A Multilingual Semantic Wiki Based on Attempto Controlled English and Grammatical Framework

Kaarel Kaljurand and Tobias Kuhn

Institute of Computational Linguistics, University of Zurich

ESWC 2013, Montpellier

2013-05-30

Existing wiki systems

- wiki
    - user-friendly collaborative environment for knowledge management
    - content typically unconstrained natural language (NL), therefore not easily automatically processable
    - powered by software, e.g. MediaWiki
    - e.g. Wikipedia
- semantic wiki (= wiki + formal semantics)
    - provides: richer query language, consistency checking (via automatic reasoning)
    - content typically NL + typed links (i.e. RDF triples)
    - software: Semantic Mediawiki, ...
- controlled natural language (CNL) based semantic wiki
    - semantic wiki using CNL for the content
    - formal languages hidden (=> can use more expressive formal languages)
    - software: AceWiki

Multilingual CNL-based Semantic Wiki

- multiple languages
  - natural: English, German, ...
  - formal: first-order logic, OWL, ...
  - languages for content vs user interface
- CNL-based
  - backed by formal grammar(s)
  - formal languages are hidden
- semantic
  - content automatically kept in sync via precise translation
  - consistency checking, question answering, ... (depending on the domain)
- wiki
  - user-friendly
  - collaborative

Possible use cases

- multilingual ontology editor
  - e.g. environment where users agree on the content and multilingual vocabulary of an OWL-style geography ontology
- tourist phrasebook
  - book structure (chapters and sections)
  - multilingual content presented in parallel
- catalog of museum objects (paintings, painters)
  - each object on its own wiki page
  - rich queries (e.g. "which Dutch painter painted which French painter?")
- SWIFTTT systems (see: David Karger's talk)

Background technologies

Attempto Controlled English (ACE)

- subset of natural English
  - conjunction, disjunction, negation, *if-then*, ...
  - anaphoric references: pronouns, definite noun phrases, variables
  - quantifiers: *every*, *no*, *at least 3*, ...
  - content words: proper names, common nouns, verbs, adjectives, ...
- grammar is fixed, but users can change content words
- deterministic ambiguity handling
  - anaphora resolution (`France borders Spain and it borders Portugal.`)
  - quantifier scope (`Every country borders a country.`)
  - attachment (`Every EU-country borders a country that is a EU-country and is a NATO-country.`)
- well-defined translation to and from first-order logic, OWL, ...
- end-user documentation: construction and interpretation rules, as restrictions of English

ACE reasoning via translation to OWL

Every country that does not border a sea is a landlocked-country.

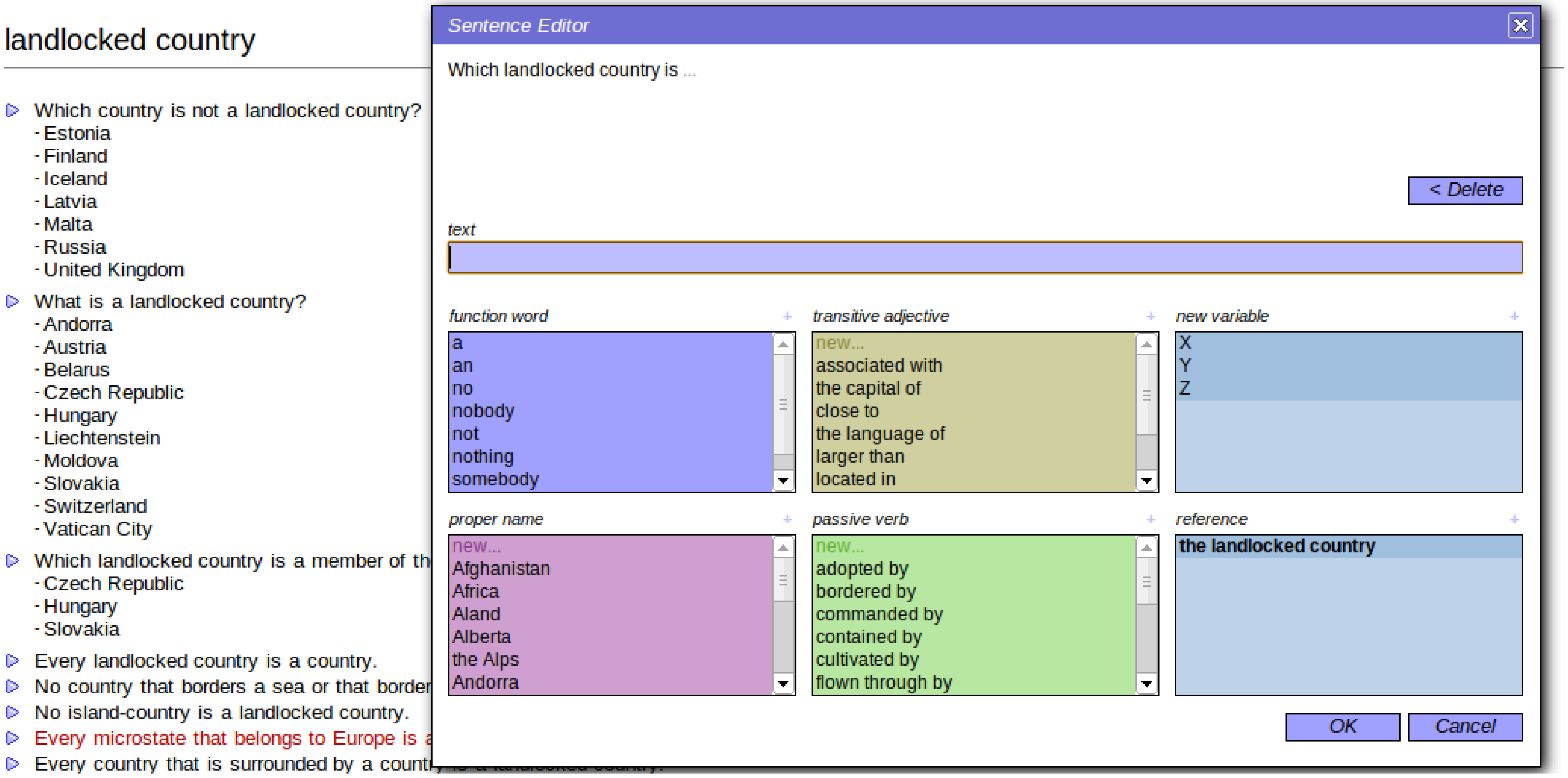
```
SubClassOf(
   ObjectIntersectionOf(
      :country
      ObjectComplementOf(
         ObjectSomeValuesFrom(
            :border
            :sea
         )
      )
   )
   :landlocked-country
)
```

