

# Collaborative multilingual knowledge management based on controlled natural language

**Editor(s):** Roberto Garcia, Universitat de Lleida, Spain; Heiko Paulheim, University of Mannheim, Germany; Paola Di Maio, Universal Interfaces Research Lab, ISTCS, Edinburgh, UK

**Open review(s):** Prateek Jain, IBM Research, USA; Eero Hyvönen, Aalto University, Finland; Stephan G. Lukosch, Delft University of Technology, The Netherlands

Kaarel Kaljurand <sup>a,\*,\*\*</sup>, Tobias Kuhn <sup>b,a</sup> and Laura Canedo <sup>a</sup>

<sup>a</sup> *Institute of Computational Linguistics, University of Zurich, Switzerland*

*E-mail: kaljurand@gmail.com, kuhntobias@gmail.com, canedo@ifi.uzh.ch*

<sup>b</sup> *Chair of Sociology, in particular of Modeling and Simulation, ETH Zurich, Switzerland*

**Abstract.** User interfaces are a critical aspect of semantic knowledge representation systems, as users have to understand and use a formal representation language to model a particular domain of interest, which is known to be a difficult task. Things are even more challenging in a multilingual setting, where users speaking different languages have to create a multilingual ontology. To address these problems, we introduce a semantic wiki system that is based on controlled natural language to provide an intuitive yet formal interface. We use a well-defined subset of Attempto Controlled English (ACE) implemented in Grammatical Framework to automatically produce precise bidirectional translations between ACE and language fragments of a number of other natural languages, making the wiki content accessible multilingually. Because ACE has a partial but deterministic mapping to the Web Ontology Language, our wiki engine can offer automatic reasoning and question answering on the wiki content. Users speaking different languages can therefore build, query, and view a common knowledge base in their own language. We present the results of a user evaluation where participants using different languages were asked to write and assess statements about European geography. Our results show that users reach a high level of consensus, which is not negatively affected by the presence of automatic translation.

**Keywords:** semantic wiki, multilinguality, controlled natural language, Attempto Controlled English, Grammatical Framework

## 1. Introduction

The last few years have shown great progress on the technical side towards the realization of what is called the Semantic Web. Off-the-shelf triples stores, for example, can nowadays handle billions of RDF

triples. User interfaces, however, have enjoyed comparatively little attention in this research community. In 2007, Semantic Web pioneer Ora Lassila stated that “most of the remaining challenges to realize the Semantic Web vision have nothing to do with the underlying technologies involving data, ontologies, reasoning, etc. Instead, it all comes down to *user interfaces and usability*” [29]. However, this shift of focus has not yet happened, and the vast majority of Semantic Web researchers still work on technical aspects. In his keynote speech at the Extended Semantic Web Conference 2013 in Montpellier and in a subsequent blog

---

\* Corresponding author. E-mail: kaljurand@gmail.com.

\*\* The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement FP7-ICT-247914. The authors would like to thank Norbert E. Fuchs for useful comments on the draft of this paper.

post, David Karger pointed out the fact that virtually none of the 36 main-track papers at that conference presented research on end-user applications [19], the only exception being the paper on which this extended article is based [17].

The problem of Semantic Web user interfaces is even more pronounced in the multilingual case. Formal knowledge representations should ideally be usable internationally and accessible in different languages. The creation of such multilingual ontologies obviously involves users speaking different languages, who do not necessarily have a sufficient command of a given lingua franca such as English. (See [6] for a recent problem statement regarding the Multilingual Semantic Web.) To address this problem, we propose here an approach based on semantic wikis, controlled natural language, and rule-based machine translation.

Wikis are user-friendly collaborative environments for building knowledge bases in natural language. The most well-known example is Wikipedia, an encyclopedia that is being built by around 100,000 users in hundreds of different languages, and numerous other wikis exist for more specific domains. Semantic wikis [5] combine the main properties of wikis (ease of use, read-write, collaboration, and linking) with knowledge engineering technology (structured content, knowledge models in the form of ontologies, and automatic reasoning). Semantic wiki editors simultaneously work with the natural language content and its underlying formal semantics representation. The resulting wikis offer more powerful content management functions, e.g. dynamically created pages based on semantic queries and detection of semantic errors in the content, but have to somehow meet the challenge of keeping the user interface as simple as expected from wikis. The existing semantic wiki engines (e.g. Semantic Mediawiki [20] and Freebase [3]) support the inclusion of semantics in the form of RDF-like subject-predicate-object triples, e.g. typed wiki links (predicates) between two articles (the subject and the object).

Our approach to semantic wikis is different from these existing approaches by being based on controlled natural language (CNL) [38,26]. CNLs look (almost) like natural language, but are restricted for the sake of human comprehension, reliable machine translation, and/or semantic parsing. For CNLs like Attempto Controlled English (ACE) [11], the syntax is precisely defined, the sentences have a formal (executable) meaning, and they come with end-user documentation describing syntax, semantics and usage patterns. With proper editing tools, such languages support the cre-

ation of texts that are natural yet semantically precise, allowing for reliable and efficient human-computer interaction in knowledge representation tasks. CNL-based wikis — such as AceWiki [21], on which our approach is based — can offer greater semantic expressivity compared to traditional semantic wikis (e.g. OWL instead of RDF), while providing a user interface that is simpler to use.

Concretely, we present a semantic wiki system with an underlying CNL grammar implemented in a rule-based grammar formalism called Grammatical Framework (GF) [35]. This grammar describes a well-defined subset of ACE that is automatically translatable into the Web Ontology Language (OWL) [13], thereby allowing for automatic semantic reasoning over the wiki content. Additionally, the grammar facilitates a precise bidirectional automatic translation between ACE and language fragments of a number of other natural languages. Users can build, query, and browse OWL ontologies via a user-friendly multilingual interface presenting the formal content in their own natural language. As an additional feature for technical users, the underlying multilingual grammar is itself integrated into the wiki and can be collaboratively edited to extend the vocabulary and even customize the multilingual representations of ACE sentences. Our approach is implemented as an extension of the existing AceWiki system.

This work was conducted in the EU research project MOLTO<sup>1</sup>. This article is an extended version of a conference paper [17], and includes content previously reported in a project deliverable [8]. It is structured as follows: in Section 2 we review related work; in Section 3 we introduce the core features of the existing tools and technologies employed in the rest of the paper (namely ACE, GF and AceWiki); in Section 4 we describe the multilingual GF implementation of ACE; in Section 5 we discuss the novel features of AceWiki-GF, a GF-based extension of AceWiki; in Section 6 we provide an evaluation of our system; and in Section 7 we summarize our main results and outline future work.

## 2. Related work

The relevant related work includes approaches on multilingual CNLs, CNL-based wikis, multilingual

---

<sup>1</sup><http://www.molto-project.eu>

wikis, multilingual ontologies, and ontology verbalization.

Many controlled natural languages, both general purpose and domain-specific, have been developed based on different natural languages [32,26]. However, there has not been any effort to bring them under the same semantic model or to synchronize their development in a community-driven manner [30]. Our multilingual ACE grammar is an experiment in this direction, and our current implementation is partly an extension of existing work on a multilingual version of ACE implemented in GF [36]. Similar existing approaches use GF to build a bidirectional interface between a controlled fragment of Latvian and OWL, using ACE as an interlingua [14], and to verbalize business rules in a multilingual CNL [10].

The approach of a CNL-based wiki was first demonstrated by the AceWiki system, which is the basis of our work and will be discussed below. MoKi [12] is another approach of a semantic wiki engine offering a “lightly-structured access mode” for its structured content (represented in OWL). In this mode the content is displayed as an uneditable ACE text, and editing is only supported for the simpler *isA* and *partOf* statements using templates that combine CNL with HTML forms or using a native OWL syntax. In terms of multilinguality our wiki system has some similarities to an existing OWL ontology editor that allows users to view ontologies in three CNLs, two based on English and one on Chinese [2]. As the main difference compared to these approaches, our system uses the CNLs as the only user interface for both editing and viewing.

The research on GF has so far not focused on wiki-like environments. Tool support exists mostly for users constructing single sentences (not longer texts) and working alone (not in collaboration). A notable exception is an approach that investigates the use of a GF-based grammar and editor for a multilingual wiki of restaurant reviews [31]. In order to add or edit a restaurant review, users have to work on the language-independent level by manipulating abstract syntax trees, but can also simultaneously view their emerging multilingual verbalizations. In our wiki engine, we chose to hide the abstract trees, and let the users work solely on the level of natural language, via a look-ahead editor.

Even though mainstream (non-semantic) wiki engines generally allow for the wiki articles to be written in multiple languages, these different language versions exist independently of each other and only

article-level granularity is offered by the system for interlinking the multilingual content.

