE and its associated tools.

Keyword List

Attempto Controlled English, ACE, OWL DL, ontology

Contents

1. Introduction

Please notice that we changed the title of this report from ‘Verbalising Formal Languages in Attempto Controlled English II’ to ‘Attempto Controlled English and the Semantic Web’ since the report describes three tracks of research that go beyond verbalisation.

First, we describe ongoing work on a bidirectional mapping between Attempto Controlled English (ACE) and OWL DL. ACE is a well-studied controlled language, with a parser that converts ACE texts into Discourse Representation Structures (DRS). We show how ACE can be translated into OWL DL (by using the DRS as interlingua) and how OWL DL can be verbalized in ACE. This mapping renders ACE an interesting companion to existing OWL front-ends.

Second, we report on work that was performed in cooperation with working group A2 (Dresden) on using ACE as ontology language for protein interactions, and on accessing biomedical literature. Linking the biomedical literature to other data resources is notoriously difficult and requires text mining. Text mining aims to automatically extract facts from literature. Since authors write in natural language, text mining is a great natural language processing challenge, which is far from being solved. We propose an alternative: If authors and editors summarize the main facts in ACE, text mining will become easier and more powerful. We define a simple model to capture the main aspects of protein interactions. To evaluate the model we collected a dataset of 459 paragraph headlines about protein interaction from literature. 56% of these headlines can be represented exactly in ACE and another 23% partially. These results indicate that our approach is feasible and can be built into future editors.

Third, we summarise advances concerning ACE and its associated tools. The most important change is a simplification of the DRS representation of nouns, specifically of plurals. Our REWERSE partners had requested this change.

2. A Bidirectional Mapping between OWL DL and Attempto Controlled English

2.1. Motivation

Existing OWL tools like Protégé (protege.stanford.edu), SWOOP (www.mindswap.org/2004/SWOOP) etc. are user-friendly graphical editors, but for complex class descriptions they require the user to possess a large knowledge of Description Logics (DL). [15] lists the problems that users encounter when working with OWL DL, and express the need for a 'pedantic but explicit' paraphrase language.

To answer this need, we envision a text-based system that allows the users to express the ontologies in the most natural way – in natural language. Such a system would provide a natural syntax for logical constructions such as disjointness or transitivity, i.e. it would not use keywords but instead a syntactic structure to represent those complex concepts. It would also hide the sometimes artificial distinction between an ontology language and a rule language. The system would be tightly integrated with an OWL DL reasoner, but the output of the reasoner (if expressed in OWL DL as a modification of the ontology) would again be verbalized in natural language, so that all user interaction takes place in natural language and the central role in the system is carried by plain text.

As a basis of the natural language, we have chosen Attempto Controlled English (ACE), a subset of English that can be converted through its DRS representation into first-order logic representation and automatically reasoned about [5] (see[1] for more information).

The current version of ACE offers language constructs like countable and mass nouns; collective and distributive plurals; generalized quantifiers; indefinite pronouns; negation, conjunction and disjunction of noun phrases, verb phrases and sentences; and anaphoric references to noun phrases through proper names, definite noun phrases, pronouns, and variables. The intention behind ACE is to minimize the number of syntax and interpretation rules needed to predict the resulting DRS, or for the end-user, the reasoning results. At the same time, the expressivity of ACE must not suffer. The small number of ACE function words have a clear and predictable meaning and the remaining content words are classified only as verbs, nouns, adjectives and adverbs. Still, ACE has a relatively complex syntax compared to the OWL representation e.g. in the OWL Abstract Syntax specification [14], but as ACE is based on English, its grammar rules are intuitive (already known to English speakers) and experiments show that ACE can be learned in a few days. [2] show also that users are likely to prefer ACE to visibly formal languages such as SQL.

Our work addresses the following issues:

1. Show that there is a mapping from a subset of ACE (which we call OWL ACE) into a syntactic subset of OWL DL (i.e. a subset which does not use all the syntactic constructs in OWL DL but is still capable of expressing everything that OWL DL can express).
2. Show that the two involved subsets and the mapping from one to the other are easy to explain to the users. This means that the entailment and consistency results given by the OWL DL reasoners "make sense" on the ACE level.
3. Show that there is a mapping from the syntactic subset of OWL DL into OWL ACE. This mapping (which can be called a verbalization) must, again, be easily explainable.
4. Implement a converter from OWL DL to the chosen syntactic subset of OWL DL. By this, we will be able to handle all OWL DL ontologies on the web.
5. If needed, extend ACE to provide a more natural syntax or more syntactic variety for expressing the OWL DL constructs.
6. Extend the verbalization process to target a richer syntactic subset of OWL ACE.
7. Extend all the aspects of this mapping in order to be compatible with future standards of OWL DL, e.g. OWL 1.1 [13] or extensions of it, e.g. SWRL [9].

So far, we have focused on the first 3 steps. In the following, we describe a mapping from OWL ACE to OWL DL (in RDF/XML syntax), the problems encountered, the OWL ACE subset and the verbalization of OWL DL.

2.2. From ACE to OWL

The Attempto Parsing Engine (APE) translates the ACE text

```
Bill who is a man likes himself. Bill is William. Every businessman who
is richer than at least 3 things is a self-made-man or employs a
programmer who knows Bill.
```

(Note that the example is somewhat artificial to demonstrate concisely the features of OWL DL as expressed in ACE.)

into the DRS

```
[A, B, C, D, E, F]
object(A, atomic, named_entity, person, cardinality, count_unit, eq, 1)
named(A, Bill)
object(C, atomic, man, person, cardinality, count_unit, eq, 1)
predicate(E, state, be, A, C)
predicate(B, unspecified, like, A, A)
object(D, atomic, named_entity, person, cardinality, count_unit, eq, 1)
named(D, William)
predicate(F, state, be, A, D)
   [G, H, I, J]
   object(H, atomic, businessman, person, cardinality, count_unit, eq, 1)
   predicate(J, state, be, H, I)
   property(I, richer_than, G)
   object(G, group, thing, object, cardinality, count_unit, geq, 3)
   =>
   []
      [K, L]
      object(K, atomic, self-made-man, person, cardinality, count_unit, eq, 1)
      predicate(L, state, be, H, K)
      v
      [M, N, O]
      object(M, atomic, programmer, person, cardinality, count_unit, eq, 1)
      predicate(N, unspecified, know, M, A)
      predicate(O, unspecified, employ, H, M)
```

The DRS [3] for a complete overview of the DRS language used to represent ACE texts) makes use of a small number of predicates, most importantly *object* derived from nouns and *predicate* derived from verbs. The predicates share information by means of discourse referents (denoted by capital letters) and are further grouped by embedded DRS-boxes, that represent implication (derived from ‘every' or ‘if... then...'), negation (derived from various forms of English negation), and disjunction (derived from ‘or'). Conjunction – derived from relative clauses, explicit ‘and', or the sentence end symbol – is represented by the co-occurrence in the same DRS-box.

The mapping to OWL DL does not modify the existing DRS construction algorithm but only the interpretation of the DRS. It considers everything in the top-level DRS to denote individuals (typed to belong to a certain class), or to denote relations between individuals. Individuals are introduced by nouns, so that propernames (‘Bill', ‘William') map to individuals with type *owl:Thing* and common nouns to an anonymous individual with the type derived from the corresponding noun (e.g. class *Man*). Properties are derived from transitive verbs (‘likes') and transitive adjectives. Special meaning is assigned to the copula ‘be' which introduces equality between individuals.