Which country is a landlocked-country?

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

AceWiki

- expressive semantic wiki system
- front-end language ACE; background reasoning language OWL
- monolingual, fixed grammar, no ambiguity handling



landlocked country

▷ Which country is not a landlocked country?
- Estonia
- Finland
- Iceland
- Latvia
- Malta
- Russia
- United Kingdom

▷ What is a landlocked country?
- Andorra
- Austria
- Belarus
- Czech Republic
- Hungary
- Liechtenstein
- Moldova
- Slovakia
- Switzerland
- Vatican City

▷ Which landlocked country is a member of th
- Czech Republic
- Hungary
- Slovakia

▷ Every landlocked country is a country.

▷ No country that borders a sea or that border

▷ No island-country is a landlocked country.

▷ Every microstate that belongs to Europe is a

▷ Every country that is surrounded by a countr is a landlocked country.

Sentence Editor

Which landlocked country is ...

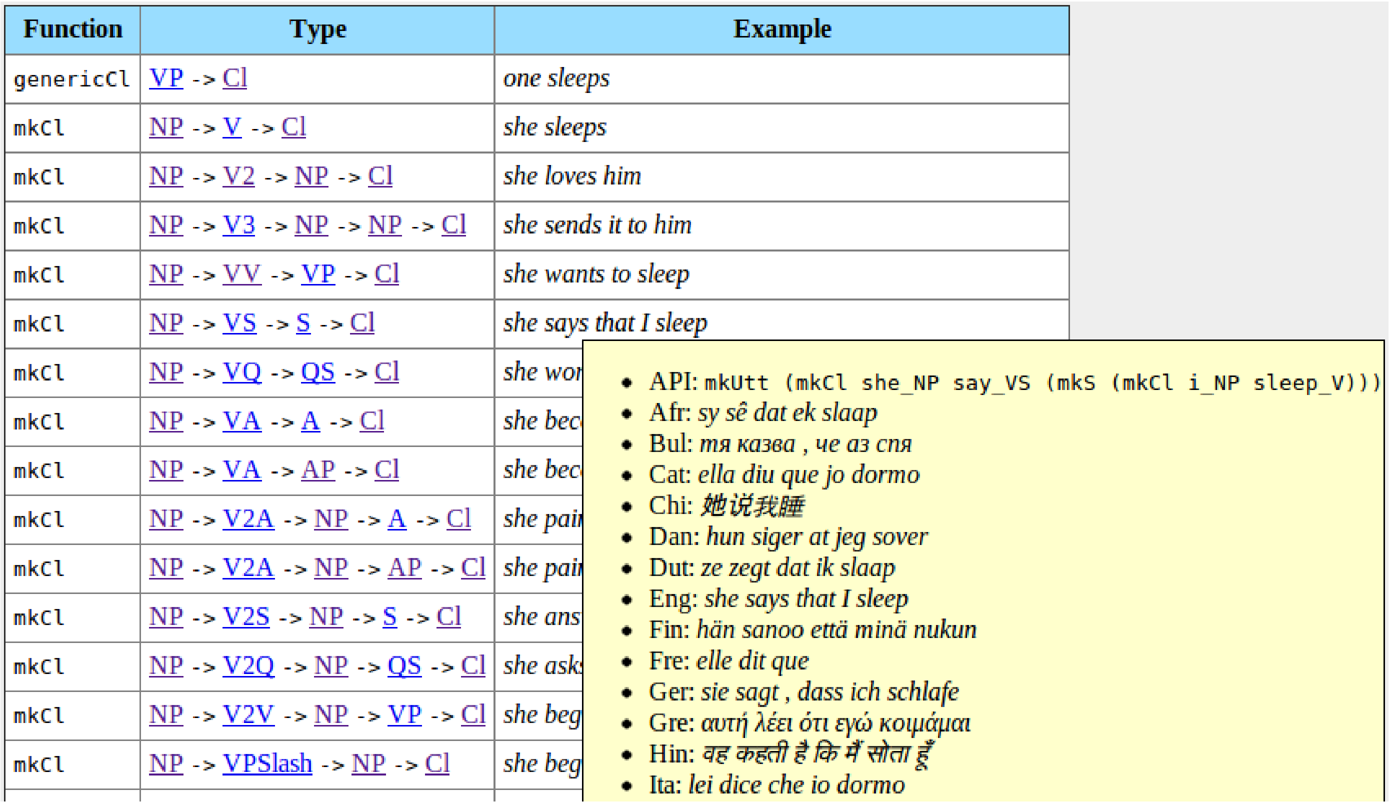
text

Grammatical Framework (GF)

- framework for multilingual grammar engineering
    - functional programming language optimized to handle natural language
    - resource grammar library implementing common morphological and syntactic structures
    - mildly context sensitive
- grammar = language-neutral abstract syntax + multiple concrete syntaxes that implement the abstract functions and categories, specifying words, word order, agreement, etc.
    - `border : Country -> Country -> Relation`
    - **English:** `border x y = x!Nom + "borders" + y!Nom`
    - **Estonian:** `border x y = x!Gen + "naaber on" + y!Nom`
- translation = parse a string in language A to tree(s) + linearize these tree(s) as strings in language B
- parsing (translation, look-ahead, ...) based on Parallel Multiple Context-Free Grammars
- various tools + bindings to Python, Java, Javascript, Prolog, ...