Ontology languages (such as RDF, OWL and SKOS) typically support language-specific annotations of ontology constructs (e.g. classes and axioms) allowing the ontological axioms to be presented multilingually. This typically involves a lot of manual work and gives no guarantees of multilingual consistency. Alternatively, one can verbalize just the entities (named classes, properties and individuals) and present the axioms using a standard OWL syntax. This however results in awkward constructs especially in languages other than English, as multilingual entity names are combined with OWL’s keywords in English (e.g. *SubClassOf*, *some*, *only*) and its fixed syntactic structure. Such ontology verbalizations are clearly insufficient and hard to read, especially for expressive ontology languages like OWL, as has been argued by others [9] and motivated them to introduce a sophisticated lexical annotation ontology to be attached to the domain ontology as linguistic knowledge.

### 3. Underlying technologies

#### 3.1. Attempto Controlled English

Attempto Controlled English (ACE) [11] is a general purpose CNL based on first-order logic. ACE can be seen as both a natural language understandable to English speakers, as well as a formal language with a precisely defined syntax and semantics understandable to automated theorem provers. ACE offers many natural language constructs, the most important of which are countable and mass nouns (e.g. ‘man’, ‘water’); proper names (‘John’); generalized quantifiers (‘at least 2’); indefinite pronouns (‘somebody’); intransitive, transitive and ditransitive verbs (‘sleep’, ‘like’, ‘give’); negation, conjunction and disjunction of noun phrases, verb phrases, relative clauses and sentences; and anaphoric references to noun phrases through definite noun phrases (‘the man’), pronouns (‘it’), and variables (‘X’). Texts built from these units are deterministically interpreted via Discourse Representation Structures (DRS) [18], which can be further mapped to formats supported by existing reasoners (e.g. OWL, SWRL, FOL, TPTP). See Figure 1 for an example of how ACE constructs are mapped to OWL.

The ACE sentence structures and their unambiguous interpretations are explained in the end-user documentation in the form of *construction* and *interpre-*

Every country that does not border a sea is a  
landlocked-country.  
(= If there is a country X and it is false that X  
borders a sea then X is a landlocked-country.)

```
SubClassOf(
  ObjectIntersectionOf(
    :country
    ObjectComplementOf(
      ObjectSomeValuesFrom(
        :border
        :sea
      )
    )
  )
  :landlocked-country
)
```

Which country is a landlocked-country?

```
ObjectIntersectionOf(
  :country
  :landlocked-country
)
```

Fig. 1. Declarative sentences in ACE map to OWL axioms, and ACE questions to OWL class expressions. ACE content words ('country', 'landlocked-country', 'border', 'sea') are directly mapped to OWL entities, while the ACE function words ('every', 'does not', ...) map to various OWL constructors, reflecting their ACE meaning and textual scope. As a general design principle, ACE offers various syntactic forms to express the same semantic form, e.g. all *every*-sentences can be paraphrased as *if-then*-sentences.

*tation* rules. The grammar of ACE and its mapping to DRS cannot be modified by the end-users, but the application-specific vocabulary can be specified as a lexicon of content words (nouns, verbs, ...) with their morphological forms and mapping to logical atoms.

While originally designed for software specifications, ACE has been developed in recent years with the languages and applications of the Semantic Web in mind. ACE fragments have been mapped to and from languages like OWL, SWRL and DL-Query [15]; ACE-based tools for building OWL ontologies have been developed, such as ACE View [16] and AceWiki; and controlled experiments have shown that ACE is a user-friendly language for specifying OWL ontologies, providing a syntax that is easier to understand and use compared to the standard OWL syntaxes [25].

### 3.2. Grammatical Framework

Grammatical Framework (GF) [35] is a functional programming language for building multilingual grammar applications. Every GF program consists of an *abstract syntax* (a set of functions and their categories) and a set of one or more *concrete syntaxes* which describe how the abstract functions and categories are

```
abstract Geography = {
  cat
  Country ;
  Relation ;
  fun
  germany : Country ;
  switzerland : Country ;
  border : Country -> Country -> Relation ;
}

concrete GeographyEng of Geography = {
  lincat
  Country = Str ;
  Relation = Str ;
  lin
  germany = "Germany" ;
  switzerland = "Switzerland" ;
  border c1 c2 = c1 ++ "borders" ++ c2 ;
}

concrete GeographyGer of Geography = {
  param
  CaseGer = Nom | Acc | Dat ;
  lincat
  Country = CaseGer => Str ;
  Relation = Str ;
  lin
  germany = table { _ => "Deutschland" } ;
  switzerland = table {
    Dat => "der Schweiz" ;
    _ => "die Schweiz" } ;
  border c1 c2 =
    c1 ! Nom ++ "grenzt an" ++ c2 ! Acc ;
}
```

Fig. 2. Example of a simple grammar that can convert the English sentence 'Germany borders Switzerland.' to the German 'Deutschland grenzt an die Schweiz.', or the other way around, via the language-neutral abstract tree `border germany switzerland`. The grammar consists of an abstract module that defines the categories and functions relevant for the domain, and a set of concrete modules, each of which defines how the categories and functions are linearized in the given language, determining language-specific details such as word order and case agreement. (Note that this is a self-contained example. In a real grammar, some of the structural complexity can be hidden behind library calls, resulting in a more elegant and maintainable code.)

linearized (turned into surface strings) in each respective concrete language. The resulting grammar describes a mapping between concrete language strings and their corresponding abstract trees (structures of function names). This mapping is bidirectional — strings can be *parsed* to trees, and trees *linearized* to strings. As an abstract syntax can have multiple corresponding concrete syntaxes, the respective languages can be automatically *translated* from one to the other by first parsing a string into a tree and then linearizing the obtained tree into a new string. (See Figure 2.)

While GF can be used to build parsers and generators for formal languages, it is optimized to handle natural language features like morphological variation, agreement, and long-distance dependencies. Additionally, the GF infrastructure provides a *re-*

source grammar library (RGL), a reusable grammar library of the main syntactic structures and morphological paradigms currently covering about 30 natural languages [34]. As the library is accessible via a language-independent API, building multilingual applications remains simple even if the programmers lack detailed knowledge of the linguistic aspects of the involved languages. These features make GF a good framework for the implementation of CNLs, especially in the multilingual setting [37]. The development of GF has focused on parsing tools, grammar editors, and extending the grammar library to new languages. The current algorithm for parsing GF grammars is based on Parallel Multiple Context-Free Grammars and allows for incremental parsing, which enables look-ahead editing [1].

### 3.3. AceWiki

AceWiki<sup>2</sup> [23] is a CNL-based semantic wiki engine, implemented in Java using the Echo Web Framework<sup>3</sup>. It uses OWL for the internal semantic representation and an OWL-compatible subset of ACE for the wiki content. An integrated OWL reasoner provides automatic consistency checking, classification, and query answering. The used ACE subset is formally defined in a grammar notation called Codeco [24]. This grammar notation is specifically designed for look-ahead features, which are required by the sentence editor to propose only syntactically legal continuations of the sentence.

The content of an AceWiki instance consists of a set of articles, each containing a sequence of entries which are either declarative sentences (corresponding to OWL axioms), questions (corresponding to OWL class expressions), or informal comments (not interpreted by AceWiki). Whenever the wiki content is changed, the integrated reasoner determines the effect of the change, flags inconsistencies, and updates the dynamically generated parts of the wiki such as concept hierarchies and answers to questions.

Content words (proper names, nouns, transitive verbs, relational nouns and transitive adjectives) in the wiki sentences map to OWL entities: proper names represent OWL individuals, nouns are OWL classes, and the relational words correspond to OWL properties. Each content word gets its own wiki article.

```
if_thenS
(vpS
 (termNP X_Var) (v2VP contain_V2 (termNP Y_Var)))
(neg_vpS
 (termNP Y_Var) (v2VP contain_V2 (termNP X_Var)))

ACE:      if X contains Y then Y does not contain X
Dutch:   als X Y bevat , dan bevat Y niet X
Finnish: jos X sisältää Y:n niin Y ei sisällä X:ää
German:  wenn X Y enthält , dann enthält Y X nicht
Spanish: si X contiene Y entonces Y no contiene X
```

Fig. 4. Abstract tree and its linearizations into five languages which express the OWL asymmetric property axiom, which is assigned by the ACE-to-OWL mapping to the ACE sentence. The linearizations feature different word orders depending on the language. The tree abstracts away from linguistic features like word order, case, and gender, although it still operates with syntactic notions such as *negated verb phrase*.