An embedded implication-box introduces a *subClassOf*-relation between class descriptions – the head of the implication maps to a class description, the body to its superclass description. Transitive verbs (‘employ', ‘know') and transitive adjectives (‘richer than') introduce a property restriction with *someValuesFrom* a class denoted by the object of the verb or adjective, and the copula introduces a class restriction. Co-occurrence of predicates maps to *intersectionOf*. Negation and disjunction boxes introduce *complementOf* and *unionOf*, respectively. Any embedding of them is allowed. The plural form of the word ‘thing' and the usage of numbers and generalized quantifiers (‘more than', ‘less than', ‘at least', ‘at most') allow to define cardinality restrictions. Thus our DRS has the following meaning (in DL notation):

$$\text{bill} \in \top$$
$$\text{m1} \in \text{Man}$$
$$\text{william} \in \top$$
$$\text{bill} = \text{m1}$$
$$\text{bill} = \text{william}$$
$$\text{likes}(\text{bill}, \text{bill})$$

$$(\text{Businessman} \sqcap \text{isRicherThan} \geq 3) \sqsubseteq$$
$$(\text{SelfMadeMan} \sqcup (\exists\ \text{employs}\ (\text{Programmer} \sqcap (\exists\ \text{knows}\ \{\text{bill}\}))))$$

Note that an ACE construct like ‘A man who owns a dog likes an animal.' describes relationships between individuals and not classes, since the corresponding DRS does not have any embedded DRSs. In full English, this sentence is ambiguous by also having a reading which relates classes. In ACE, one would have to use ‘every' instead of ‘a' to get this reading.

OWL ACE allows properties to have superproperties. A superproperty (e.g. ‘likes') for a given property (e.g. ‘loves') can be defined as:

```
Everybody who loves somebody likes him/her.
```

Describing the transitivity of properties and inverse properties is quite ‘‘mathematical'' in ACE, but there does not seem to be a better way in natural languages, unless one defines keywords such as ‘transitive' or ‘inverseOf'' which then have to be explained to the average users. Consider e.g.

```
If a thing A is taller than a thing B and B is taller than a thing C then
A is taller than C.
```

```
If a thing A is taller than a thing B then B is shorter than A. If a
thing A is shorter than a thing B then B is taller than A.
```

Note that property definitions make use of indefinite pronouns (‘everybody', ‘somebody') or a noun ‘thing', which all map to *owl:Thing*.

The current mapping does not target all the syntactic variety defined in the OWL DL specification, e.g. elements like *disjointWith* or *equivalentProperty* cannot be directly expressed in ACE, but their semantically equivalent constructs can be generated.

2.3. Problems and Missing Features

Now we look at some of the problems that we have encountered when implementing the mapping from ACE to OWL DL. On the one hand, some expressions that can be concisely handled in OWL DL do not have an elegant counterpart in ACE. This calls for an extension of the grammar of ACE. On the other hand, some DRS structures cannot be directly mapped into OWL DL syntax which differs from DRS syntax by being heavily influenced by the standard Description Logics' syntax. This calls for a preprocessing of the DRS structures.

The biggest problem that we have encountered is that *allValuesFrom* cannot be expressed in ACE in the most natural way, i.e. by using constructions like ‘only', ‘nothing but' or ‘nothing else than'. Note that existing approaches to verbalizing *allValuesFrom* tend to use ‘only' (see [15]) and ‘always' (see [6]). ACE has excluded ‘only' even as a general adverb, in order to reduce the possible ambiguity that this word might introduce. Therefore a concise form to express e.g. the statement Carnivore = ∀eat.Meat is missing in ACE.

```
*Every carnivore eats only meat.
```

```
*Everything that eats only meat is a carnivore.
```

In order to express this meaning, the ACE user can choose double negation (essentially using the equivalence $\forall R.C = \neg \exists R.\neg C$) or an *if-then* construction (essentially using the mapping Φ to first-order logic syntax $\Phi_{\forall R.C}(x) = \forall y.R(x,y) \rightarrow \Phi_C(y)$. E.g. the DL statement Carnivore $\sqsubseteq \forall$eat.Meat can be expressed in ACE in the following way (the equality sign points to a different formulation that gives exactly the same DRS representation)

```
No carnivore eats something that is not a meat.

(= If there is a carnivore then it does not eat something that is not a
meat.)

Everything that a carnivore eats is a meat.

(= If a carnivore eats something then it is a meat.)

For every carnivore everything that it eats is a meat.

(= If there is a carnivore then everything that it eats is a meat.)
```

The opposite direction, i.e. the DL statement $\forall$eat.Meat $\sqsubseteq$ Carnivore can be expressed in ACE as

```
If there is a thing that does not eat something that is not a meat then
the thing is a carnivore.

If there is something and everything that it eats is a meat then it is a
carnivore.
```

Some of those constructions might even be acceptable in verbalizations of existing ontologies or paraphrases of existing ACE texts (i.e. they might be suitable for reading and confirmation), but they are unacceptable as the only way to express *allValuesFrom* in ACE.

Some problems emerge from the difference of the Description Logics' syntax and the DRS syntax. E.g. complex class descriptions as arguments to *someValuesFrom* are difficult to map to OWL DL, since the DRS representation resembles more a rule language than a DL-style property restriction.

The ACE negation does not generate an implication-box, but for class descriptions like 'No man is a woman.' it would be desirable. Therefore, we first convert the negation-box into an implication-box (containing a negated *then*-part).

The fact that *inverseOf* is symmetrical is also difficult to implement because the ACE-way of expressing this creates two implication-boxes which have to be handled as one unit in the mapping.

Some OWL DL features are missing altogether. Currently, there is no support for enumerations (*oneOf*). One possibility would be to extend ACE with NP disjunction.

```
*Every student is John or Mary or Bill.

*Everybody likes John or Mary or likes John or Bill.

*Everybody who is John or Bill is a man and is a student.
```

Also, at this point, ACE has no support for datatype properties. One could imagine using ACE's *of*-construction (or Saxon genitive) for that purpose, e.g.

```
John's age is more than 21 years.

If somebody drinks some beer then his own age is more than 21 years.
```

And finally, metalevel constructions such as URIs, imports, annotation properties, versioning, etc, which essentially make OWL DL a Semantic Web language cannot be cleanly expressed in ACE.

2.4. Explaining OWL ACE

As is the case with full ACE, in order to be successful, OWL ACE must be easy to learn for the average users. This means that the user can quickly resolve the syntactic and semantic errors that he encounters when inputting an (OWL) ACE text. Assuming that full ACE has achieved the required simplicity we now look at the various restrictions to OWL ACE as compared to full ACE.

Some of those restrictions are easy to explain: there is no support for intransitive and ditransitive verbs, prepositional phrases, adverbs, intransitive adjectives, and most forms of plurals. Also, query sentences (e.g. ''Who employs Bill?") are not allowed in OWL ACE.

In addition, there are constraints on the DRS structure which might be difficult to explain to the average user. E.g. disjunction is not allowed to occur at the toplevel DRS and negation at the toplevel

is handled by converting it first into an implication, or alternatively, as a negation of the equivalence of individuals. A further restriction requires the predicates in the implication-box which defines a subclass relation between class descriptions to share one common discourse referent as the subject argument, unless the subject is directly or indirectly an object of a predicate that binds it to the common subject. This allows us to exclude sentences like 'If a man sees a dog then a cat sees a mouse.' but to include sentences like 'If a man sees a dog that sees a cat then the man sees a mouse.' The first sentence does not seem to map nicely to OWL DL but instead to a more powerful rule language (such as SWRL). Also, no subject can occur as an object and no object can be repeated in the implication-box ('Every man hates a dog that bites him.')

An ACE level suggestion to avoid the unwanted implication structure is to use only *every*-sentences which put a natural restriction on how the subject can be used in the sentence. *Every*-sentences can express complex structures via relative clauses (which can be conjoined, disjoined or negated using verb phrase conjunction, disjunction or negation, respectively). A further restriction is to avoid any kind or anaphoric references, apart from relative pronouns ('who', 'which', 'that') or references to top-level objects (i.e. individuals).

2.5. From OWL to ACE

The mapping in the opposite direction must handle all OWL DL constructs, some of which the ACE-to-OWL mapping does not produce. A bigger issue is raised by the naming conventions used for OWL classes and properties. Those names are not under the control of current OWL editing tools and the user is guided only by informal style-guides, which mainly discuss the capitalization of names (see e.g. [8]). OWL ACE would prefer classes to be named by singular nouns, and properties by transitive verbs or adjectives. Real-world OWL ontologies, however, can contain class names like *SpicyPizza*, *MotherWith3Children* and property names like *accountName*, *brotherOf*, *isWrittenBy*. Still, [12] analyze the linguistic nature of class and property names in real-world OWL ontologies and find that those names fall, in most cases, quite well into the categories of nouns and verbs, respectively, with only a small overlap in linguistic patterns used.