GF Resource Grammar Library (RGL)

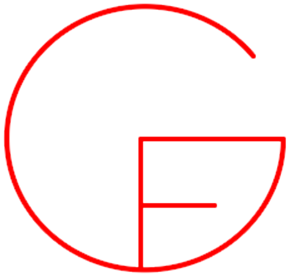
- morphology and syntax for ~30 languages via language-neutral API
- developers do not need detailed knowledge of the languages that they want to support in their application



Function	Type	Example
`genericCl`	VP -> Cl	*one sleeps*
`mkCl`	NP -> V -> Cl	*she sleeps*
`mkCl`	NP -> V2 -> NP -> Cl	*she loves him*
`mkCl`	NP -> V3 -> NP -> NP -> Cl	*she sends it to him*
`mkCl`	NP -> VV -> VP -> Cl	*she wants to sleep*
`mkCl`	NP -> VS -> S -> Cl	*she says that I sleep*
`mkCl`	NP -> VQ -> QS -> Cl	*she wor*
`mkCl`	NP -> VA -> A -> Cl	*she bec*
`mkCl`	NP -> VA -> AP -> Cl	*she bec*
`mkCl`	NP -> V2A -> NP -> A -> Cl	*she pai*
`mkCl`	NP -> V2A -> NP -> AP -> Cl	*she pai*
`mkCl`	NP -> V2S -> NP -> S -> Cl	*she ans*
`mkCl`	NP -> V2Q -> NP -> QS -> Cl	*she ask*
`mkCl`	NP -> V2V -> NP -> VP -> Cl	*she beg*
`mkCl`	NP -> VPSlash -> NP -> Cl	*she beg*

- API: `mkUtt (mkCl she_NP say_VS (mkS (mkCl i_NP sleep_V)))`
- Afr: *sy sê dat ek slaap*
- Bul: *тя казва , че аз спя*
- Cat: *ella diu que jo dormo*
- Chi: *她说我睡*
- Dan: *hun siger at jeg sover*
- Dut: *ze zegt dat ik slaap*
- Eng: *she says that I sleep*
- Fin: *hän sanoo että minä nukun*
- Fre: *elle dit que*
- Ger: *sie sagt , dass ich schlafe*
- Gre: *αυτή λέει ότι εγώ κοιμάμαι*
- Hin: *वह कहती है कि मैं सोता हूँ*
- Ita: *lei dice che io dormo*

AceWiki

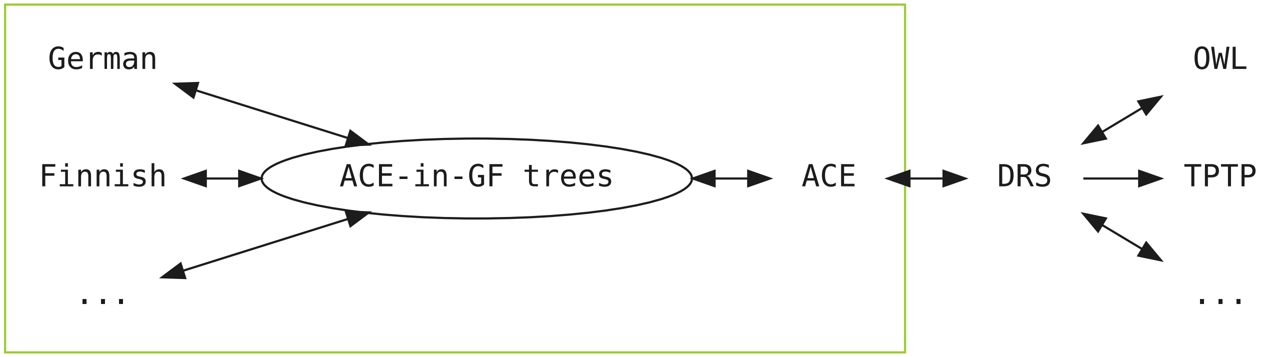


Implementation of AceWiki-GF

- integrate ACE with GF (**ACE-in-GF**)
  - implement a multilingual grammar of ACE in the GF framework
  - cover the languages supported by the GF resource grammar
  - not fine-tuned to any particular language (apart from ACE)
- integrate AceWiki with GF (**AceWiki-GF**)
  - implement connection to GF tools (GF Webservice / Cloud Service)
  - add support for the management of multilinguality, ambiguity, grammar

ACE-in-GF (main idea)

An ACE grammar implemented in GF adds multiple natural languages as front-ends to ACE. As a result, these languages can be mapped to and from various formal languages already supported by ACE.



ACE-in-GF (main idea)

German

Jedes Land, das nicht an ein Meer grenzt, ist ein Binnenland.