## 4. ACE-in-GF

ACE-in-GF (for a detailed description see [7] and the project website<sup>4</sup>) is a GF implementation of a large subset of ACE, which corresponds roughly to the subset supported by AceWiki and to the input language of the ACE-to-OWL translator [15]. The implementation of ACE-in-GF relies heavily on the RGL by importing most of the language structures of ACE from the GF resources for standard English via the language-independent API of RGL (Figure 3). This language-neutral approach allows us to easily plug in other RGL-implemented languages thus creating a multilingual grammar which bidirectionally maps ACE to fragments of other natural languages (Figures 4 and 5). The result is an almost automatic definition of multiple CNLs based on the RGL languages. The RGL-based design guarantees that the quality and language-coverage of the grammar increases with the increasing quality and coverage of the RGL. Still, some fine-tuning is necessary to override certain grammatical structures that cannot be specified in a completely language-independent way. The current grammar supports almost all the RGL languages. However, for some languages, certain ACE constructs (e.g. verb phrase coordination and some complex types of questions) have not yet been implemented since they are not directly available via the RGL. As ACE supports a certain degree of syntactic sugar, the incomplete syntactic coverage does not necessarily reduce the semantic expressiveness of the grammar.

In order to use the grammar in domain-specific applications, one needs to supply a multilingual lexicon

<sup>2</sup><http://attempto.ifi.uzh.ch/acewiki/>

<sup>3</sup><http://echo.nextapp.com/>

<sup>4</sup><https://github.com/Attempto/ACE-in-GF>

```

incomplete concrete AttemptoI of Attempto = open Syntax, LexAttempto, ... in {
  -- ACE noun phrase uses the RGL noun phrase structure
  lincat NP = Syntax.NP ;

  ...

  -- noun phrase with the determiner 'every' e.g. 'every country'
  lin everyNP = Syntax.mkNP every_Det ;

  -- verb phrase with a passive transitive verb and a noun phrase
  -- e.g. 'bordered by Germany'
  lin v2_byVP v2 np = mkVP (passiveVP v2) (Syntax.mkAdv by8agent_Prep np) ;

  ...
}

```

Fig. 3. Fragment of a GF grammar for ACE listing the linearization rules for the functions `everyNP` and `v2_byVP`. There are around 100 such rules. This incomplete concrete GF module (functor) implements the ACE sentence structures using the operators of RGL's `Syntax` module (e.g. `every_Det`, `mkVP`). A concrete language implementation parametrizes this functor with a concrete language resource (English in case of ACE) and possibly overrides some of the rules with language-specific structures. For the function categories, the grammar uses categories that are also used in the ACE user-level documentation, e.g. noun (N), transitive verb (V2), noun phrase (NP), relative clause.

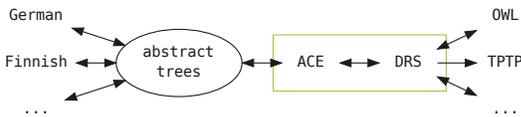


Fig. 5. Bidirectional mapping between a formal language like OWL and a natural language like Finnish facilitated by the multilingual GF implementation of ACE and various mappings between ACE and other formal languages.

where words are classified according to ACE content word classes (common nouns, proper names, transitive verbs, transitive adjectives), and come with the morphological forms required by the grammar. Different languages have different morphological complexity, e.g. while ACE operates with two noun forms (singular and plural) and three verb forms (infinitive, third person singular and past participle), other languages (e.g. Finnish) need many more forms to be used in the various ACE sentence structures. The implementation of this lexicon can rely on RGL's smart paradigm operators [34], which automatically derive the complete morphological features of a word from a few input arguments, e.g. the dictionary form and the gender information.

The ACE syntax is designed to be unambiguous, meaning that every sentence generates just a single abstract tree. For the other implemented languages, ambiguity is reduced to a minimal level, but not completely eliminated. This has practical reasons: ACE has been carefully designed to contain no ambiguity at all, but we cannot expect that this automatically leads to a one-to-one mapping to unambiguous subsets of other languages. Therefore, our approach is to let users work with representations that might be slightly ambiguous

in certain cases, giving them the opportunity to explore and resolve the ambiguity in languages where it is visible.

The ACE concrete syntax of the grammar is continuously tested against the existing AceWiki grammar (implemented in Codeco), and has near-perfect properties with respect to coverage and ambiguity. The grammar, however, overgenerates with respect to DRS-style anaphoric references, e.g. allowing the user to introduce a definite noun phrase where there is no corresponding antecedent. Such restrictions are difficult to implement in GF but easy in Codeco (because Codeco was designed for exactly this purpose).

We evaluated the multilingual aspects of the grammar by having native speakers look at the multilingual representations of a set of automatically generated sentences (see [8]; we will not go into the details of this evaluation here). The multilingual representations were found to correspond well to their ACE counterparts. The evaluation subjects mostly flagged the translations for stylistic reasons (too much verbosity, etc.) which is not unexpected for a CNL.

## 5. AceWiki-GF

AceWiki-GF is the result of combining AceWiki and GF with the goal to create a CNL-based multilingual semantic wiki engine. This required the following modifications to the existing (monolingual) AceWiki engine:

- the Codeco grammar and parser for ACE was replaced by the GF-implemented multilingual grammar and a GF parser;

- the English-specific lexicon editor was replaced by a simple GF source editor which can be used to edit GF grammar modules, including lexicon modules;
- instead of concrete ACE sentences, a wiki article consists of a sequence of abstract tree sets representing sentences in a language-neutral form;
- new types of wiki pages were introduced: “free” pages that are not linked to an ontological entity and “technical” pages for grammar modules.

The existing user interface has largely been preserved; the main additions are the disambiguation dialog and a menu for setting the content language, which also determines the user interface language (Figure 6). The wiki still follows the principle of hiding the internal formal representations. Users are supposed to interact with the system only via natural language sentences. (Interested users can still look at the parser output, which provides information on syntax trees, translation alignment diagrams, and more.)

### 5.1. Structure and linking

In general, AceWiki-GF follows the existing AceWiki structure: a wiki is a set of articles, each containing a sequence of sentences. New is the fact that also the grammar definition is part of the wiki and can be referenced from the articles using wiki links.

A GF grammar is structured in a way that is naturally representable as a set of wiki articles. Each grammar module can be stored as a wiki article and linked to the modules that it imports. Furthermore, grammar modules have internal structure — sets of categories and functions (which reference categories) — which can be linked to wiki content because the content is represented as a set of trees (i.e. structures of function names). One of the benefits of having a grammar definition as part of the wiki is that it provides an integrated documentation of the language that the wiki users are required to use. Note that the full grammar also contains modules which are part of the general RGL and thus not editable and also not part of the wiki. This resource can be made accessible via external links to the online RGL documentation.

### 5.2. Sentence editing and ambiguity

AceWiki-GF uses the regular sentence editor of AceWiki, with its look-ahead feature for displaying all words that can syntactically follow a partially com-

pleted sentence. The language of the sentence, of course, depends on the chosen wiki language. In case an entry is ambiguous, i.e. parsing the sentence results in multiple abstract trees, the sentence is stored in this ambiguous representation (ambiguity does not exist in ACE and only to a very restricted degree in the other languages). In other languages, such ambiguous sentences may lead to more than one linearization, in which case the ambiguity can be resolved by wiki users in this language. At the moment, disambiguation cannot be done in the language where an ambiguous sentence was written, but this could be supported by the implementation of “disambiguation syntaxes” [37]. They would override the linearizations of the ambiguous constructs, phrasing them in an unambiguous, although possibly more formal-looking way, which could be used to distinguish the different readings in the case of ambiguity.

### 5.3. Lexicon and grammar editing

Our wiki makes the grammar available as a set of interlinked grammar modules falling into the following categories:

- ACE resource grammar (about 30 modules which are typically identical to their English resource grammar counterparts, sometimes overriding certain structures);
- ACE application grammar, reflecting the OWL-compatible subset of ACE (one module);
- instantiation of this grammar for each supported language with additional modules that describe language-specific overriding of some of the functions;
- content word lexicon module(s) for each language.

