

Cicero: Tracking Design Rationale in Collaborative Ontology Engineering

Klaas Dellschaft, Hendrik Engelbrecht, José Monte Barreto, Sascha Rutenbeck,
and Steffen Staab

Universität Koblenz-Landau, ISWeb Working Group
Universitätsstr. 1, 56070 Koblenz, Germany
{klaasd, engelbrecht, monte, srutenbeck, staab}@uni-koblenz.de,
WWW home page: <http://isweb.uni-koblenz.de>

Abstract. Creating and designing an ontology is a complex task requiring discussions between domain and ontology engineering experts as well as the users of an ontology. We present the Cicero tool, that facilitates efficient discussions and accelerates the convergence to decisions. Furthermore, by integrating it with an ontology editor, it helps to improve the documentation of an ontology.

1 Introduction

Creating and designing an ontology is a complex task that requires the collaboration of domain and ontology engineering experts. For coming to a consensual model of a domain that is expressed by an ontology, the participants in the engineering process must discuss their different viewpoints in an efficient manner. Thus, discussions are an important part of collaborative ontology engineering.

In the following, we will present the Cicero tool that has been developed in the context of the NeOn project.¹ It facilitates an asynchronous discussion and decision taking process between participants of an ontology engineering project. Two main objectives of capturing discussions in Cicero can be distinguished:

- *Higher efficiency:* Cicero supports its users in discussing the design rationale of ontologies. The whole discussion including the pro and contra arguments is recorded, leading to fewer redundancies in disputes. It has been shown that the applied discussion methodology facilitates efficient discussions and accelerates convergence to a solution.
- *Enhanced documentation:* The captured discussions reflect the design rationale of an ontology. By attaching a discussion to the entities in the ontology, it is possible later to understand why certain elements are modeled as they are. Furthermore, prior discussions can easily be resumed if e. g. new requirements have to be taken into account.

¹ <http://www.neon-project.org/>. This work has been supported by the European project *Lifecycle Support for Networked Ontologies* (NeOn, IST-2006-027595).

The first objective is accomplished by the Cicero tool itself.² Its underlying argumentation model and discussion workflow are described in section 3 and 4. The second objective is accomplished by integrating Cicero with an ontology editor, which is described in section 5.

2 Use Case

Examples for collaborative ontology engineering are the development processes of the AGROVOC thesaurus³ or the Gene Ontology⁴. They are both maintained by teams of ontology engineers and domain experts that are coordinated by a central organization like FAO or the Gene Ontology Consortium respectively.

The world-wide users of the publicly available ontologies can influence their further development by e.g. proposing new terms or definitions for inclusion or by suggesting the reorganization of sections of the ontologies. The issues are then discussed between the members of the maintenance team. During the discussion, it may be necessary to ask the issue creator for further clarification or for commenting on the proposed solution(s).

3 Cicero Argumentation Model

The argumentation model, that is underlying the Cicero tool, is based on the DILIGENT argumentation framework [1] and the Potts and Bruns model [2]. They are both extensions of the idea of the Issue Based Information Systems [3, 4]. All these models help in structuring an *issue* or problem and to simultaneously derive possible solutions with the help of discussions.

On the one hand, the Potts and Bruns model extends the IBIS approach to discussions that lead to the creation or change of a concrete artifact (e.g. elements in an ontology). The discussion serves as a connection between the old and the new version of the changed artifact, thus documenting its design rationale or provenance. On the other hand, the DILIGENT argumentation framework amongst others introduces certain argument types to IBIS, that accelerates issue resolution. In [1], it has been shown that participants of a discussion, who mainly used the identified argument types, needed less time for coming to a successful conclusion of the discussion.

The Cicero argumentation model combines the general structure for representing discussions from the DILIGENT argumentation framework with the idea of annotating ontology elements and changes with the corresponding discussions. In Cicero, a discussion always starts with an issue that is raised by either of the participants (i.e. ontology/knowledge engineers or users). Subsequently, solutions are proposed and discussed. After some time, it has to be decided which of the proposed solutions will be implemented in the ontology (see Fig. 1).

² It can be downloaded from <http://isweb.uni-koblenz.de/Research/Cicero/>.

³ <http://www.fao.org/agrovoc/>

⁴ <http://www.geneontology.org/>

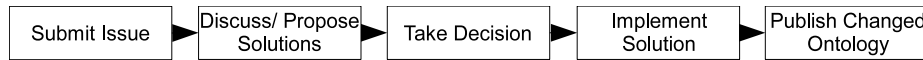


Fig. 1. Prototypical workflow for discussions about the design rationale.

4 Using the Model in Cicero

Cicero is an extension of the Semantic MediaWiki. On the one hand, this has several advantages: For example, one can also use the Wiki for maintaining documents related to the collaborative ontology engineering, e. g. requirements documents. Furthermore, the Semantic MediaWiki allows for accessing its contents as RDF streams which is useful for integrating it with other tools like the NeOn toolkit. But on the other hand, it has to be extended with functionality that supports the discussion workflow presented in the previous section.

4.1 Discussions

In Cicero exists for each discussion thread an overview page summarizing the issue and all proposed solutions (see Fig. 2). The page is automatically generated as soon as a new issue is created. The actual discussion consisting of solution proposals and arguments then takes place on the talk page of the overview page.

The screenshot shows the Cicero interface for an issue. The main content area is titled "Prj:AGROVOC/Issue:Broader Term of Geographical Region". It includes a description of the issue, a list of solution proposals, and a table of issue properties.