ACE-in-GF tree

```
baseText (sText (s (vpS (everyNP (relCN (cn_as_VarCN country_CN)
  (neg_predRS which_RP (v2VP border_V2 (thereNP_as_NP
   (aNP (cn_as_VarCN sea_CN))))))) (npVP (thereNP_as_NP
    (aNP (cn_as_VarCN landlocked_country_CN)))))))
```

ACE

Every country that does not border a sea is a landlocked-country.

OWL

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

ACE in GF

- implementation of the ACE syntax
  - extension of *Angelov and Ranta (CNL 2009)*
  - focus on the subset of ACE that can be mapped to OWL
  - almost 100% coverage at almost 0% ambiguity
  - no direct generation of discourse representation structures (DRS)
- support most RGL languages
  - *Bulgarian*, *Catalan*, *Chinese*, *Danish*, *Dutch*, *English*, *Finnish*, *French*, *German*, *Greek*, *Hindi*, *Italian*, *Latvian*, *Norwegian*, *Polish*, *Romanian*, *Russian*, *Spanish*, *Swedish*, *Thai*, *Urdu*
  - RGL-based design provides automatic increase in quality and language-coverage over time
- status
  - some precision problems, e.g. anaphoric references do not obey DRS accessibility constraints
  - ambiguity and coverage problems in some languages

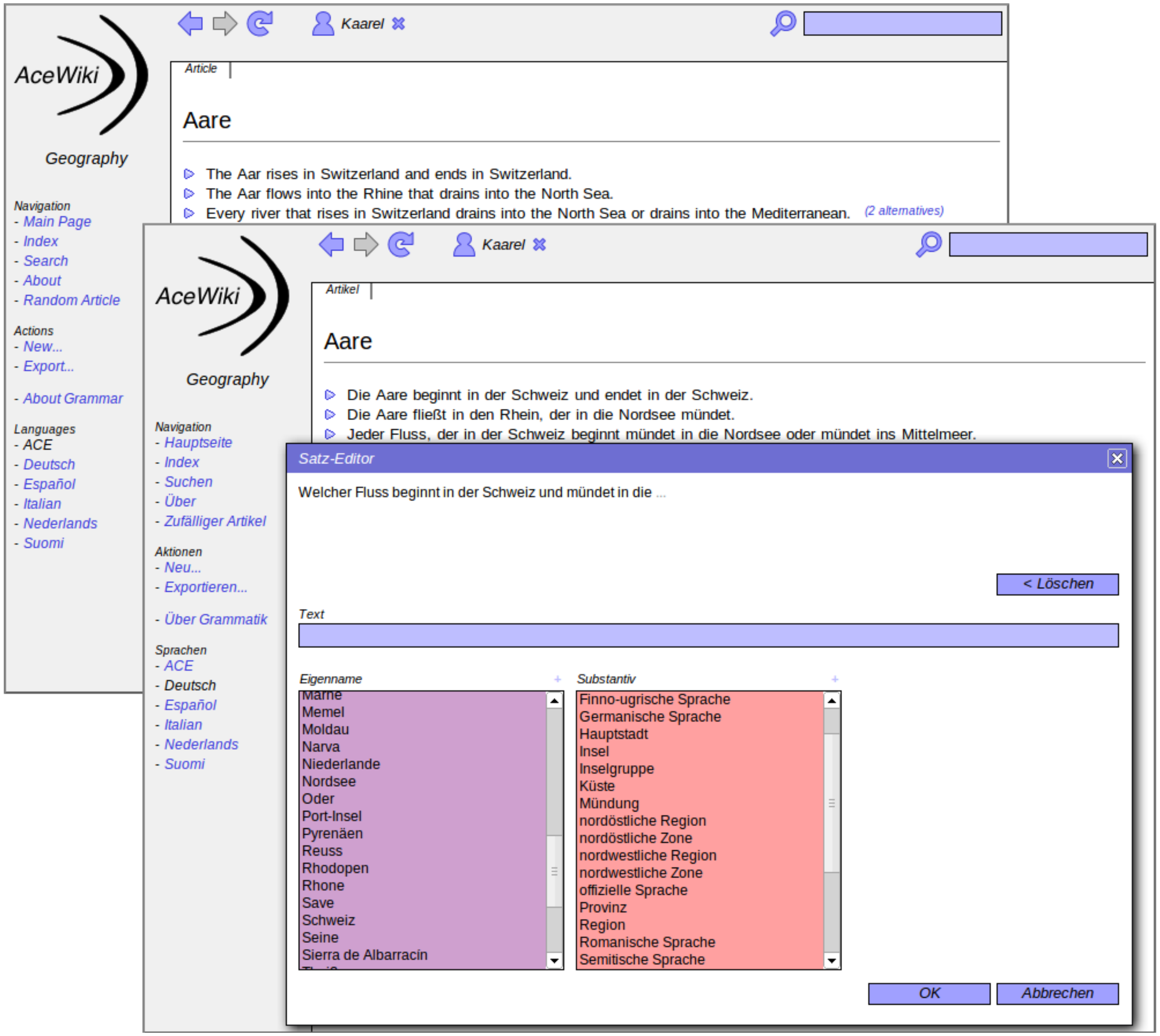
ACE-in-GF translation example

```
ACE: every person that speaks a language X does not forget X .
Bul: всеки човек който говори език X не забравя X .
Cat: cada persona que parla una llengua X no oblida X .
Chi: ? ? ? X ? ? ? ? ? ? ? ? X ?
Dan: hver person , som taler et sprog X glemmer ikke X .
Dut: elke persoon , dat een taal X spreekt vergeet niet X .
Fin: jokainen henkilö , joka puhuu kieltä X ei unohda X:ää .
Fre: chaque personne qui parle une langue X n' oublie pas X .
Ger: jede Person , die eine Sprache X spricht vergißt X nicht .
Gre: κάθε πρόσωπο που μιλά μία γλώσσα τον X δεν ξεχνά τον X .
Hin: हर [person_CN] , जो [language_CN] X बोलता है X नहीं भूलता है .
Ita: ogni persona che parla una lingua X non dimentica X .
Lav: ikviena persona , kas saka valodu X neaizmirst X .
Nor: hver person , som snakker et språk X glemmer ikke X .
Pol: każda osoba , która rozmawia z językiem X nie zapomina X .
Ron: orice persoană care vorbeşte o limbă X nu îl uită pe X .
Rus: каждый лицо , который говорит на языке X не забывает X .
Spa: cada persona que habla una lengua X no olvida X .
Swe: varje person , som talar ett språk X glömmer inte X .
Tha: ????? ??? ?? ??? ??? ???? X ??? ??? X
Urd: جو زبان , شخص ر? X ?? بولتا X ?? بھولتا ?ی?ن
```