In order to add a new word to the wiki, a line needs to be added to the lexicon wiki page, i.e. the page that corresponds to the lexicon module. Although editing the lexicon technically means editing the GF grammar, the lexicon module is conceptually much simpler than the general grammar module. The structure of lexicons in all the supported languages is roughly the same even if some languages are morphologically more complex (e.g. have more case endings). The language-specific lexical structures are hidden from the user behind language-neutral categories like N and V2 and constructed by functions like mkN and mkV2 which are capable of determining the full word paradigm on the basis of only one or two input forms. Thus, support for

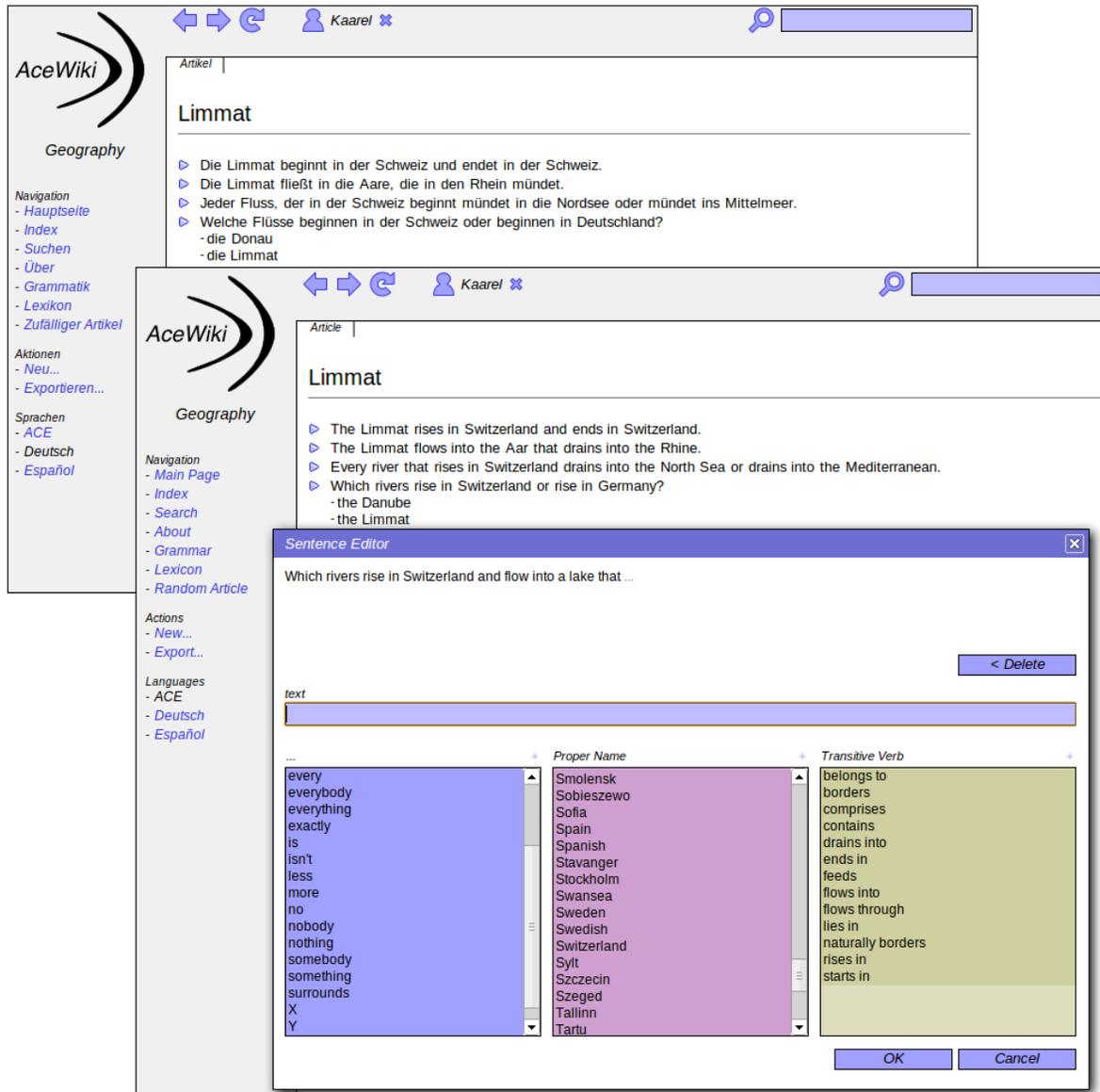


Fig. 6. Multilingual geography article displayed in ACE and German. The wiki language (of both the content and the user interface) can be changed in the left sidebar. Otherwise the user interface is the same as in AceWiki, with the look-ahead editor that helps to input syntactically controlled sentences, in this case offering proper names and common nouns as possible continuations.

multilinguality does not increase the conceptual complexity of the wiki. (See Table 1.)

Wiki users experienced in GF are also able to modify the full grammar, although we do not see many compelling use cases for that as ACE itself is pre-defined and thus changing its grammar should not be allowed (e.g. it would break the functioning of the mapping to OWL). Its verbalization to other languages, however, is sometimes a matter of taste, and could be therefore made changeable by the wiki users,

e.g. users can add an alternative formulation of an ACE sentence in some language by using a GF variant. Also, the possibility to define arbitrary GF operators can make certain lexicon entry tasks more convenient.

A change to the underlying grammar (even if only in the lexicon module) can have the following consequences for the content: (1) removing a function can render some of the wiki entries (the ones whose trees use this function) invalid, the user must then reformulate the respective sentences to conform to the

Table 1

Entries in the multilingual lexicon for the function ‘country’. Smart paradigms like mkN are used to create the internal structure of the entry. In many cases giving only the lemma form to the word class operator is sufficient to get a correct internal structure. In some cases further forms or information about gender (in some languages) need to be added. This makes the user interface to the lexicon relatively simple and homogeneous across languages. At the grammar source level, the lexicon for each language is in a separate module. At the user interface level it can be presented as a table where rows correspond to lexical functions and columns to the language modules.

| Language | Operator call                           |
|----------|---|
| Danish   | mkN "land" "landet"                     |
| Dutch    | mkN "land" neuter                       |
| Finnish  | mkN "maa"                               |
| French   | mkN "pays" masculine                    |
| German   | mkN "Land" "Länder" neuter              |
| Italian  | mkN "paese"                             |
| Swedish  | mkN "land" "landet" "länder" "länderna" |

new grammar/lexicon; (2) altering the linearization of a function might cause some sentences to become unparseable or ambiguous in the corresponding language. This does not have an immediate effect on the stored wiki content because the storage is based on trees, but if an existing sentence is submitted again to the parser then it might fail or result in more trees than before. A general change to a grammar module (e.g. removing a category) can also make the whole grammar invalid, which obviously should be avoided.

#### 5.4. Underlying semantic representation

Just as in AceWiki, each AceWiki-GF entry has a corresponding OWL representation, in this case obtained by linearizing the abstract tree of the entry (using the ACE concrete syntax) into an ACE representation, and then mapping it to an OWL class expression (DL-Query), if it is an ACE question, or to an OWL axiom, otherwise. The entity IRIs in the resulting OWL representation are derived directly from the lemma and word class information of the ACE content words.

When a wiki entry is added or edited, its corresponding OWL representation is integrated into the underlying knowledge base and the OWL reasoner is consulted to check its consistency and update the answers to the questions in the wiki. In case a new declarative entry introduces inconsistency, it is disabled and does not participate in future reasoning. The user interface of the wiki displays such entries in red color. Each en-

try also comes with a ‘‘Assert/Retract’’ flag that lets the users enable (if consistent) or disable the entries manually.

Ambiguous wiki entries map, in general, to more than one OWL axiom. There are various possible strategies for handling such cases, and we have not committed ourselves to a particular one. For example, one could arbitrarily choose a single axiom (such as the first one), or alternatively insert all of them combined in a disjunction. In any case, users should be given the opportunity to resolve these ambiguities.

#### 5.5. Multilinguality

Users can change the language of the wiki at any time via a language selection menu in the left sidebar. This changes the language of the wiki content, the user interface labels, and the sentence editor. The wiki content and the sentence editor get these multilinguality capabilities from the underlying multilingual ACE grammar (discussed in Section 4). Some minor aspects of the wiki environment currently remain monolingual, namely titles of articles, the free-form (not machine-interpreted) comments, and comments in the grammar modules. We plan to improve this by including titles to the grammar, and letting users manually translate free-form comments.

The multilingual setting leads to a third type of user in addition to wiki readers and wiki editors: wiki translators. Their main task is to translate the existing words into other languages. Concretely, they have to reference the correct operators in the RGL morphological API and to check if the automatically generated translations are accurate with respect to ACE. AceWiki-GF is designed to be read and edited by users without technical knowledge, but translation requires (at least at the moment) some specific knowledge of the used controlled language and the RGL morphology API.

#### 5.6. Implementation

AceWiki-GF has been implemented as an extension of the AceWiki code base, and uses several existing tools and libraries. The GF Cloud Service (see [4] and the project website<sup>5</sup>) provides linearization and parsing (and the related look-ahead) services for GF grammars. It also provides basic support for storing, updating and recompiling the grammar files that make up the

<sup>5</sup><http://cloud.grammaticalframework.org/gf-cloud-api.html>

grammar. The ACE parser APE<sup>6</sup> provides the mapping of ACE sentences to the OWL form (as is the case also for the monolingual AceWiki). The current implementation of AceWiki-GF is available on GitHub<sup>7</sup> and can be used via some demo wikis<sup>8</sup>.

## 6. Evaluation

In previous work, two usability experiments have been performed on AceWiki with altogether 26 participants [22]. The results showed that AceWiki and its editor component are easy to learn and use. Another study confirmed that writing CNL sentences with the editor is easier and faster than writing other formal languages [28]. In terms of understandability, it has also been demonstrated that ACE is more effective than the OWL Manchester Syntax [25] and than other formal languages used to query and document source code structure [27]. Based on these previous results, we present here a user experiment that focuses on the new multilinguality features of AceWiki-GF.