Mapping from OWL to ACE also involves parsing RDF, which is the normative syntax for OWL DL. So far, we have implemented a simple prototype in XSLT, which generates ACE from the XML Presentation Syntax of OWL [7] and hope that more OWL tools will support the Presentation Syntax as an alternative output format. The current mapping directly generates ACE. An alternative would target the DRS instead, and use an existing general mapping from the DRS to a canonical ACE form (the so called Core ACE form) [4].

Currently, the ACE representation ends up being quite repetitive and unordered. For large ontologies this might become a problem and a more complex strategy is needed. Consider e.g. the following sentences.

```
Every wine which originates-from France is a french-wine.

Everything which is a wine and which originates-from France is something
which is a french-wine.

If there is a wine and it originates-from France then it is a french-
wine.

If there is a wine W and W originates-from France then W is a french-
wine.
```

Those sentences are equivalent, as far as the mapping to OWL DL is concerned. Still, one could argue that some of those sentences are more readable than others, e.g. the *every*-construction with a relative clause is more readable than the *if-then*-constructions with full clauses. On the other hand, relative clauses cannot express more complex structures (without causing ambiguity in the output), thus the more general *if-then* construction must be used. A flexible ACE generation system could use relative clauses in case they allow to correctly express all the references in the DRS and revert to using *if-then* sentences in case a more flexible reference system is needed. It might turn out that the expressivity provided by *every*-sentences (using relative clauses) is enough to verbalize OWL DL.

Note also, that a variety of different verbalizations can be achieved by changing the input ontology with a reasoner which restructures the ontology and/or modifies it by adding/removing certain (possibly redundant) information. I.e. we could provide a relatively direct OWL-to-ACE mapping, but use a reasoner to customize the verbalization procedure for our needs.

2.6. Related Work

Some existing results show the potential and the need for a natural language based interface to OWL, and to the Semantic Web in general. [10] discusses the so-called ‘‘people axis'' of the Semantic Web, i.e. technologies which would make the Semantic Web accessible to the widest possible audience. He describes Pseudo Natural Language which provides an interface to RDF, and points to the need for a dedicated natural interface to extensions of RDF, such as OWL. [16] proposes writing OWL ontologies in a controlled language, but does not provide a natural syntax for writing terminological statements (i.e. TBoxes). TRANSLATOR (www.ruleml.org/translator) is a tool which maps the DRS representation of ACE sentences into RuleML syntax, covering full ACE.

There is more work on the verbalization of ontologies, although not in controlled languages, so that the output of such systems cannot be edited and parsed back into a standard OWL representation. [11] discuss inferences (so called *natural language directed inference*) to be applied on the ontology which are necessary to make the verbalization of the ontology linguistically more acceptable, e.g. the verbalization must not violate the Gricean maxims. [6] paraphrase OWL class hierarchies and use a part-of-speech tagger to analyze the linguistic nature of class names and then split the names apart to form more readable sentences.

2.7. Future Work

The current mapping lacks support for datatype properties and enumerations. Also, *allValuesFrom* cannot be directly generated, but its semantics can be achieved by using double negation. We will add support of those constructs along with support of proposed extensions to the current version of OWL DL, such as qualified cardinality and local reflexivity restrictions. Some of those changes require modification of the ACE syntax. ACE also needs support for namespaces, at least at the tokenizer level, to be called a Semantic Web language.

The ACE parser uses currently a large lexicon of content words to know which words belong to which part-of-speech. ACE texts containing domain specific words cannot be parsed unless the built-in general-purpose lexicon is updated to contain knowledge about these words. This makes parsing faster and allows us to point out spelling mistakes. On the other hand, the dependency on the lexicon can make the system less convenient to use. The restrictions that the OWL ACE subset of ACE sets on ACE syntax, might be strong enough, so that the part-of-speech information of the words could be unambiguously derived from their context (e.g. a determiner such as ‘every', ‘a' or ‘no' signals that the following word is a noun). We are thus in search for a lexicon-independent subset of ACE and explore its relation to OWL ACE.

We will also study if the OWL ACE subset is easier or harder to teach to the users than full ACE.

3. Improving Text Mining with Controlled Natural Language: A Case Study for Protein Interactions

3.1. Introduction

We introduce a new paradigm of how to make knowledge of scientific papers accessible by computers. We focus on the fields of life sciences – particular biology – but our approach could be used in other fields as well.

Our approach consists of letting authors express their scientific results in a formal summary that could be an integral part of the papers they publish. We argue that it is more reasonable to let the authors formalize their own results, instead of trying to extract these results from the articles.

This section explains our motivation, introduces the language Attempto Controlled English (ACE) and compares it with other knowledge representation languages. Section 3.2 shows how ACE is used to build an ontology for protein interactions. In section 3.3 we use this ontology as foundation for the expression of scientific results and we show how 89 selected articles could have been summarized in ACE. Section 3.4 shows the benefits of our approach and section 3.5, finally, gives a short outlook.

3.2. Motivation

Biomedical scientists are challenged by an ever-increasing amount of scientific papers. The indexing service *PubMed* (www.pubmed.gov) shows the huge quantity of literature that the scientists have to face. It contains at the moment about 16 million articles and grows every year by over 600'000 articles. All these biomedical articles are written in natural language. That means that we cannot easily process them with computers. But, facing the quantity of literature, it is clear that we need computational support in order to manage the contained knowledge.

In the last years, *text mining* and *information extraction* – which build both upon natural language processing (NLP) – gained an increasing interest in biomedical sciences. They aim to extract some kind of formal knowledge from natural language texts, which is generally considered a very demanding task. Even the basic problem of *named entity recognition*, that aims to identify named entities (e.g. protein names) in natural texts, is far from being solved. Other major aspects of text mining are the extraction of relationships (e.g. protein interactions), the automatic classification of texts, and the generation of new hypotheses on the basis of the available literature [18]. The *BioCreAtIvE* contest nicely shows, that even sophisticated tools for text mining have a considerable lack of precision and recall: For a simple *named entity recognition*-task the precision ranged up to 86% and the recall was at most 84% [27]. Another attempt is described in [19]: Information about protein-interactions was extracted from a data set of 1.2 million sentences that were taken from biomedical abstracts. They achieved a precision of 91%, but with a poor recall of only 21%.

As a first step towards a better management of biomedical literature, controlled vocabularies like *MeSH* (www.nlm.nih.gov/mesh/meshhome.html) and the *Gene Ontology (*www.geneontology.org) have been created. They serve to classify biomedical publications and to link them to other resources. *GoPubMed* (www.gopubmed.org), for example, is a search engine that connects the abstracts from PubMed with the formal structure of the Gene Ontology. Thus a researcher can exploit the Gene Ontology for the search of relevant literature. Such tools are very valuable for scientists and there has been a notable progress in the last years, but it will never be possible to extract all the information correctly. There is inherent ambiguity and vagueness in natural language that prevents its perfect processing by computers.

For this reason we present an alternative approach: The authors of scientific articles formally summarize their own results. Such formal summaries are added to the articles which makes them processable by computers. This requires a formal language that on the one hand is easy to learn and understand, and on the other hand is expressive enough to represent even complicated scientific results.

It is clear that this approach is not applicable for papers that have been written without the formal summaries, and that means that we still need NLP or manual extraction for such papers. Thus we propose rather a concept for the future than a solution for today's problems. To explore our approach we use Attempto Controlled English as knowledge representation language.

3.3. Formalization of Scientific Results

Since we want to access scientific results by computers, we have to formalize this knowledge at some point. Today researchers write their results in natural language. To extract these results and to formalize them, manual or computer-supported text mining is necessary. Thus the formalization is accomplished by computer-programs or by humans, and in either case it is done without the help of the corresponding researchers. The article is the only source of information. Since such articles are highly domain-specific, they require a lot of background knowledge. Therefore the formalization is a very demanding task, even for humans. Altogether this causes a lot of knowledge to be lost in the vast amount of biomedical literature.