AceWiki integration with GF

- wiki content is based on a (single) GF grammar
  - provided by GF Webservice / Cloud service
  - optimized for ACE-in-GF (but other GF grammars can also be used)
- wiki entry is GF abstract tree set
  - viewed via linearization(s)
  - can represent ambiguity
- multilingual viewing and editing of wiki content
  - grammar-based look-ahead editing that shows next possible tokens
  - ambiguity resolution via another concrete language
- grammar integrated into the wiki
  - GF grammars are very modular
  - grammar modules as wiki articles (wiki-linking of grammar and content)
  - grammar can be changed while editing the wiki

ACE-based geography article



Aare

- The Aar rises in Switzerland and ends in Switzerland.
- The Aar flows into the Rhine that drains into the North Sea.
- Every river that rises in Switzerland drains into the North Sea or drains into the Mediterranean.

Aare

- Die Aare beginnt in der Schweiz und endet in der Schweiz.
- Die Aare fließt in den Rhein, der in die Nordsee mündet.
- Jeder Fluss, der in der Schweiz beginnt mündet in die Nordsee oder mündet ins Mittelmeer.

Welcher Fluss beginnt in der Schweiz und mündet in die ...

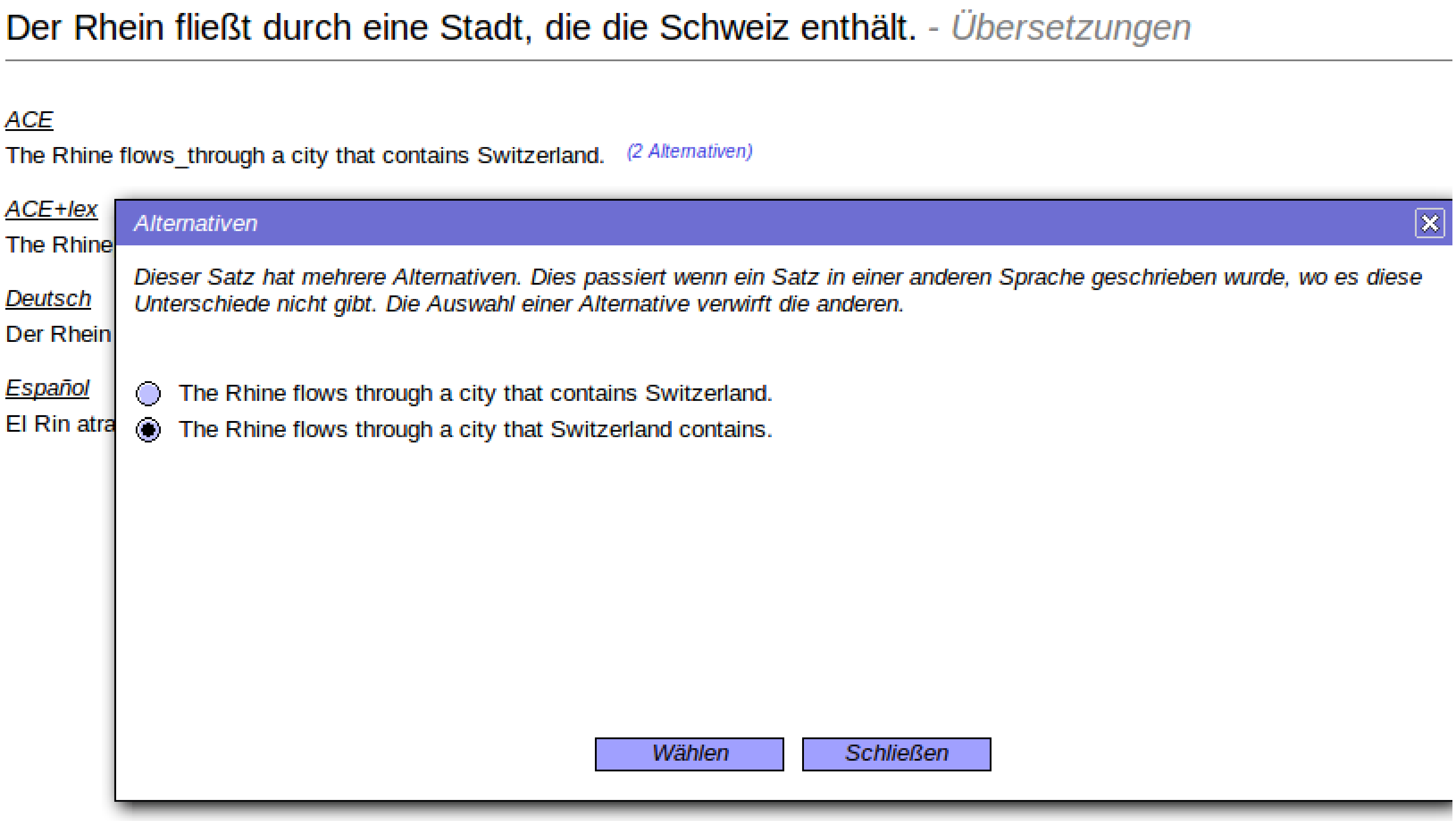
Eigenname

Memel
Moldau
Narva
Niederlande
Nordsee
Oder
Port-Insel
Pyrenäen
Reuss
Rhodopen
Rhone
Save
Schweiz
Seine
Sierra de Albarracin