With the evaluation presented here, we want to assess the quality of interlingual communication via multilingual wiki articles. This is our hypothesis:

*A group of users reaches almost the same level of agreement on the content of an article presented to them in different languages as when the article is presented to all of them in the same language.*

Concretely, we measured the degree to which users agree on the truth or falsehood of sentences written in AceWiki-GF. The following sections describe in detail the steps of the evaluation.

### 6.1. Design

The experiment was designed to evaluate the effectiveness, efficiency and usability of AceWiki-GF when users of different languages are collaboratively working on the same content. This includes the tasks of writing sentences about a certain topic in the user's own language, reading and understanding sentences written by other users, and correcting mistakes made by other users in the form of identifying and deleting false sentences. When users work in different languages, the quality of the automatic translation is of

course an important factor for the effectiveness of their communication. To determine whether two users understand each other, we asked them to write true and false sentences, specifying which sentences they considered to be true and which ones they considered to be false. This gave us a simple yet robust measure of the degree to which users can reach consensus, within and across languages. For example, if a sentence such as 'Every river flows into a lake or flows into a sea or flows into a river.' is perceived to be true by an English speaker, then after being automatically translated into another language, it should also be perceived as true by the speakers of the second language. The cross-language understanding can then be evaluated by measuring the agreement on sentences being true or false. The experiment thus consisted of two tasks: In the main editing task, participants created articles in their native language or in a language they were fluent in. In the post-editing task, the participants read in their respective language automatically translated articles written by other users and evaluated the truth or falsehood of the sentences.

In order to have a baseline for comparison, every participant accessed two articles during the post-editing task: one that had been translated from another language, and one that was originally written in the same language, i.e. has not undergone any translation process. Even in the latter case, we cannot expect perfect agreement, as mistakes can never be completely avoided and, more importantly, people have different views on the world and tend to interpret certain sentences differently. The above exemplary sentence on rivers probably appears true on first sight to most readers, but it is false if one considers "endorheic basins", which are fed by rivers but are neither lakes nor seas nor rivers.

We developed a domain vocabulary of around 500 words covering the main topics of European geography: countries, their capitals, languages, and major natural features (rivers, mountains, etc.). This domain was chosen because it was likely to be equally known to all possible participants. Also, individuals, concepts and relations in this domain can be effectively illustrated in the form of a map, a medium also likely to be familiar to everybody. The vocabulary was developed for English, German and Spanish, and was designed to contain no lexical ambiguity, i.e. each word in the lexicon corresponds to exactly one lexical entry in all languages. The small number of languages, compared to what ACE-in-GF can potentially offer, was mainly due to the fact that building large multilingual vocabularies

<sup>6</sup><http://github.com/Attempto/APE>

<sup>7</sup><http://github.com/AceWiki/AceWiki>

<sup>8</sup><http://attempto.ifi.uzh.ch/acewiki-gf/>

is very time consuming. Furthermore, we had to make sure that we are able to recruit a sufficient number of participants for each of the chosen languages. Yet, we believe that results can be reliably extrapolated to other languages for our specific scenario.

Participants were asked to enter both simple, existentially quantified sentences and complex, universally quantified sentences. Examples of the first type are ‘The Limmat flows through Zurich.’ and ‘Zurich is not a capital of Switzerland.’. These are sentences that mention specific domain objects like ‘Zurich’ or ‘Limmat’, and can in general easily be extracted from a map. Examples of the second type are ‘Every country that does not border a sea is a landlocked country.’ and ‘If a river X flows into a river Y then Y does not flow into X.’. Such universal statements express general knowledge, referring to concepts like ‘country’ or ‘river’. We were interested in both types of statements for our evaluation.

Participants for the experiment were recruited via university mailing lists from different European countries and by word-of-mouth spreading. Participants were only required to have a good command of one of the three languages involved in the evaluation — English, German or Spanish — and to be somewhat familiar with computers so that they could easily get the gist of the wiki and of using an editor to write controlled languages. The communication with the evaluation participants occurred via email and the experiment was performed via a dedicated AceWiki-GF instance that was available online, called ‘AceWiki Geography’. There were no strict time limits given to the participants.

The wiki was presented to each participant as a monolingual environment, i.e. participants could see the wiki only in their own language. This was to ensure that the GF-based automatic translation was the only way to communicate across languages. The participants were working in a localized version of the wiki, i.e. not just the wiki content but the entire graphical user interface was shown in their language. Finally, we designed a questionnaire to gather feedback from the participants.

### 6.1.1. Setup

In order to obtain clearly interpretable evaluation results, the AceWiki-GF environment was customized and several features were disabled:

1. only the grammar-backed content was allowed in the wiki, i.e. it was not possible to enter free-form comments;

2. the grammar could not be changed, i.e. users had to work with the available vocabulary and available syntactic structures;
3. the wiki language could not be changed, i.e. users could read and edit the wiki in only one language and they could not observe how their contribution was interpreted in other natural and formal languages such as OWL or GF tree structures;
4. the OWL reasoning capability of the wiki was disabled;
5. users had to register and log in so that everybody’s contribution could be identified and tracked;
6. direct collaboration with other users was forbidden (participants were instructed to edit only their own pages);
7. using the “Assert/Retract” flag in AceWiki-GF, each sentence could be tagged as “true” or “false”, which renders the sentences in black or red, respectively;
8. the wiki was configured to automatically disambiguate structurally ambiguous entries by always picking the first reading. The ACE-in-GF grammar allows by design for very little structural ambiguity which made it unlikely that many users would be exposed to the wiki disambiguation dialogue, so we decided to exclude this feature from our evaluation. As explained above, also lexical ambiguity was avoided.

In addition, the coverage of the ACE-in-GF grammar was somewhat reduced for the evaluation purposes:

1. only single-sentence statements were allowed;
2. question sentences were disabled because we wanted to base the evaluation on the agreement over the truth value of the sentences (which questions do not have);
3. animate indefinite pronouns (‘everybody’, ‘nobody’, ‘somebody’) were removed as these do not fit the inanimate domain of geography;
4. the quantifier ‘for every’ and the reflexive construct ‘itself’ were removed as their implementation contained bugs in German and Spanish.

The grammar was fixed prior to the evaluation and was not changed during the evaluation.

### 6.1.2. Procedure

Participants were divided into three language groups of equal size: English, German and Spanish. The procedure for the participants consisted of the following four steps:

1. Introduction
2. Main editing task
3. Questionnaire
4. Post-editing task

In the introduction step, the participants were given a brief description of the experiment, including an overview of the sequence of tasks they would have to carry out. The description of the main editing task and a link to the questionnaire were provided to the participants in the introduction, but the detailed description of the post-editing task was only provided after the participants had filled out the questionnaire, i.e. when we were sure they had finished the main editing task. The introduction was also meant to familiarize the participants — most of whom we expected to have no prior knowledge of ACE nor AceWiki — with the concept of controlled natural languages and the AceWiki-GF environment, to the extent that they would be able to successfully participate in the evaluation. The participants were asked to watch a screencast of six minutes on the AceWiki-GF environment which contained instructions for the registration and login procedure, basic article creation and sentence editing. Also the domain of the wiki — European geography — was introduced, pointing out some existing sources that describe this domain in the form of maps and encyclopedic articles.

The main editing task asked the participants to create a new wiki page and write at least four true sentences, for example ‘Every river that rises in Switzerland drains into the North Sea or drains into the Mediterranean.’, and at least four false sentences, for example ‘Every country that borders Germany is a member of the European Union.’, and to tag them explicitly as “true” or “false” using the “Assert/Retract” flag. They were additionally encouraged to use some complex structures and expressions, and they were allowed to edit and delete their sentences as often as they wished during the period of the experiment. In general, they were given freedom with regard to how many sentences to write or which words to use in their sentences, with the restriction that they could not create new words.

Once the main editing task was completed, the participants were presented with a questionnaire asking about their background and their impressions of the system and of the constraints enforced by the controlled environment. Specific questions were asked about the user-friendliness of the look-ahead editor and the complexity of the sentence formation task.

For the final post-editing task, the participants were directed to two articles written by other participants and were asked to remove all false sentences in these articles. They were instructed to ignore minor syntactic errors possibly present in the sentences and to only remove a sentence if it was clearly false in terms of its meaning. Both articles were copies of articles that other participants had produced, with a randomized order of the sentences and with their true/false color-coding removed. For each participant, one of the articles was originally written in their own language and the other was translated from another language. The participants did not know which was which (in fact, they did not even know that there was such a difference). Half of the participants did the post-editing on the translated article first; the other half started with the article that did not involve translation. In this way, each article of the main task was post-edited exactly twice: once with and once without translation. The distribution of the six possible translation directions between the given three languages was perfectly balanced.

## 6.2. Results

The results of the experiment were collected in the form of the final wiki content, system logs (which also registered edits and deletions of sentences that otherwise would not end up in the wiki content), and feedback via the questionnaire. We analyzed these data in a number of different ways.

