



An Introduction to AceWiki

Tobias Kuhn

Department of Informatics
University of Zurich

Second Workshop on Controlled Natural Language
14 September 2010
Marettimo (Italy)

This Talk

Introduction to the AceWiki system

- Practical introduction on how to use AceWiki
- Discussion of some theoretical and technical aspects
- Running “Hands on” Exercise

AceWiki Attributes

- natural** Uses Attempto Controlled English (ACE)
- semantic** Integrates an OWL reasoner
 - wiki** Collaborative web-based knowledge management
- modern** Dynamic Ajax-based web interface
- alpha** Relatively stable, but some important features are still missing
- free** Open Source under LGPL
- popular** about 20 web site clicks and 1.6 downloads per day

AceWiki Interface

AceWiki

Navigation:

- [Main Page](#)
- [Index](#)
- [Random Article](#)
- [Search:](#)

Actions:

- [New Word...](#)
- [Export...](#)

Article | [Noun](#) | [References](#) | [Individuals](#) | [Hierarchy](#)

planet

- ▶ We use here the definition of "planet" according to the International Astronomical Union (see <http://www.iau.org>) without the restriction to solar planets.
- ▶ Every planet is a celestial-body.
- ▶ No planet is a star.
- ▶ No planet is a dwarf-planet.
- ▶ The distinction between planet and dwarf-planet has been introduced by the International Astronomical Union in 2006.
- ▶ No planet is a moon.
- ▶ Every planet orbits a star.
- ▶ Every planet that does not orbit the Sun is an extrasolar planet.
- ▶ Which planets orbit the Sun?
 - Earth
 - Jupiter
 - Mars
 - Mercury
 - Neptune
 - Saturn
 - Uranus
 - Venus
- ▶ Every planet is a terrestrial planet or is a gas giant.

1. Getting Started

Visit the following AceWiki instance:

`http://192.168.1.172:9077/webapps/acewikicnl2010/`

This wiki will be used during this talk for the exercises. It is still empty but we will step by step fill it with content.

2. Individuals

Add some individuals to the wiki, for example:

- *Italy*
- *CNL 2010*

Add an individual that represents you:

- *Tobias Kuhn*

Design Principles

CNL user interfaces in general ...

❶ ... should follow the natural spirit of CNLs.

“Sentence” instead of “axiom”, “noun phrase” instead of “concept description”, etc.

❷ ... should solve the writability problem of CNLs.

Error messages and suggestions, predictive editor, or language generation features

❸ ... should not let users confuse CNL with natural language.

Users can learn — consciously or unconsciously — the differences between the CNL and the full natural language.

Predictive Editor

A predictive editor shows all possible continuations of a partial text.

Challenges:

- Potentially high number of possible words
- Sentence creation should be quick (not much more than one mouse click per word)
- Novice versus experienced users
- Creation of new words
- Suitable grammar notation → *my talk tomorrow*

3. Concepts

Add some concepts to the wiki, for example:

- *person*
- *country*
- *workshop*
- *researcher*

Connect the existing individuals to the concepts, for example:

- *Italy is a country.*

Only write statements that are true in the real world.

Articles, Words and Statements

Articles \Leftrightarrow **Words**. Every word has exactly one article and every article is assigned to exactly one word.

- Proper names (Individuals)
- Nouns (Concepts)
- *Of*-constructs (Relations)
- Transitive Verbs (Relations)
- Transitive Adjectives (Relations)

Statements. Every article consist of zero or more ordered statements.

- Declarative ACE Sentences (Assertions)
- ACE Questions (Queries)
- Comments (Informal Notes)

4. Hierarchies

Structure the existing concepts in hierarchies, for example:

- *Every researcher is a person.*
- *Every person is an entity.*
- *Every workshop is an event.*

Define new concepts when necessary.