Substantiv

Finno-ugrische Sprache
Germanische Sprache
Hauptstadt
Insel
Inselgruppe
Küste
Mündung
nordöstliche Region
nordöstliche Zone
nordwestliche Region
nordwestliche Zone
offizielle Sprache
Provinz
Region
Romanische Sprache
Semitische Sprache

Ambiguity resolution



Der Rhein fließt durch eine Stadt, die die Schweiz enthält. - *Übersetzungen*

ACE

The Rhine flows_through a city that contains Switzerland. *(2 Alternativen)*

ACE+lex

The Rhine

Deutsch

Der Rhein

Español

El Rin atra

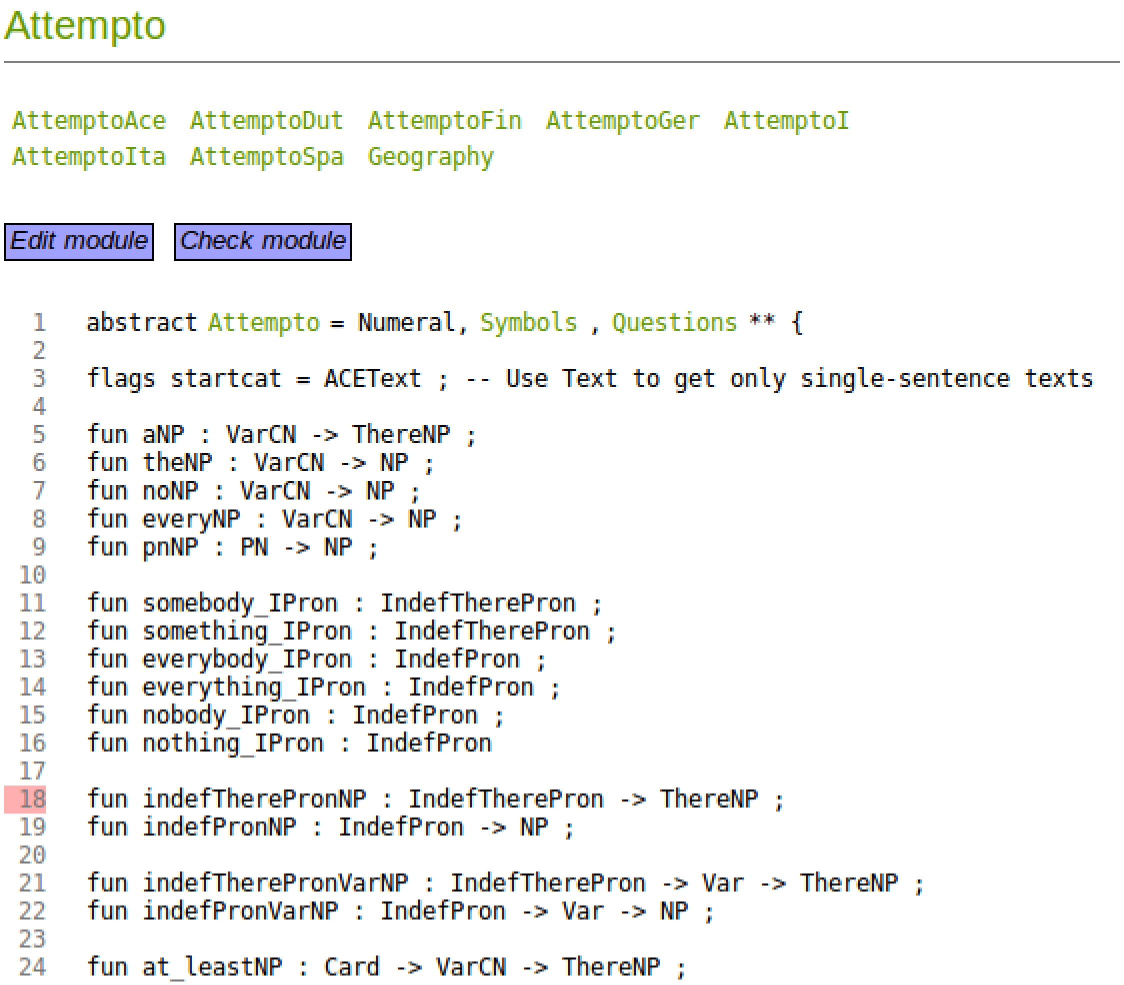
Alternativen

Dieser Satz hat mehrere Alternativen. Dies passiert wenn ein Satz in einer anderen Sprache geschrieben wurde, wo es diese Unterschiede nicht gibt. Die Auswahl einer Alternative verwirft die anderen.

○ The Rhine flows through a city that contains Switzerland.