We claim that most of these problems can be solved, if we simply let the authors of scientific articles formalize their own results. The researchers themselves are the most qualified to understand their results, and thus they can give the most precise formal representation. This is not even a big extra-effort for a scientist, since he already has a - more or less – formal model of the domain in his mind, and must write an abstract anyway. He just needs to learn how to express his knowledge in a formal way. This means that we need to provide an intuitive, yet formal language in which a scientist can write his results.

3.4. Attempto Controlled English

Attempto Controlled English (ACE) ([20] and www.ifi.unizh.ch/attempto) is a controlled natural language, i.e. a subset of natural English with a restricted grammar. It does not make any restrictions on the vocabulary, apart from defining the meaning of some function words like ‘every' or ‘of'. ACE looks like English, but it is in fact a formal language. ACE texts can be translated unambiguously into first-order logic.

ACE provides interpretation rules that define the semantics of ACE texts. Some ACE sentences would be ambiguous in natural English, but the interpretation rules of ACE allow only one interpretation.

In order to be able to write ACE texts, one has to learn the restrictions on the grammar and on the vocabulary. Thus, like every formal language, ACE has to be learned. However, since it looks like natural English, everyone is able to understand ACE texts with almost no training. This is a big advantage over other formal languages.

ACE text can be translated into a representation of first-order logic. This is done by the Attempto parser APE (www.ifi.unizh.ch/attempto/tools/cape.html). APE generates a logical representation of ACE text and uses for this representation Discourse Representation Structures [21]. Such structures are equivalent to expressions in first-order logic.

Furthermore, APE creates a paraphrase that shows the interpretation of an ACE text. If a writer is not familiar with the ACE interpretation rules, then he can check the paraphrase for the validation of his ACE text.

3.5. Comparison of Knowledge Representation Languages

In order to show the benefits of ACE, we compare it with three other knowledge representation languages: first-order logic, Description Logics (DL), and Web Ontology Language (OWL).

First-Order Logic. Logic is a very old discipline that originated in philosophy. First-order logic is by far the most widely used, studied, and implemented version of logic [25]. The main advantages of first-order logic are its expressiveness, its thorough formal foundation, and the huge amount of theoretical results.

Description Logics. Description Logics are a family of knowledge representation languages [23]. As the name suggests, DL builds upon classical logic, i.e. the semantics of the DL notation are founded in logic.

The basic elements of DL are *individuals*, *concepts*, and *roles*. Individuals stand for single objects, concepts stand for classes of objects, and roles for binary relations between objects. Table 1 shows the basic elements of DL with their equivalents in first-order logic and their common notation. We will use these three types of elements as foundation for ontologies in ACE.

Table 1. Basic elements of DL with their equivalents in first-order logic (FOL)

DL Element	FOL Equivalent	Examples
individual	constant	$JOHN$, $GERMANY$
concept	unary predicate	$Human$, $Protein$
role	binary predicate	$uses$, $brother\_of$

Web Ontology Language OWL. OWL is a set of knowledge representation languages defined by the World Wide Web Consortium (www.w3.org/2004/OWL). It builds upon the well-known markup language XML, and it has a powerful, but complicated syntax. OWL belongs to the *semantic web* project that has the objective to give machine-understandable meaning to web contents (www.w3.org/2001/sw/).

We can now compare the four introduced knowledge representation languages: first-order logic, DL, OWL, and ACE. Note that these four languages are not independent. DL and ACE build upon first-order logic, and OWL is inspired by DL. While first-order logic and DL focus on the theoretical concepts of knowledge, OWL and ACE concentrate on the implementation and application of knowledge representation. Nevertheless we dare to give a direct comparison between these four languages.

Table 2 shows how the fact ‘everyone who is a manager has a car' is expressed in the four different languages. The OWL representation (using the XML syntax) is the most verbose and – from the human perspective – the least readable one. The representations in first-order logic and DL are more concise, but they are still not understandable for people who are not familiar with formal notations. The ACE representation, in contrast, should be immediately understandable for any English speaking person. It looks perfectly like natural English and thus the reader might not even recognize that it is a formal language.

Table 2. Example in first-order logic (FOL), DL, OWL, and ACE

FOL	$\forall X(manager(X) \rightarrow \exists Y(car(Y) \wedge has(X,Y)))$
DL	$Manager \sqsubseteq \exists has.Car$

OWL	`<owl:Class rdf:ID="Manager">`
	`  <rdfs:subClassOf>`
	`  <owl:Restriction>`
	`  <owl:onProperty rdf:resource="#has"/>`
	`  <owl:someValuesFrom rdf:resource="#Car"/>`
	`  </owl:Restriction>`
	`  </rdfs:subClassOf>`
	`</owl:Class>`

ACE	Every manager has a car.

We can state that controlled natural languages like ACE minimize the gap between machines and humans. A reader is able to understand such languages with almost no training. Furthermore, writing sentences in a controlled natural language is possible with only little effort, especially if the writer is supported by a authoring tool (see section 3.7.3).

3.6. ACE Ontology for Protein Interactions

We need a clear foundation for the formal representation of knowledge about protein interactions. For that reason we have defined an ontology which we can use to summarize scientific articles. This

section shows how ACE can be used as an ontology language, and how to build a terminology for protein interactions.

3.6.1. Ontologies

The term *ontology* is adopted from philosophy and denotes the study of existence and of its basic categories. In computer science the term is used for the formal representation of a certain domain of the real world. An ontology specifies the basic elements – i.e. the entities of the domain and their interrelations – which are needed for a formal representation of knowledge.

The main goal of an ontology is to provide a *shared understanding* of a certain domain. This shared understanding can serve as basis for the communication between people, for the interoperability between systems, for the improvement of reusability and reliability of software systems, and for the specification of software [26]. Furthermore ontologies are an excellent basis for the formal representation of knowledge [22].

Ontologies are not yet broadly established in science, but they are expected to gain a very important role in the future, especially in life sciences.

3.6.2. ACE as an Ontology Language

In order to provide basic structures for ontologies in ACE, we adopt the elements from DL: *individuals*, *concepts*, and *roles*. We call them *ontology elements*, and use them to represent the foundation of ontologies in ACE.

In a first step we will show how these ontology elements are expressed in ACE. For that reason we need to create a lexicon that defines the words used in the ontology. We introduce a new lexicon format that we call Ontology Lexicon Format (OLF), and that allows us to define individuals, concepts, and roles as well as their representations in ACE. In addition, the lexicon allows us to specify the hierarchy of concepts and roles, and to define the domain and range for each role. Altogether an OLF lexicon defines the basic structure of the ontology. This information is needed for the authoring tool that we will present in section 3.7.3.

The user does not need to know about the OLF lexicon. The authoring tool supports the users in writing ACE texts and hides the technical details. It helps them to choose the right words from the lexicon, to use these words as intended by the ontology, and to write texts that are compliant with the ACE syntax.

3.6.3. Ontology Elements

As already mentioned, we adopt the basic elements of DL: individuals, concepts, and roles. Now we show how these ontology elements are expressed in ACE. Furthermore we introduce an additional structure: context information.

Individuals. Individuals stand for single objects of the domain. They are represented in ACE as *proper names* like `IRAK2' or `Alzheimer'.

Concepts. Concepts stand for classes of objects and there are two possibilities how to express them in ACE. *Common nouns* are the most straightforward way. The noun ‘protein', for example, can stand for the concept of all proteins. As a second possibility we can use *positive forms of adjectives*. The adjective ‘organic', for example, might be used for the concept of all organic substances.

Roles. Roles stand for binary relations between objects, and they can be expressed in four different ways. First of all, we can use *transitive verbs* for expressing roles. For example, we can use ‘interacts-with' to express a relationship between proteins. Next, we can combine transitive verbs with *adverbs*. For example, we can use the adverb ‘directly' together with the transitive verb ‘interacts-with' to express the role ‘directly interacts-with'. As a third possibility we can use *of-construct*s like ‘is a part of'. Due to the syntax of ACE, ‘of' is the only allowed preposition for nouns. Finally, we can use *constructs with comparative forms of adjectives* like ‘is larger than'. Such constructs typically represent transitive relationships.