### 6.2.1. General numbers

We had ten participants for each of the three languages, i.e. 30 participants in total. Most of them had a linguistics or computer science background, and the ratio of female to male participants was 14 to 16. They spent on average 37 minutes using AceWiki-GF, creating in total 316 sentences — not counting two sentences that led to an internal error. Of these, 171 sentences were marked as true and 145 as false. Therefore, each participant wrote on average 5.7 sentences marked as true and 4.8 sentences marked as false.

### 6.2.2. User agreement

The degree of agreement between participants from the same and different languages was the main outcome of this experiment. To get a feeling for why participants might disagree on the truth or falsehood of a statement even in the absence of translation, let us have a look at an example. One of the participants wrote (in German) ‘Every mountain contains a valley.’ and marked it as true. Both post-editing participants (ac-

cessing it in German and English, respectively), however, deleted this sentence, thereby stating that they thought it to be false. Apparently, the participants had different views of the world and interpreted at least some of the terms ‘mountain’, ‘contains’, and ‘valley’ differently. The first participant might have thought that every mountain must have a certain size (otherwise it would just be a hill), is a geographical entity that is subject to rain and erosion, and therefore must have some trenches that, in this participant’s view, are to be called valleys. The other participants, in contrast, might have followed the line of argument that there is no logical necessity for a mountain to have valleys, or that small mountains only have trenches that are not big enough to be called valleys, or that the verb ‘contain’ does not apply to mountains and their valleys. In any case, this example shows that this kind of disagreement can easily arise without any translation process involved.

Coming back to our hypothesis, we wanted to show that translation does not significantly lower the level of agreement. The agreement level can be defined as  $(T_k + F_d)/S$ , where  $S$  is the total number of sentences,  $T_k$  is the number of sentences originally marked as true and not deleted in the post-editing, and  $F_d$  is the number of sentences originally marked as false and deleted in the post-editing. Figure 7 shows the achieved agreement levels. In the case where the same language was used during the main task and post-editing — meaning that there was no translation — the average agreement level was 82.2%. This means that the respective participants disagreed on 17.8% of the sentences with respect to whether they were true or false. In the case where post-editing was performed in a different language — meaning that there was a translation process — the average agreement level was 84.0%. That means that the agreement level was even slightly higher with translation, but we have every reason to assume that this is just a statistical artifact, as the difference is not significant at all: We get a  $p$ -value of 0.87 when applying a Wilcoxon signed rank test to compare the two samples. The null hypothesis that the two samples come from identical distributions cannot be refuted. Therefore, the level of agreement is about the same in our two samples, which is consistent with our hypothesis.

The above calculation, however, does not prove that there is no difference between the two scenarios: It only shows that we have so far no reason to assume that they are different. In fact, no statistical test can prove that two samples come from the same distribu-

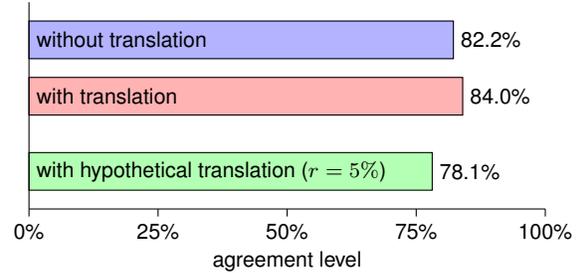


Fig. 7. The agreement between the participants of the experiment with respect to the truth or falsehood of sentences was not significantly different when translation was involved. ‘‘Hypothetical translation’’, assuming a constant translation error rate  $r = 0.05$ , shows a significantly lower agreement level than for the actual translation.

tion, as no sample size can eliminate the possibility of an arbitrarily small difference of the distributions. For that reason, we need to come up with a more specific hypothesis that is testable. First, let us assume that translation introduces a constant translation error rate  $r$  with  $0 < r < 1$  that has the effect that the level of agreement between two given users is  $(1 - r) \times a$  if translation is involved, where  $a$  would have been the level of agreement under the same circumstances if no translation was necessary. With this definition we can state a more specific hypothesis:

*The translation error rate for AceWiki-GF is less than 5%.*

Under these assumptions, we can use our sample that did not involve translation and calculate the agreement level it would have produced with a translation error rate of 5%. We can call this a ‘‘hypothetical translation’’ and Figure 7 shows that its agreement level is 78.1% ( $= 0.95 \times 82.2\%$ ). We can now compare this to the original sample that involved translation, and we can evaluate with a one-tailed test whether we can refute the null hypothesis that the translation error rate is 5% or more. Doing this with a one-tailed Wilcoxon signed rank test gives us a  $p$ -value of 0.046. This means that we can reject the null hypothesis, which verifies our hypothesis that the translation error rate of AceWiki-GF is less than 5%.

Whether a sentence was initially marked as true or false did not have a large effect on the level of agreement. Sentences originally marked as true were kept in the post-editing phase in 81.8% (without translation) and 82.7% (with translation) of the cases; sentences originally marked as false were deleted in 84.1%

(without translation) and 84.7% (with translation) of the cases.

### 6.2.3. Syntactic features

We were interested in the syntactic features of the 316 original entries, specifically the frequency distribution of their words and grammatical constructs and whether the distributions differed depending on the language and the true/false tag. Our main concern was that certain grammatical constructs are not easily usable or easily discoverable in German and Spanish, compared to ACE. This is a reasonable hypothesis because ACE has been carefully designed over many years, while its German and Spanish counterparts were obtained almost automatically and little effort has so far gone into fine-tuning them.

This analysis was done in a completely language-neutral way by looking at the underlying tree of every entry. This is possible because the functions in the ACE-in-GF abstract syntax are relatively syntactic in nature, referring to syntactic objects like “relative clause” and “noun phrase”, rather than semantic objects like “relation” or “concept”. In other words, this means that if an ACE user chose to express a sentence with a relative clause then the readers in German and Spanish would also see a relative clause, just in their respective languages. In a more semantic grammar such syntax-based analysis of the accessibility of the language features would not work, and one would have to look directly at the logical form as we do in the next section.

There were 315 unique trees, i.e. only one tree was repeated, which happened in the same language, namely German. It turned out that almost all non-lexical functions of the grammar were used by the participants. The exceptions were “variable in apposition”, e.g. ‘a country X’, and negated object relative clause, e.g. ‘that a country does not border’. This is not surprising as these constructs are conceptually relatively complex, and can be rephrased by simpler equivalent constructs.

Looking at the most frequently used constructs, no unusual distribution of grammatical constructs per language was observed, apart from two cases:

1. The users of German avoided relative clauses much more than the users of ACE and Spanish. This can be explained by looking at the orthographic conventions of German relative clauses, how they were implemented in the German grammar, and how the look-ahead editor presented them. Relative clauses are always

surrounded by commas in German, the grammar however required the omission of the final comma due to a bug in the RGL. Also, as the look-ahead editor always presents single follow-up tokens, users could only see a comma as an indicator of an available relative clause. In this case, however, it would be more user-friendly to show more of the possible follow-up context, i.e. a comma followed by the possible relative clause pronouns. These discrepancies might explain the low usage of relative clauses in German.

2. The sentences of the ACE users contained very few definite noun phrases (‘the country’, ‘the capital of’). This is probably due to a bug which caused the look-ahead editor to classify the token ‘the’ as a proper name (because it also occurred as part of proper names such as ‘the EU’). This misclassification made this token less discoverable in the look-ahead editor and the users who relied on the clicking on the category boxes for the construction of sentences were less likely to use definite noun phrases.

When looking only at the sentences marked as “true”, we did not observe a different distribution. This means that the sentences marked as “false” were also syntactically normal, i.e. users did not generate false sentences by just randomly clicking on the word selection lists of the look-ahead editor.

We conclude that the use of the different languages did not result in widely different sentence patterns, and that the possible deviations could be effectively discovered by the wiki developers using the described comparison of construct distributions in different languages. The issues that we noticed can be easily fixed by modeling the German orthographic conventions more closely and by generalizing the look-ahead editor to propose more context.

From the 500 entries in the lexicon, the participants used 230 entries. The usage distribution of the content word vocabulary differed across languages, but in an unsurprising way — German-speaking users wrote more about Switzerland and Germany, while the Spanish-speaking users more about Spain.

### 6.2.4. Semantic features and mapping to OWL

We analyzed the semantic properties of the resulting wiki entries by looking at their OWL mapping (in a few cases it resulted in a SWRL rule, which we will consider here, for simplicity, a special kind of OWL axiom). We were interested in the types of axioms that resulted and the cases where the mapping to OWL

failed, as this reflects the categories and complexity of the wiki entries.