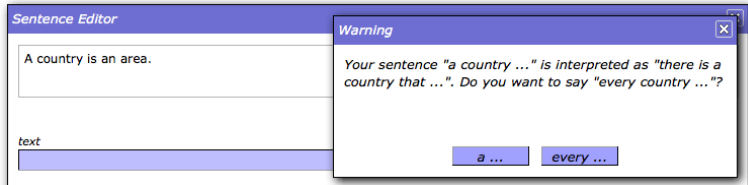
ACE and Full Natural Language in AceWiki

ACE — *Full English*

- ACE sentences are the main content → normal font
- Full English in labels, explanations, comments, etc. → *italics*
- Users can immediately see what is ACE and what not

Pattern-based Suggestions

- Can some frequent errors be automatically recognized?
- Two small user experiments showed that the most common error is using “a” instead of “every” at the sentence beginning.
- Simple correction:
A student studies at a university.
→
Every student studies at a university.
- AceWiki can now automatically detect and correct such errors:



5. Relations

Define some relations in the form of *of*-constructs, verbs, and transitive adjectives, for example:

- *member of*
- *organizes*
- *located in*

Write some simple statements using such relations, for example:

- *Mike Rosner organizes CNL 2010.*
- *Italy is located in Europe.*

Reasoning in AceWiki

- Seamless reasoner integration
- Reasoning happens silently in the background
- ACE sentences are translated into OWL
- OWL Reasoners:
 - HermiT (internal, default)
 - Pellet (internal, not part of the AceWiki package)
 - OWLlink (connection to external reasoners): Pellet, FaCT++, RacerPro, ...

Questions

- Questions in ACE are automatically answered
- Simple *wh*-questions with exactly one *wh*-word:
Which cities are located in a country that borders Switzerland?
- Correspond to class descriptions in OWL
- The answers are the individuals that belong to the given class description
- More expressive query languages like SPARQL could be supported in the future

6. Questions

Add some questions to the wiki and see what happens, for example:

- *Who is a person?*
- *What is CNL 2010?*
- *Who is a member of a group that is located in Italy?*
- *Which workshops are organized by Mike Rosner?*
- *Who organizes more than 2 events?*

Expressivity Layers

- Trade-off: [Expressivity/Complexity versus Reasoning Costs](#)
- Layered Approach:
 - ① Simple statements, immediate feedback from the reasoner
 - ② More complex statements, feedback every x hours or upon explicit request
 - ③ Complex statements, no reasoning within the wiki
 - ④ Very complex statements, formal but too complex to do reasoning
- Parameters for the bounds of layers 1–3
- Expressivity layers in AceWiki:
 - ① OWL Full or an OWL profile (OWL EL, OWL QL or OWL RL)
 - ② (Layer 2 is not yet available)
 - ③ OWL + SWRL
 - ④ First-order Logic

7. Complex Statements

Add some complex statements to AceWiki, using negation, “*if ... then*”, number restrictions, etc:

- *No person is an object.*
- *Every city is located in exactly 1 country.*
- *Everything that organizes something is a person.*
- *If a country X borders a country Y then Y borders X.*

Unique Name Assumption

- Individuals with different names are implicitly considered distinct:
 - *Italy is not Switzerland.*
 - *Italy is not Germany.*
 - ...
- Neither applied in standard logic nor in natural language:
 - *"Bobby", "Bob Dylan", "Robert Zimmerman"*
 - *"Burma", "Birma", "Myanmar"*
- Closed systems like AceWiki should force users to use the same name for the same thing
- Exception: Abbreviations
 - *"the United Nations" = "the UN"*
 - *"the Second Workshop on Controlled Natural Language 2010" = "CNL 2010"*

Consistency

- **Consistency is very important:** required for all other reasoning tasks
- Inconsistencies are probable in a collaborative wiki approach
- Consistency assurance in AceWiki:
 - New sentences are added only after a consistency check
 - Error message is generated when consistency is violated
 - → **AceWiki ontology is always consistent**