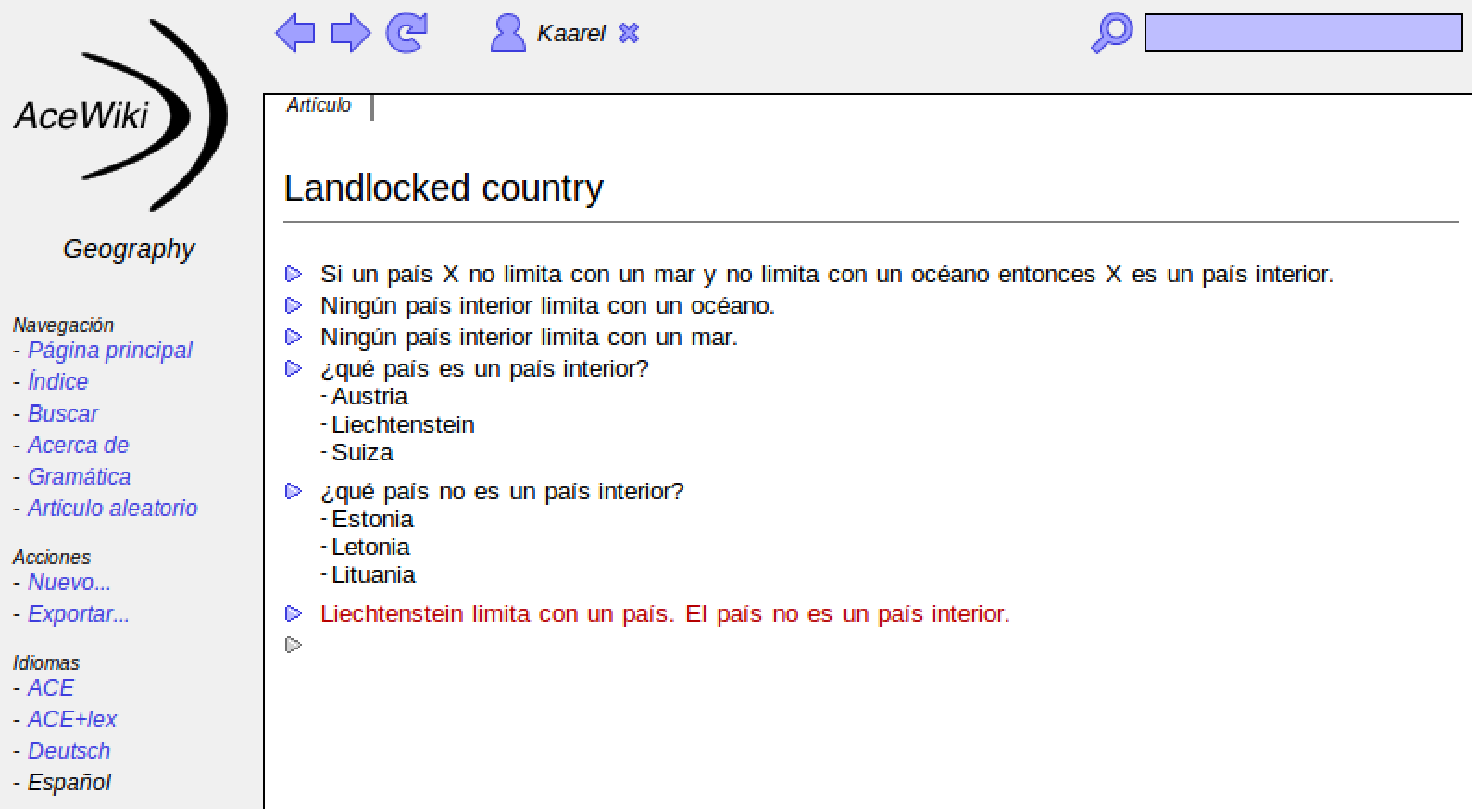
● The Rhine flows through a city that Switzerland contains.

Grammar module page


Attempto
AttemptoAce AttemptoDut AttemptoFin AttemptoGer AttemptoI
AttemptoIta AttemptoSpa Geography
Edit module
Check module

```
abstract Attempto = Numeral, Symbols , Questions ** {

flags startcat = ACEText ; -- Use Text to get only single-sentence texts

fun aNP : VarCN -> ThereNP ;
fun theNP : VarCN -> NP ;
fun noNP : VarCN -> NP ;
fun everyNP : VarCN -> NP ;
fun pnNP : PN -> NP ;

fun somebody_IPron : IndefTherePron ;
fun something_IPron : IndefTherePron ;
fun everybody_IPron : IndefPron ;
fun everything_IPron : IndefPron ;
fun nobody_IPron : IndefPron ;
fun nothing_IPron : IndefPron

fun indefTherePronNP : IndefTherePron -> ThereNP ;
fun indefPronNP : IndefPron -> NP ;

fun indefTherePronVarNP : IndefTherePron -> Var -> ThereNP ;
fun indefPronVarNP : IndefPron -> Var -> NP ;

fun at_leastNP : Card -> VarCN -> ThereNP ;
```

Automatic question answering



Landlocked country

- Si un país X no limita con un mar y no limita con un océano entonces X es un país interior.
- Ningún país interior limita con un océano.
- Ningún país interior limita con un mar.
- ¿qué país es un país interior?
  - Austria
  - Liechtenstein
  - Suiza
- ¿qué país no es un país interior?
  - Estonia
  - Letonia
  - Lituania
- Liechtenstein limita con un país. El país no es un país interior.