Context Information. The examination of the results of scientific papers on protein interactions showed that normal roles are often not sufficient to express the needed information. We can express simple statements like ‘P1 interacts-with P2', but we cannot express statements with contextual information like ‘P1 interacts-with P2 in Yeast' or ‘P1 interacts-with P2 in Microfilament for Motor-Activity'. In order to be able to express such results, we want to allow roles to have such additional information. In natural English we usually express such information with prepositional phrases, and

this is exactly the way we will do it in ACE. Figure 1 illustrates the examples without and with context information.

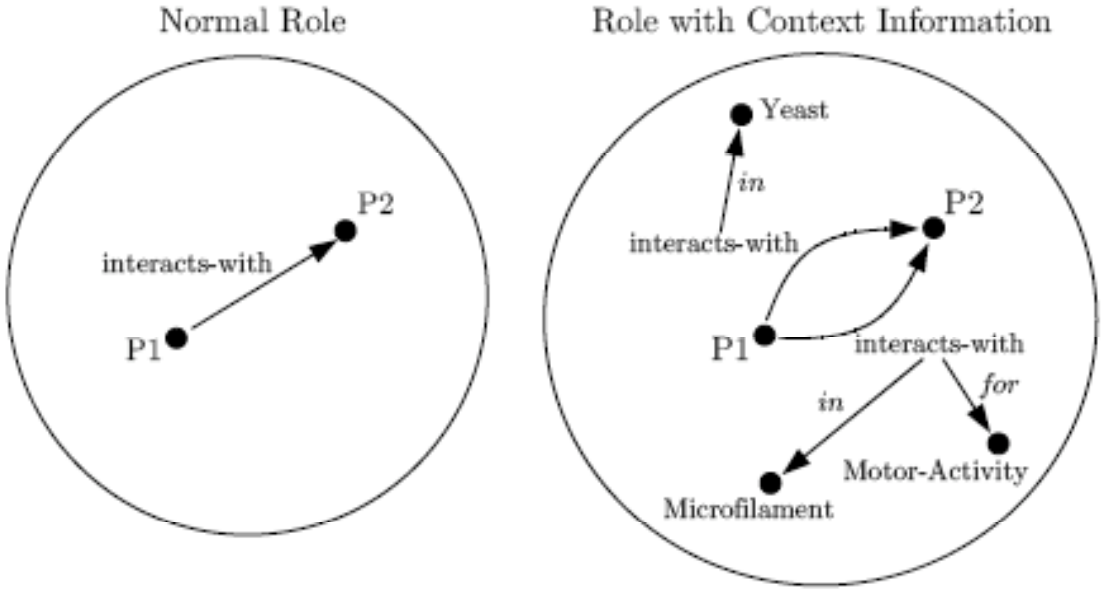


Fig. 1. Normal roles and roles with context information

Using these ontology elements, we can express for example the sentence

```
P1 is a protein and directly interacts-with P2 in Yeast.
```

where 'P1', 'P2', and 'Yeast' are individuals, 'protein' stands for a concept, and 'directly interacts-with' stands for a role. The phrase 'is a' is used to assign the individual 'P1' to the concept 'protein'. The conjunction 'and' connects the statements flanking left and right. The preposition 'in', finally, connects to the context 'Yeast'.

3.6.4. Terminology for Protein Interactions

In order to be able to express knowledge about protein interactions, we need to create a terminology. First, we define terms that allow us to make statements about the structure of proteins and protein-complexes. For the sake of a clear structure, we introduce the term *protein-unit*, which is either a protein or a protein-complex, and the term *protein-component*, which is either a protein-unit or a region of a protein. In order to describe the structure of such regions, we define terms like `residue', `secondary-structure', or `domain'.

Next we define the terms for the description of interactions between proteins like `interacts-with' or `binds'. We can also express more complicated interactions like `increases the phosphorylation of'.

Furthermore, we define some terms for expressing additional information about proteins, like the localization to a certain cellular component or the participation in a certain process. The big picture of this terminology for protein interactions is shown in Figure 2.

3.7. ACE Summaries

Our goal is to show how scientists could write formal summaries of their results. Some questions come naturally to the mind: What are these results about? How complex is it to formulate them in a formal language? In the following we present an empirical study of the feasibility of our approach.

3.7.1. ACE Summaries for 89 Selected Articles

Since we want to show how results of papers about protein interactions could have been written in ACE in the first place, we picked 89 articles that concern protein interactions. Such articles mostly have a section called "Results" which is subdivided into subsections. The headings of these subsec-

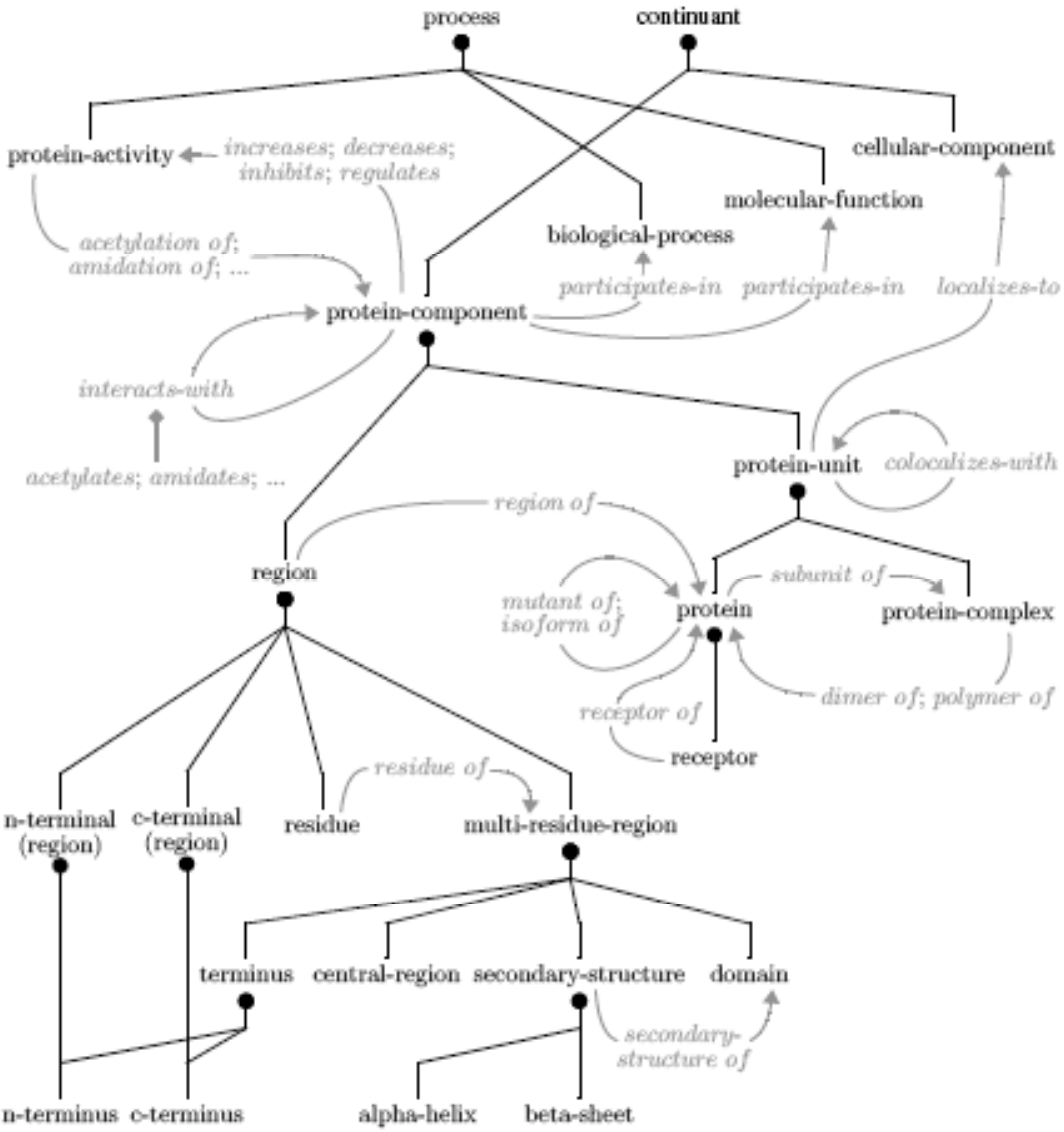


Fig. 2. The structure of the ontology for protein interactions

tions are short descriptions of the corresponding results. It turned out that these headings are highly suitable for a manual translation into ACE. Please note that the intended methodology is *not* to express the results first in natural language and then to translate them into ACE. We do this just to demonstrate the feasibility of our approach.