8. Inconsistency

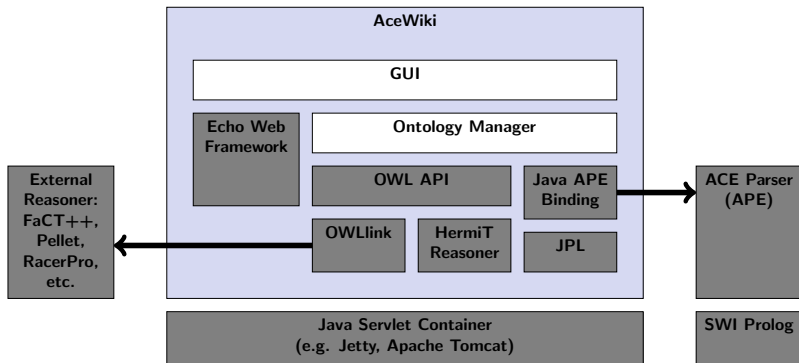
Try to add a statement that is in contradiction with the existing knowledge, for example:

- *No country is located in Europe.*
- *Mike Rosner does not organize an event.*

See what happens. Remove this statement afterwards.

Architecture

AceWiki uses a number of third-party libraries and systems:



Storage, Import and Export

Persistent Storage:

- 1 directory per ontology
- 1 file per article

Server-side Import:

- AceWiki data format

Client-side Export:

- ACE text & lexicon
- OWL
- CSV tables
- AceWiki data format

9. Export

Download the content onto your computer using the export feature of AceWiki. Use for example one of the following formats:

- ACE Text
- Statement Table
- AceWiki data file

Evaluation

- Two small usability experiments with earlier versions of AceWiki
 - Altogether 26 untrained participants
 - Task: Collaborative creation of a knowledge base
- Results:
 - 78%-81% of the sentences were correct and sensible
 - 61%-70% of them were complex (containing negation, implication, disjunction or number restriction)
 - Creation of a correct sentence every 5–6 minutes
 - Definition of a new word every 5–7 minutes
- → Even untrained users can effectively use AceWiki

Missing Features

Some crucial features are still missing:

- User management
- History / Undo Facility

Running Exercise

10. Feedback

If you like to give feedback about your impression of AceWiki, please do so by writing a comment in the article with your name.

ACE Editor

- Web-based editor for ACE texts
- Not a finished tool, but a basis for domain-specific tools
- Uses the same predictive editor as AceWiki, but with a larger subset of ACE
- <http://attempto.ifi.uzh.ch/webapps/aceeditor/>

ACE Editor Interface

The screenshot shows the ACE Editor interface with a browser window at the top displaying the URL `http://atempto.ifi.uzh.ch/webapps/aceeditor/`. The main content area contains a list of sentences:

- ▼ Every man is a human.
 - Syntax Tree: A hierarchical tree diagram for the sentence "every man is a human .". The root node is "specification", which branches into "a" and "vp". "a" branches into "np" and "vp". "np" branches into "det" (every) and "nbar" (man). "vp" branches into "aux" (is) and "np". The second "np" branches into "det" (a) and "nbar" (human).
- ▶ Every woman is a human.
- ▼ Every human is a person and every person is a human.
 - Paraphrase: A text box containing the sentence "Every human is a person. Every person is a human."
 - Syntax Boxes: A diagram showing the sentence "every human is a person and every person is a human ." with boxes around "every human", "is a person", "every person", and "is a human".
- ▶ Every person who owns at least 2 cars is rich.
- ▶ No man is a woman.
- ▶ If X loves Y then X likes Y.
- ▶ John is a man.
- + |

Conclusion

- Ontology access, management and reasoning via CNL
- Such interfaces are easy to use even for untrained users
- Hide the complexity of formal logic and reasoning
- Predictive editor approach works

Future

Some ideas for the future of AceWiki:

- Fully fledged and highly customizable AceWiki
- Grammar editor: create, extend and modify CNL grammars (syntax and semantics)
- General reasoner interface: plug in your own reasoner, rule engine, theorem prover, data base management system, ...
- → Useful in industry (?)

Thank you for your attention!

Questions & Discussion