Evaluation of ACE-in-GF

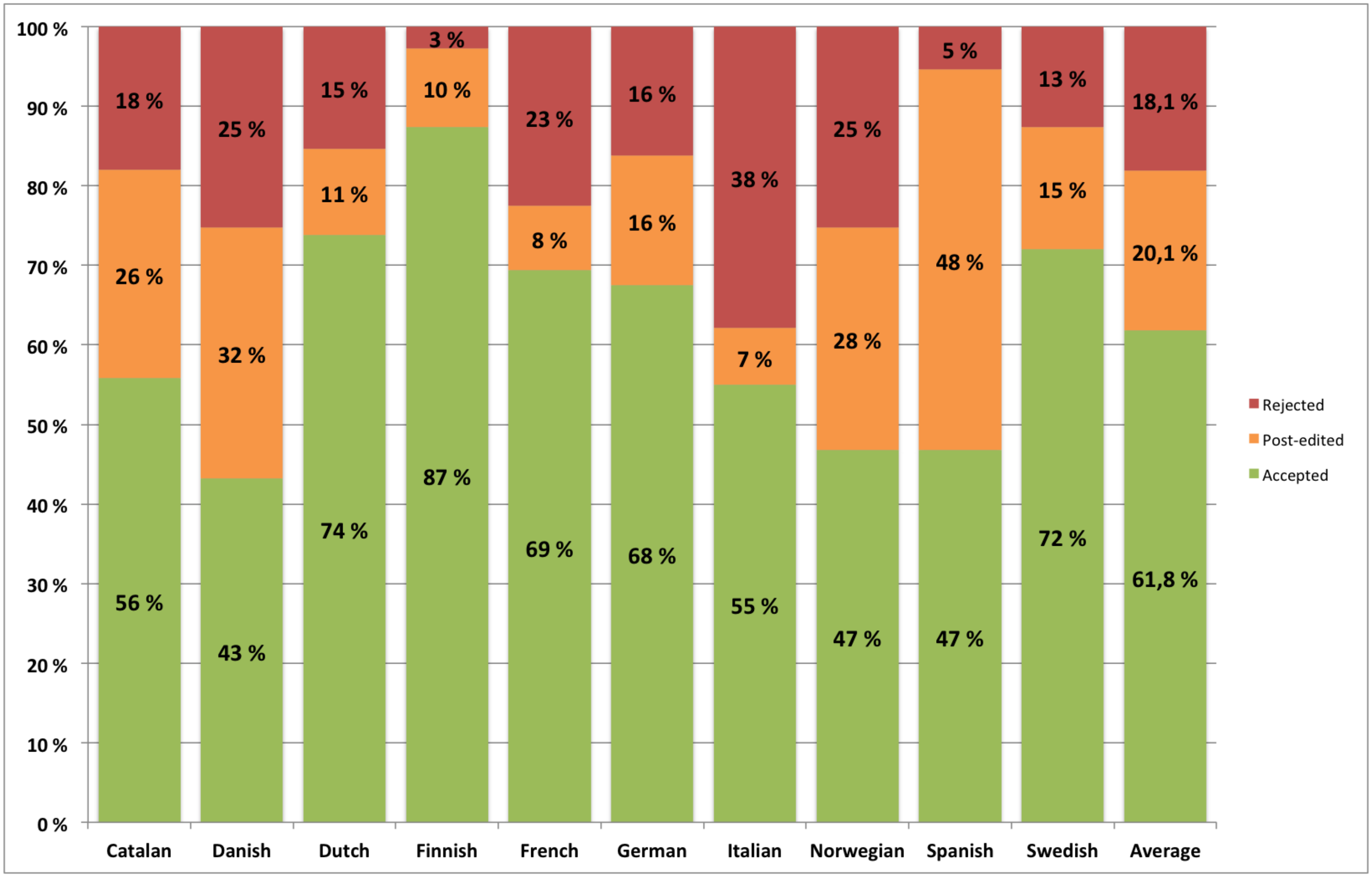
Design

- generate ~100 ACE sentences/questions and automatically translate them to all the languages
  - full coverage of all the grammar functions
  - large coverage of OWL axiom structures (subclass, range, domain, transitivity, ...)
- measure translation accuracy from ACE to other languages
- use Google Translate as the baseline
- 20 human evaluators (2 per language) as the gold standard

Results

- participants preferred ACE-in-GF translations to Google translations and post-edited them less
- many edits were stylistic (e.g. users preferred elliptical sentences)

Evaluation of ACE-in-GF



Evaluation of AceWiki-GF

Design

- develop a 500-word geography lexicon
  - 3 languages: English, German and Spanish
  - 3 authors (incl. native speakers of German and Spanish, and a GF engineer)
- ask users of different languages to supply the wiki with sentences and tag each as *true* or *false*
- ask them then to evaluate others' sentences as *true* or *false*
- measure the user (dis)agreement and how much it is influenced by the automatic translation

Results

- 30 participants entered 316 sentences, covering all syntactic functions
- AceWiki-GF user interface was found to be easy to use
- agreement level was ~83% with no significant influence from the translation

Future work

- generalize to handle other types of grammars and reasoning
- improve collaborative grammar editing features
- improve ambiguity management (e.g. automatic reasoning-based ambiguity resolution)
- use the wiki content to automatically generate documentation, grammar fragments, look-ahead editor customizations, etc. for novice users

Links

- ACE-in-GF
  - multilingual ACE grammar
  - source code: http://github.com/Attempto/ACE-in-GF
- AceWiki-GF
  - multilingual CNL-based semantic wiki engine
  - source code: http://github.com/AceWiki/AceWiki
  - demo wikis: http://attempto.ifi.uzh.ch/acewiki-gf/
- MOLTO project: http://www.molto-project.eu/
  - see the deliverables of WP11
- Attempto project: http://attempto.ifi.uzh.ch/
- Grammatical Framework project: http://www.grammaticalframework.org/
  - GF Summer School @ Frauenchiemsee (August 2013): http://school.grammaticalframework.org/2013/

Thank You!