The 89 articles contain 457 such headings. 184 of them are ignored, because they are not formulated as facts (e.g. “Functional characterization of Pellino2” (see article *PMID 12860405*) or because they contain information that is not about protein interactions.

total:		457	*(100%)*
ignored:	(not a fact)	87	*(19%)*
	(off-topic)	97	*(21%)*
used:		273	*(60%)*

We then tried to translate the 273 remaining headings into ACE. For 154 of them there is a perfect match, which means that the complete information can be expressed in ACE; e.g. the heading “Interaction of Act1 with TRAF6’” (see article *PMID 12459498*) can be rephrased perfectly as “Act1

interacts-with TRAF6". For another 62 headings only a part of the information is expressed; e.g. the heading "The mtFabD protein is part of the core of the FAS-II complex" (see article *PMID 16213523*) can only partially be rephrased as "MtFabD is a subunit of FAS-II". For the remaining 57 headings there is no translation at all.

used:		273	*(100%)*
matched:	(perfect)	154	*(56%)*
	(partial)	62	*(23%)*
unmatched:		57	*(21%)*

Let us take a closer look at the reason, why 119 headings cannot be re-phrased in ACE at all, or only partially. 56 of them could not be rephrased because their content is not covered by our model, but they could be expressed with an extended model. Another 21 headings describe relations of relations, like the heading "Kal-GEF1 activation of Pak does not require GEF activity" (see article *PMID 15950621*). In this case, there is a relation between two objects ("Pak activates Kal-GEF1") and this relation itself stands in another relation ("... does-not-require GEF-activity"). We cannot express such structures in ACE. In order to be able to express such relations of relations in a satisfying way, we would need to extend the language ACE. Furthermore there are 11 headings with fuzzy statements (e.g. "ANKRD contains potential CASQ2 binding sequences ..." (see article *PMID 15698842*) and 31 headings that we did not understand.

not perfectly matched:	119	*(100%)*
not covered by our model:	56	*(47%)*
relations of relations:	21	*(18%)*
fuzzy:	11	*(9%)*
not understood:	31	*(26%)*

Thus, altogether we could rephrase 79% of the relevant headings, either partially or perfectly. This makes us confident that our approach is feasible for practical use. The reason, why 119 headings are not rephrased perfectly, is mostly our simple model and our lack of understanding. If we used a more detailed model, and if we let the scientists themselves express their own results in ACE, then we expect to be able to express much more than 79% of the results.

3.7.2. ACE Summary as an Integral Part of an Article

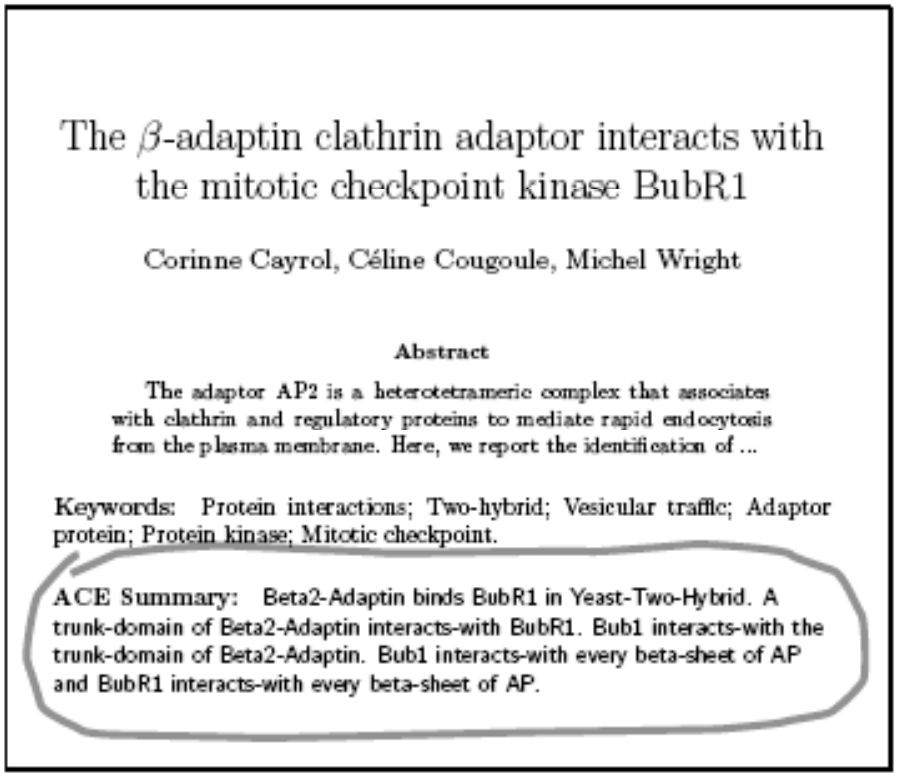
Since ACE looks like natural English, every reader of a scientific article is able to understand ACE texts. Thus the ACE summary of the results could be an integral part of the article. Together with the abstract and a keyword list, the ACE summary gives a concise insight into the content. Figure 3 shows how an article with an ACE summary could look like.

In contrast to the abstract, the ACE summary is readable by both, humans and machines; and in contrast to the keyword list, the ACE summary does not only mention the objects of interest, but describes the relations among them. Thus, every published article could be a contribution to a constantly growing knowledge base.

3.7.3. Authoring Tool

Now we sketch a tool that would help writing ACE texts using an OLF lexicon. Such a tool would allow the user to write his results in ACE with almost no training. It would be similar to the look-ahead editor ECOLE [24], and it would solve several problems.

- The tool would help the user to comply with the standard nomenclature. The user would only be allowed to use the defined words. It would also prevent typing errors.
- It would make sure that the created sentences comply with the ACE syntax. At every stage, the tool would allow to proceed only in a way that leads to a correct ACE sentence. Thus the user would not need to know about the syntax of ACE.
- The tool would be aware of the structure of the ontology. It would take this information from the OLF lexicon. In this way it would make sure, for example, that the domains and ranges of roles are respected.



The β-adaptin clathrin adaptor interacts with the mitotic checkpoint kinase BubR1

Corinne Cayrol, Céline Cougoule, Michel Wright

Abstract

The adaptor AP2 is a heterotetrameric complex that associates with clathrin and regulatory proteins to mediate rapid endocytosis from the plasma membrane. Here, we report the identification of ...

Keywords: Protein interactions; Two-hybrid; Vesicular traffic; Adaptor protein; Protein kinase; Mitotic checkpoint.

ACE Summary: Beta2-Adaptin binds BubR1 in Yeast-Two-Hybrid. A trunk-domain of Beta2-Adaptin interacts-with BubR1. Bub1 interacts-with the trunk-domain of Beta2-Adaptin. Bub1 interacts-with every beta-sheet of AP and BubR1 interacts-with every beta-sheet of AP.

Fig. 3. Article with ACE summary: The frontpage of an article with an ACE summary could look like this. For this demonstration the article [1] is used

We give now an example how this tool could be used. Suppose that an author of a scientific paper wants to write down the fact that the protein YETI binds to Kinesin proteins in fruit flies.

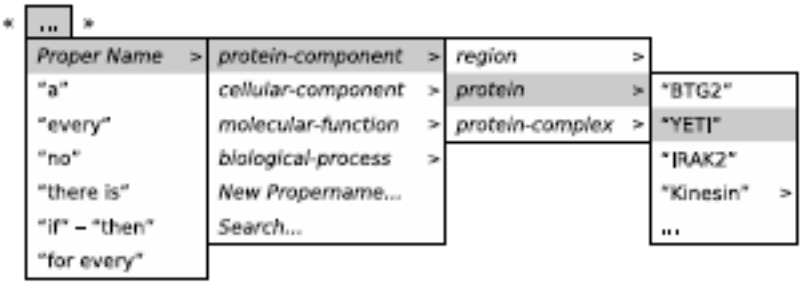
The sentences are created step by step by a simple menu.