From the total of 316 wiki entries, four were not ACE-compatible, i.e. they were — due to some bugs — incorrectly parsed or generated by the ACE-in-GF grammar. There were 26 entries (including the four non-ACE entries) which could not be mapped to OWL, i.e. the grammar does not always correctly model the OWL-compatible subset of ACE. The main reasons include: sentence disjunction which cannot be directly represented in OWL as it does not support axiom disjunction; cardinality constraints together with relational nouns, some forms of which the ACE-to-OWL translator does not support; and odd sentences with respect to anaphoric references, e.g. the ‘then’ part of the ‘if-then’ sentence does not anaphorically reference the ‘if’ part (which in general cannot be represented in OWL). AceWiki-GF as well as the original AceWiki is designed to handle such OWL mapping deficiencies by highlighting the failing entries and pinpointing the source of the failure by an error message, so that the participants can rephrase the sentence, often in a semantically equivalent way, or otherwise delete them. From the entries that were marked as “true” only one entry failed to map to ACE and 11 entries failed to map to OWL. Table 2 shows the frequency distribution of OWL axiom types. 32% of the axioms correspond to the universal statements that the evaluation participants were encouraged to also enter.

Loading the OWL-compliant sentences that were marked as “true” into the wiki with reasoning turned on (which was turned off during the experiment), the knowledge base was found to be consistent, meaning that these sentences did not contain a contradiction. However, the wiki also found no inconsistencies if the complete set of OWL-compliant sentences were loaded. This is not unexpected as the set of sentences was small and users were not coordinated to edit the same articles and write about the same objects thus there was little chance that they would write contradictory statements, especially considering that with the OWL-based open-world reasoning (even with Unique Name Assumption which the wiki enforces) some types of inconsistencies are not immediately captured.

#### 6.2.5. User feedback in the questionnaire

Apart from providing information on their background, the participants had to answer the following three questions in the questionnaire:

1. Was AceWiki Geography easy or difficult to use in general?
2. Was the sentence editor easy or difficult to use?
3. Was creating true and false statements easy or difficult to perform?

This gave us a quantitative measure of the participants’ subjective experience with the wiki. For each of these questions, participants could choose from “very difficult” (value 0), “difficult” (1), “medium” (2), “easy” (3), and “very easy” (4). For all three questions, the average answer was close to but slightly below “easy”: 2.93 for the first question, 2.77 for the second, and 2.70 for the third. Given that it was indeed a rather complicated task involving a powerful tool, we consider these results very satisfactory.

Participants could also give free-form feedback in the questionnaire. Unsurprisingly, around 80% of the participants reported that the controlled environment did not let them express everything that they had in mind. From their feedback, it seemed that once participants had decided which sentence to write, they occasionally hit against the wall of the controlled environment, where coming up with a syntactically acceptable formulation was not always straightforward. The main issue were missing content words that the participants wanted to use, for example ‘European’, ‘ocean’, ‘hill’, ‘Great Britain’. The sentence structures which the participants claimed they could not write included coordination of phrases (e.g. ‘The Danube flows through Germany, Austria and Hungary.’, ‘Die Hauptstadt von Portugal ist nicht Lissabon oder Madrid.’); adversative constructions (e.g. ‘Norway borders Sweden but not France.’); peripheral arguments of verbs (e.g. ‘There is a volcano in Switzerland.’, ‘French is a language spoken in France.’); triadic relations (e.g. ‘X lies between Y and Z.’); and comparative constructions (‘more than’). The lack of noun phrase coordination was reported most often and by speakers of all languages involved, usually referring to it as the “lack of repetitive structures”.

In most cases the grammar simply lacked the reported structures. Obviously, the lexicon did not cover all words needed to describe the European geography. Some of the syntactic structures have been excluded in ACE by design, mostly because they feature structural ambiguity that might not be always visible to the users, but that nevertheless would make a deterministic mapping to a formal logical form impossible. This is the case with the elliptical noun phrase coordination in the above examples. In a few cases, however, the

Table 2

The OWL axiom distribution by type (listing only types that occurred more than once)

| Axiom type              | All | True | Comment  |
|-------------------------|-----|------|--|
| ClassAssertion          | 104 | 54   | Usually a simple statement asserting an individual into a class ('Limmat is a river.')   |
| ObjectPropertyAssertion | 92  | 60   | Usually a simple statement asserting a relation between two individuals ('The Aar flows into the Rhein.')  |
| DisjointClasses         | 44  | 22   | General statement asserting a disjointness of two classes ('No language is a country.')  |
| SubClassOf              | 41  | 19   | General statement relating two classes ('Every capital is a city.', 'Every river that flows into a lake is ...')   |
| SWRL rule               | 8   | 4    | General statements that failed to map to OWL but could be mapped to SWRL. Typically similar to SubClassOf-statements but with a more flexible anaphoric reference structure. |
| Total                   | 290 | 160  |  |

constructs were available but simply not easily discoverable. The participants also reported that the categorization of some words in the look-ahead editor was confusing, such as the preposition 'in', which appeared in the category "adjectives", and the definite article 'the' that appeared in the category "proper names". This was caused by the fact that the GF look-ahead engine returned the results in a flat list, and in order to make it available via the AceWiki look-ahead editor, we structured the list by word class, but with a context-insensitive method which sometimes worked incorrectly.

20% of the participants reported technical issues using the evaluation wiki, such as the website crashing, needing to reload the webpage, and the look-ahead editor not registering mouse clicks for some seconds. However, these were reported to be only temporary hiccups from which the wiki engine recovered on its own. We therefore believe that these technical issues did not affect the evaluation.

## 7. Conclusion and future work

The main contribution of our work is the study of CNL-based knowledge engineering in a semantic wiki environment. The main novelty with respect to previous work is making the wiki environment multilingual. As underlying technologies we have used Attempto Controlled English, which is a user-friendly formal language and provides a mapping to the expressive ontology language OWL, and Grammatical Framework, which was used to implement ACE in a multilingual way as a modular wiki-friendly grammar. We have built the wiki implementation on top of Ace-

Wiki, an existing monolingual semantic wiki engine. In order to make our system multilingual, the architecture of AceWiki was generalized. Although the underlying implementation has become more complex, the user interface has largely remained the same. On the (multilingual) lexicon editing side, this is mainly due to the support for smart paradigms that GF provides via its RGL. Our evaluation with 30 participants has shown that automatic translation barely affects communication among users of AceWiki-GF.

The presented work can be extended in various more general directions. Although the current system is ACE-based, its general architecture allows for any grammar to be used as the basis of the wiki content as long as it is implemented in GF. Such alternative grammars might not map naturally to a language like OWL and are thus less interesting in the context of the Semantic Web. Examples of existing GF grammars are a tourist phrase book, a query language over a museum catalog, and a collection of mathematics exercises, all of which could be loaded to AceWiki-GF with only little additional work. Wikis for such domains would mainly profit from the supported multilinguality and less from OWL-based reasoning, or may need other forms of reasoning.

Another direction is to improve the grammar editing features of the environment and to develop the system into a tool for collaboratively designing CNLs. The wiki users could take for example the ACE grammar as starting point and customize it for a specific domain, possibly changing some of its original features and design decisions. The wiki sentences could then serve as unit/regression test sets to check the currently effective grammar implementation.

The sentence editor could integrate some ideas from other CNL editors, e.g. the GF Syntax Editor<sup>9</sup> or the OWL Simplified English editor [33], which operate more on the abstract tree level and thus avoid the problem of ambiguous entries. These approaches also simplify smaller edits, such as replacing a word in the beginning of the sentence. The fact that they abstract away from linguistic details like case and gender might make them preferable for users with only basic knowledge of the underlying language.

More experiments and case studies will be needed in the future to investigate how this kind of collaborative ontology editing scales up to larger teams, longer time periods, more complex ontologies, and a larger number of supported languages. Other aspects that deserve more attention include the ambiguity handling, users' responses to reasoner feedback, and grammar editing.

On a longer term, our approach could contribute to the creation of a truly multilingual Wikipedia where content is stored in a semantic representation that can be verbalized in different languages. This would make all content instantly available in all languages (once the required vocabulary is defined in a multilingual way and annotated with the relevant syntactic features). In this way, users of all language versions would get the full power of Wikipedia, breaking the current dominance of English and putting an end to the lock-out of users speaking less widespread or underrepresented languages.

## References

- [1] Krasimir Angelov. *The Mechanics of the Grammatical Framework*. PhD thesis, Chalmers University of Technology, 2011.
- [2] Jie Bao, Paul R. Smart, Nigel Shadbolt, Dave Braines, and Gareth Jones. A Controlled Natural Language Interface for Semantic Media Wiki. In *3rd Annual Conference of the International Technology Alliance (ACITA'09)*, September 2009.
- [3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In Laks V. S. Lakshmanan, Raymond T. Ng, and Dennis Shasha, editors, *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, New York NY, USA, 2008.
- [4] Björn Bringert, Krasimir Angelov, and Aarne Ranta. Grammatical Framework Web Service. In Kurt Kreutel, editor, *Proceedings of the Demonstrations Session at EACL 2009*, pages 9–12. Association for Computational Linguistics, Stroudsburg PA, USA, 2009.
- [5] François Bry, Sebastian Schaffert, Denny Vrandečić, and Klara Weiland. Semantic Wikis: Approaches, Applications, and Perspectives. In Thomas Eiter and Thomas Krennwallner, editors, *Reasoning Web. Semantic Technologies for Advanced Query Answering*, volume 7487 of *Lecture Notes in Computer Science*, pages 329–369. Springer, Berlin/Heidelberg, Germany, 2012.
- [6] Paul Buitelaar, Key-Sun Choi, Philipp Cimiano, and Eduard H. Hovy, editors. *The Multilingual Semantic Web*, volume 2 of *Dagstuhl Reports*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2013.
- [7] John J. Camilleri, Norbert E. Fuchs, and Kaarel Kaljurand. Deliverable D11.1. ACE Grammar Library. Technical report, MOLTO project, June 2012. <http://www.molto-project.eu/biblio/deliverable/ace-grammar-library>.
- [8] Laura Canedo, Norbert E. Fuchs, Kaarel Kaljurand, Maarit Koponen, Tobias Kuhn, Jussi Rautio, and Victor Ungureanu. Deliverable D11.3. Evaluations of ACE-in-GF and of AceWiki-GF. Technical report, MOLTO project, May 2013. <http://www.molto-project.eu/biblio/deliverable/evaluations-ace-gf-and-acewiki-gf>.
- [9] Philipp Cimiano, Paul Buitelaar, John McCrae, and Michael Sintek. LexInfo: A declarative model for the lexicon-ontology interface. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(1):29–51, 2011.
- [10] Brian Davis, Ramona Enache, Jeroen Van Grondelle, and Laurette Pretorius. Multilingual Verbalisation of Modular Ontologies using GF and Lemon. In Tobias Kuhn and Norbert E. Fuchs, editors, *Proceedings of the Third International Workshop on Controlled Natural Language (CNL 2012)*, volume 7427 of *Lecture Notes in Computer Science*, pages 168–184. Springer, Berlin/Heidelberg, Germany, 2012.
- [11] Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. In Cristina Baroglio, Piero A. Bonatti, Jan Małuszyński, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, *Reasoning Web, Fourth International Summer School 2008, Tutorial Lectures*, number 5224 in *Lecture Notes in Computer Science*, pages 104–124. Springer, 2008.
- [12] Chiara Ghidini, Marco Rospocher, and Luciano Serafini. Modeling in a Wiki with MoKi: Reference Architecture, Implementation, and Usages. *International Journal On Advances in Life Sciences*, 4(3):111–124, 2012.
- [13] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. Technical report, W3C, 2012. <http://www.w3.org/TR/owl2-overview/>.
- [14] Normunds Grūzītis. *Formal Grammar and Semantics of Controlled Latvian Language*. PhD thesis, University of Latvia, 2011.
- [15] Kaarel Kaljurand. *Attempto Controlled English as a Semantic Web Language*. PhD thesis, Faculty of Mathematics and Computer Science, University of Tartu, 2007.
- [16] Kaarel Kaljurand. ACE View — an ontology and rule editor based on Attempto Controlled English. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *5th OWL Experiences and Directions Workshop (OWLED 2008)*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 26–27 October 2008.

<sup>9</sup><http://cloud.grammaticalframework.org/syntax-editor/editor.html>

- [17] Kaarel Kaljurand and Tobias Kuhn. A Multilingual Semantic Wiki Based on Attempto Controlled English and Grammatical Framework. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *Proceedings of the 10th Extended Semantic Web Conference (ESWC 2013)*, volume 7882 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 2013.
- [18] Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1993.
- [19] David Karger. Keynote at ESWC Part 3: What's Wrong with Semantic Web Research, and Some Ideas to Fix it. <http://haystack.csail.mit.edu/blog/2013/06/10/keynote-at-eswc-part-3-whats-wrong-with-semantic-web-research-and-some-ideas-to-fix-it/>, 10 June 2013. Haystack Blog.
- [20] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic Wikipedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4):251–261, 2007.
- [21] Tobias Kuhn. AceWiki: A Natural and Expressive Semantic Wiki. In *Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges*, volume 543 of *CEUR Workshop Proceedings*, 2009.
- [22] Tobias Kuhn. How Controlled English can Improve Semantic Wikis. In Christoph Lange, Sebastian Schaffert, Hala Skaf-Molli, and Max Völkel, editors, *Proceedings of the Fourth Workshop on Semantic Wikis, European Semantic Web Conference 2009*, volume 464 of *CEUR Workshop Proceedings*. CEUR-WS, June 2009.
- [23] Tobias Kuhn. *Controlled English for Knowledge Representation*. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, 2010.
- [24] Tobias Kuhn. A Principled Approach to Grammars for Controlled Natural Languages and Predictive Editors. *Journal of Logic, Language and Information*, 22(1):33–70, 2013.
- [25] Tobias Kuhn. The Understandability of OWL Statements in Controlled English. *Semantic Web*, 4(1):101–115, 2013.
- [26] Tobias Kuhn. A Survey and Classification of Controlled Natural Languages. *Computational Linguistics*, 40(1):121–170, March 2014.
- [27] Tobias Kuhn and Alexandre Bergel. Verifiable source code documentation in controlled natural language. *Science of Computer Programming*, 2014.
- [28] Tobias Kuhn and Stefan Höfler. Coral: Corpus Access in Controlled Language. *Corpora*, 7(2):187–206, 2012.
- [29] Ora Lassila. Semantic web soul searching. [http://www.lassila.org/blog/archive/2007/03/semantic\\_web\\_so\\_1.html](http://www.lassila.org/blog/archive/2007/03/semantic_web_so_1.html), 19 March 2007. Wilbur and-O blog.
- [30] Martin Luts, Daniel Tikkerbär, Monika Saarmann, and Marius Kutateladze. Towards a Community-Driven Controlled Natural Languages Evolution. In Michael Rosner and Norbert E. Fuchs, editors, *Pre-Proceedings of the Second Workshop on Controlled Natural Languages (CNL 2010)*, volume 622 of *CEUR Workshop Proceedings*. CEUR-WS, 2010.
- [31] Moisés S. Meza-Moreno and Björn Bringert. Interactive Multilingual Web Applications with Grammatical Framework. In Bengt Nordström and Aarne Ranta, editors, *Advances in Natural Language Processing, 6th International Conference, GoTAL 2008, Gothenburg, Sweden*, volume 5221 of *Lecture Notes in Artificial Intelligence*, pages 336–347. Springer, Berlin/Heidelberg, Germany, August 2008.
- [32] Jonathan Pool. Can Controlled Languages Scale to the Web? In *5th International Workshop on Controlled Language Applications*. Machine Translation Archive, [www.mt-archive.info](http://www.mt-archive.info), 2006.
- [33] Richard Power. OWL Simplified English: A Finite-State Language for Ontology Editing. In Tobias Kuhn and Norbert E. Fuchs, editors, *Proceedings of the Third International Workshop on Controlled Natural Language (CNL 2012)*, volume 7427 of *Lecture Notes in Computer Science*, pages 44–60. Springer, Berlin/Heidelberg, Germany, 2012.
- [34] Aarne Ranta. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(2), 2009.
- [35] Aarne Ranta. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford CA, USA, 2011. ISBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth).
- [36] Aarne Ranta and Krasimir Angelov. Implementing Controlled Languages in GF. In Norbert E. Fuchs, editor, *Workshop on Controlled Natural Language, CNL 2009, Marettimo Island, Italy, June 8-10, 2009, Revised Papers*, volume 5972 of *Lecture Notes in Computer Science*, pages 82–101. Springer, Berlin/Heidelberg, Germany, 2010.
- [37] Aarne Ranta, Ramona Enache, and Grégoire Détrez. Controlled Language for Everyday Use: the MOLTO Phrasebook. In Michael Rosner and Norbert E. Fuchs, editors, *Proceedings of the Second Workshop on Controlled Natural Language (CNL 2010)*, volume 7175 of *Lecture Notes in Computer Science*, pages 115–136. Springer, Berlin/Heidelberg, Germany, 2012.
- [38] Adam Wyner, Krasimir Angelov, Guntis Barzdins, Danica Damljanovic, Brian Davis, Norbert Fuchs, Stefan Hoefler, Ken Jones, Kaarel Kaljurand, Tobias Kuhn, Martin Luts, Jonathan Pool, Mike Rosner, Rolf Schwitter, and John Sowa. On Controlled Natural Languages: Properties and Prospects. In Norbert E. Fuchs, editor, *Controlled Natural Language*, volume 5972 of *Lecture Notes in Computer Science*, pages 281–289. Springer, Berlin/Heidelberg, Germany, 2010.