At the beginning there is just an empty sentence that might look like this:

« ... »

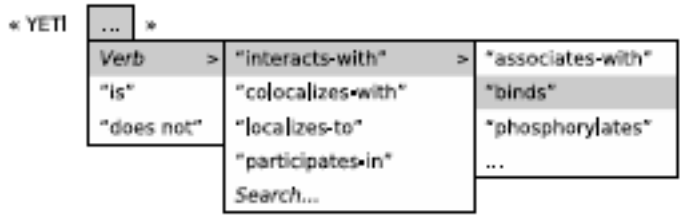
The quotes indicate the beginning and the end of the sentence and the box in the middle is used to create the content. If the user clicks on it, then a menu is displayed that shows the different options for beginning a sentence. Since we want to talk about the protein YETI we first insert the proper name 'YETI'.

This looks as follows.



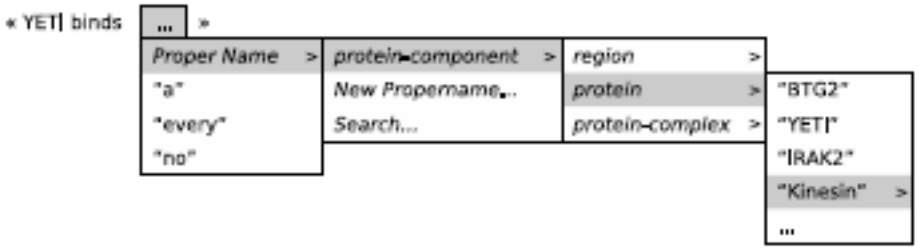
Proper names are hierarchically structured and the menu allows to navigate through this hierarchy. Alternatively, we can use the search option to find a certain term.

In the next step we get

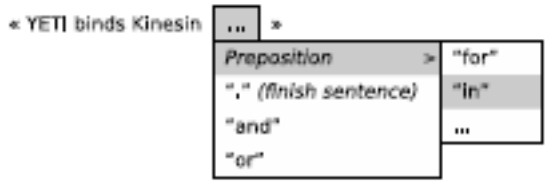


where the proper name `YETI' is now fixed as the beginning of the sentence, and we have a new menu with different entries. We want to express a *binds*-interaction, and thus we choose the verb `binds'. Like proper names, verbs are hierarchically structured and we have to navigate through this hierarchy.

In the next step we get

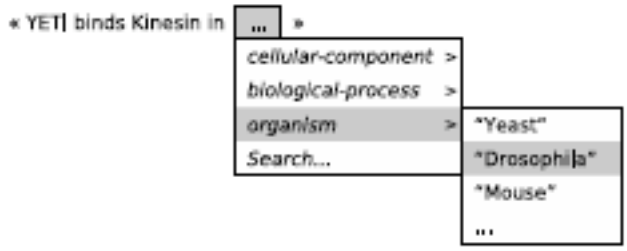


where we can define the second participant of the *binds*-interaction. We insert now the proper name 'Kinesin' and we get



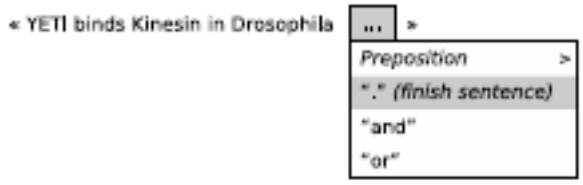
where we can finish the sentence, extend it with prepositional phrases, or we can use 'and' or 'or' to continue the sentence. Since we want to specify that the *binds*-interaction takes place in fruit flies, we have to add the preposition in'.

After that we get



where we can specify the organism. We see now, that we cannot write 'fruit fly', but we have to choose the synonymous term 'Drosophila'. In this way the author is forced to use the defined nomenclature.

Finally we get



where we finish the sentence.

For the creation of this sentence we did not need any further knowledge about ACE. Every person that is familiar with English and knows how to handle a simple menu, is able to create ACE texts with this tool.

3.8. The Benefits of our Approach

We showed in the preceding sections what we need to do for expressing scientific results about protein interactions in ACE. Now it is time to take a look at the benefits.

Today there are many databases that contain life science data, but they are mostly heterogeneous, unsynchronized, and often not up-to-date. With our approach it would be much easier to provide complete and consistent databases.

Imagine that all the scientific papers about protein interactions summarize their results in ACE. We could use these formal summaries to build up a dynamically growing knowledge base about protein interactions. Of course, we would also have to collect all the knowledge that is contained in old papers. For these, we still need some form of text mining. But once we have such a knowledge base that is continuously updated with the results of new papers, then we would be able to answer many questions. We present now some examples.

Are some results consistent with an existing knowledge base or with other papers? We can check, whether an ACE summary is consistent with an existing knowledge base. If this knowledge base contains common knowledge, then the results should be consistent, or otherwise it can be seen as an appeal against the common knowledge.

Without formal declarations, it is impossible to check a paper for consistency. Probably there exist scientific papers that contain results which are inconsistent with the common knowledge. But since this can be very difficult to find out, neither the author nor the readers might realize the special status of the results.

In the same way we can check, whether there exist papers that contradict a certain paper. That would mean that different researchers claim contradictory results. Being aware of such a contradiction might lead to a dialogue between the corresponding scientists, which might entail better and consistent results.

Are some results (or parts of them) already known? With our formal approach we can check whether a certain result, or a part of it, is already known. Results that are already considered common knowledge are usually not worth to be described as results of scientific papers (unless they contain more detailed information or if additional evidence is given). Thus it is very valuable to be able to run a check, whether a certain result is already contained in the knowledge base or not.

Furthermore a researcher might want to check, whether there exists scientific literature that has arrived at the same or similar results. Altogether our approach would help the researchers to save a lot of time, since they would not need to search ``manually'' for the relevant literature.

Is there a known answer for a certain question? If someone -- researcher or not -- has a specific question about the domain (e.g. protein interactions), then we would be able to give automatically an answer.

What is known about a certain object of interest? In some cases we do not want to ask a specific question, but we rather want to get an overview of a single object of interest (e.g. the protein IRAK2). If we ask for information about such an object then we might get something as shown in Figure 4. Such an overview could be used for a dynamic hypertext representation. This would allow us to navigate through the whole knowledge base, e.g. with an ordinary web browser. New papers that are submitted can be integrated *automatically* and thus such a web interface would be always up-to-date.

How are some objects of interest related? Instead of focusing on one single object, we might want to have an overview of the interrelations of a certain group of objects. We could extract, for example, the *interacts-with*-relations of all proteins and use this data for further examination, like the detection of clusters or hot-spots. Such examinations are already common in the research on proteins, but only with restricted data. With our approach we could consider every interaction that has been published.

3.9. Outlook

We suggest an approach of using controlled natural language for making the results of scientific papers readable and -- to some degree -- understandable by computers. But in order to achieve this goal, there is still a lot of work to do. For example, we need a authoring tool as sketched in Sect. 3.7.3,

type	IRAK2
supertypes	IRAK – Protein – Molecule
subtypes	IRAK2a, IRAK2b
interacts directly with	BTG2, XDH, Mcm3, MAX
interacts indirectly with	BCL2, Indo, Cckbr, HPCA, ID1, Ep300
phosphorylates	XDH
colocalizes with	BTG2, HPCA
localizes to	Membrane
participates in	Cell-Growth, Signal-Transduction

Fig. 4. Overview over the object 'IRAK2'. This example is purely fictitious

that would support the authors of scientific papers in the creation of ACE summaries. A prototype of such a tool does already exist. Furthermore, we need tools for the definition of terminologies and for the collection and management of knowledge.

Besides all these technical requirements, there are also political ones. There must be a commitment among the scientists of the corresponding field of research -- or at least among a large part of them -- that scientific articles get summarized in ACE. If such a summary is optional then there is little hope that it gets established.

This is the point where the publishers and editors have to come into play. The publishers would have to make ACE summaries a mandatory part of the articles, and the editors would have to check whether these summaries are correct and complete. The creation of a formal summary should be an additional requirement to consider when writing a scientific article, besides all the requirements that already exist today (e.g. about the abstract, the keyword list, and the reference list). The formal summaries can also be seen as a robust indicator for the value of a scientific paper. Information that is already known and redundant information could be ignored automatically, and wrong statements are likely to be detected at some later point in time. Thus we could use the formal summaries to quantify and qualify the contribution of a certain author, institute, or journal.

Due to the immense benefits such a system would bring along, we believe in the great potential of our approach. It could be a first step towards better communication and persistence of biomedical knowledge.

4. Extensions of the Attempto System

The language Attempto Controlled English and its associated tools were changed and extended in several directions. If necessary, the relevant documentation was updated as well.

The most important change is the simplification of the DRS representation of nouns, specifically of plurals – a feature requested by our REWERSE partners.

Here is an example. While in the original DRS representation the plural noun phrase '2 cats' is represented as

```
[A, B]
structure(A, group)
quantity(A, cardinality, count_unit, B, eq, 2)
   [D]
   part_of(D, A)
   structure(D, atomic)
   =>
   []
   object(D, cat, object)
```

in the new DRS representation we get

```
[A]
object(A, group, cat, object, cardinality, count_unit, eq, 2)
```

Though the representation is much simplified – which leads to performance gains – no information of the original representation is lost.

Other extensions of the Attempto system include:

- small clarifying changes to the language ACE 4 and to interpretation rules
- improvements and extensions of the Attempto Parsing Engine APE, preliminary OWL DL generation, new DRS representation, unknown word guessing
- a new algorithm for the resolution of anaphors
- relaxed type system for content words in the lexicon
- DRACE now supports collective plurals
- improved and faster APE web-service from a socket server, solo output
- improved functionality and design of APE web-interface
- RSS feed for news of the Attempto system

All changes are made available through the Attempto web-site.

5. Conclusions

In this report we mainly presented the translation of ACE into and from OWL DL, and the use of ACE as an ontology language.

While working on these lines of research, we noticed that ACE – though it is in general more expressive and flexible than the languages currently used or proposed for the semantic web – does not, or does not conveniently, provide some important features of web languages, for instance:

- URIs
- Unicode support
- constructs similar to OWL's allValuesFrom and oneOf
- data types similar to OWL's data type properties
- data structures similar to RDF's Bag, Alt, Seq
- procedural attachments
- built-ins like in Prolog or SWRL

We decided to extend ACE by these or equivalent features. Together with the implementation of the set of requirements requested by our REWERSE partners, namely:

- negation as failure
- prioritised rules
- decidable subsets of ACE
- modality

ACE will eventually turn into a “first-class” web language.

6. References

1. Attempto Project. Attempto website, 2006. http://www.ifi.unizh.ch/attempto.

2. Abraham Bernstein, Esther Kaufmann, Anne Göhring, and Christoph Kiefer. Querying Ontologies: A Controlled English Interface for End-users. In 4th International Semantic Web Conference, pages 112-126, November 2005.

3. Norbert E. Fuchs, Stefan Höfler, Kaarel Kaljurand, Gerold Schneider, and Uta Schwertel. Extended Discourse Representation Structures in Attempto Controlled English. Technical Report ifi -2005.08, Department of Informatics, University of Zurich, Zurich, Switzerland, 2005.

4. Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Deliverable I2-D5. Verbalising Formal Languages in Attempto Controlled English I. Technical report, REWERSE, 2005. http://rewerse.net/deliverables.html.

5. Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Attempto Controlled English Meets the Challenges of Knowledge Representation, Reasoning, Interoperability and User Interfaces. In FLAIRS 2006, 2006. To be published.

6. Daniel Hewlett, Aditya Kalyanpur, Vladamir Kovlovski, and Chris Halaschek-Wiener. Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. In End User Semantic Web Interaction Workshop (ISWC 2005), 2005.

7. Masahiro Hori, Jerôme Euzenat, and Peter F. Patel-Schneider. OWL Web OntologyLanguage XML Presentation Syntax. W3C Note 11 June 2003. Technical report, W3C, June 11th 2003. http://www.w3.org/TR/owl-xmlsyntax/.

8. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools. Edition 1.0. Technical report, University of Manchester, 2004. http://www.co-ode.org/resources/tutorials/.

9. Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004. Technical report, W3C, 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

10. Massimo Marchiori. Towards a People's Web: Metalog. Technical report, W3C, 2004.

11. Chris Mellish and Xiantang Sun. Natural Language Directed Inference in the Presentation of Ontologies. In 10th European Workshop on Natural Language Generation, Aberdeen, Scotland, August 8th{10th 2005.

12. Chris Mellish and Xiantang Sun. The Semantic Web as a Linguistic Resource. In Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Arti cial Intelligence, Peterhouse College, Cambridge, UK, December 12-14, 2005. http://www.csd.abdn.ac.uk/cmellish/papers/kbs05.pdf.

13. Peter F. Patel-Schneider. The OWL 1.1 Extension to the W3C OWL Web Ontology Language. Editor's Draft of 19 December 2005. Technical report, 2005. http://www-db.research.bell-labs.com/user/pfps/owl/overview.html.

14. Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004. Technical report, W3C, 2004. http://www.w3.org/TR/owl-semantics/.

15. Alan L. Rector, Nick Drummond, Matthew Horridge, Jeremy Rogers, Holger Knublauch, Robert Stevens, Hai Wang, and Chris Wroe. OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. In Enrico Motta, Nigel Shadbolt, Arthur Stutt, and Nicholas Gibbins, editors, Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004, volume 3257 of Lecture Notes in Computer Science, pages 63-81. Springer, October 5-8 2004.

16. Rolf Schwitter. Controlled Natural Language as Interface Language to the Semantic Web. In 2nd Indian International Conference on Artificial Intelligence (IICAI-05), Pune, India, December 20-22 2005.

17. Corinne Cayrol, Céline Cougoule, Michel Wright. The β-adaptin clathrin adaptor interacts with the mitotic checkpoint kinase BubR1. In “Biochemical and Biophysical Research Communications”, Volume 298, Issue 5, Pages 720-730. 2002.

18. Aaron M. Cohen, William R. Hersh. A survey of current work in biomedical text mining. In “Briefings in Bioinformatics”, Volume 6, Number 1, Pages 57-71. Oregon Health & Science University, Department of Medical Informatics and Clinical Epidemiology. 2004.

19. Nikolai Daraselia, Anton Yuryev, Sergei Egorov, Svetalana Novichkova, Alexander Nikitin, Ilya Mazo. Extracting human protein interactions from MEDLINE using a full-sentence parser. In “Bioinformatics'', Volume 20, Number 5, Pages 604-611. Oxford University Press. 2004.

20. Norbert E. Fuchs, Uta Schwertel, Rolf Schwitter. Attempto Controlled English -- Not Just Another Logic Specification Language. University of Zurich, Department of Informatics. 1998.

21. Norbert E. Fuchs, Stefan Hoefler, Kaarel Kaljurand, Gerold Schneider, Uta Schwertel. Discourse Representation Structures of ACE 4 Sentences. University of Zurich, Department of Informatics. 2005.

22. Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Stanford Knowledge Systems Laboratory, Palo Alto. 1993.

23. Daniele Nardi, Ronald J. Brachman. An Introduction to Description Logics. In “The Description Logic Handbook: Theory, Implementation, and Applications”. Cambridge University Press. 2003.

24. Rolf Schwitter, Anna Ljungberg, David Hood. ECOLE: A Look-ahead Editor for a Controlled Language.Centre for Language Technology, Maquarie University, Sydney. 2003.

25. John F. Sowa. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co, Pacific Grove, CA. 1999.

26. Mike Uschold, Michael Gruninger. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, Volume 11, Number 2. 1996.

27. Alexander Yeh, Alexander Morgan, Marc Colosimo, Lynette Hirschman. BioCreAtIvE Task 1A: gene mention finding evaluation. In “BMC Bioinformatics”, Volume 6. 2005.