Issue Overview

Created by:	User:Jmonte
Created at:	2007-10-05 09:06:44
Issue state:	running (until 2007-12-21 17:05:13)
Decision Mode:	preferential
Selection Mode:	single selection

Reactions on this Issue

There are

- + 3 Solution Proposals
- + 1 Arguments

[View Discussion](#)
[Change Issue Properties](#)

All Solution Proposals:

- Solution Proposal 1:** As we need to add a BT for the already existing term Geographical region in our ontology, I propose to introduce 'Region' or 'Regions' as the accordant BT (Geographical region BT 'Region' or 'Regions')
- Solution Proposal 2:** I furthermore propose to add 'Geographic region' as a synonym to Geographical region (Geographic region <synonym> Geographical region) with 'Geographic region' as the non-descriptor and Geographical region as the descriptor
- Solution Proposal 3:** I think we should use both singular and plural forms on these terms depending on if it's countable.
That means:
 - if there are more than one => plural
 - if there is only one => singular

Facts about AGROVOC/Issue:Broader Term of Geographical Region

CreatedBy	User:Jmonte
ProjectAffiliation	Prj:AGROVOC
CreatedAt	09.08.5 October 2007
DecisionMode	preferential
IssueState	running
IssueTimer	30
IssueTitle	Broader Term of Geographical Region
SelectionMode	single selection
VotingTimer	30

Fig. 2. Overview page of an issue in Cicero.

During the discussion, a user can propose solutions of the issue. The different solution proposals can then be supported or objected by all discussion participants. Three different argument types are supported by Cicero, that have been identified in [1] as accelerating issue resolution.:

- **Example:** An supporting or objecting example corresponds to a pattern that should or should not be imitated. Examples are used for illustrating similar cases that may serve as a model for the solution to which they reply.
- **Evaluation:** An evaluation gives criteria which help to assess the strengths and weaknesses of a solution proposal.
- **Justification:** A justification describes the relevant circumstances that help to understand why a certain solution is supported or objected by a user.

4.2 Taking a Decision

After some time of discussion, the decision taking procedure may be started in Cicero. During decision taking, no new solution proposals or arguments can be added to the discussion and users with the corresponding access right can cast their ballot. It depends on the settings of an issue whether users can cast their ballot for one solution proposal only or for several of the solution proposals.

Two decision taking modes can be distinguished: During *preferential voting mode*, all users with the corresponding access right can cast their ballot. The solution proposal with the most votes is subsequently marked as the decided solution. During *dictator mode*, only a single user with the corresponding access right decides on the solution of the issue.

5 Integration with NeOn Toolkit

The NeOn toolkit⁵ is an ontology engineering environment that can be easily extended. Currently, a plugin is being developed that integrates the functionality of Cicero into the NeOn toolkit.⁶ The plugin will allow for an easy creation of new discussions and for searching existing discussions that are related to specific ontology entities.

For example, if a user of the NeOn toolkit wants to create a new discussion he may select the ontology entities in the environment that should be discussed, i. e. the issue gets related to ontology entities. Before actually creating the new issue, a list of already existing issues related to the selected ontology entities can then be presented to the user. Thus, he can check whether it is really necessary to create a new issue or whether it is better to participate in the discussion of one of the already existing issues.

After a decision is taken, the NeOn toolkit is used for implementing the selected solution proposal. All changed, deleted or added ontology entities get also related to the discussion, i. e. a provenance relationship between ontology entities and their design rationale in form of the discussions is established.

⁵ <http://www.neon-toolkit.org/>

⁶ <http://www.neon-toolkit.org/wiki/index.php/Cicero>.

6 Conclusions and Related Work

Aspects of Cicero and its integration with the NeOn toolkit can be compared with Collaborative Protégé [5], Tadzebao and WebOnto [6] as well as Compendium [7, 8]. Because of lack of space, the comparison in Tab. [6] is concentrated on the aspects of Cicero highlighted in this paper.

	Collaborative Protégé	Tadzebao & WebOnto	Compendium	Cicero
IBIS-based Arg. Framework	no	no	yes	yes
Integration with Ontology Editor	yes	yes	no	yes
Establishing Provenance Links	yes	no	no	yes

Table 1. Comparison of Cicero with related work

Altogether, Cicero helps with its IBIS-based argumentation framework to have better structured and more efficient discussions. Furthermore, the integration with an ontology editor leads to a better support of the ontology engineering lifecycle in which discussions about the design rationale of ontology elements play an inherent role. Finally, Cicero and its integration with the NeOn toolkit reduces the required effort for establishing the provenance links between design rationale discussions and the affected ontology elements.

References

1. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. In: Proc. of ECAI. (2004)
2. Potts, C., Bruns, G.: Recording the reasons for design decisions. In: ICSE. (1988) 418–427
3. Kunz, W., Rittel, H.: Issues as elements of information systems. WP 131, Institute of Urban and Regional Development, University of California, Berkeley (1970)
4. Rittel, H.W.J., Webber, M.M.: Dilemmas in a general theory of planning. *Policy Sciences* 4(2) (June 1973) 155–169
5. Tudorache, T., Noy, N.: Collaborative Protégé. In: Social and Collaborative Construction of Structured Knowledge Workshop held at WWW 2007. (2007)
6. Domingue, J.: Tadzebao and WebOnto: Discussing, browsing and editing ontologies on the web. In: 11th Knowledge Acquisition Workshop. (1998)
7. Buckingham Shum, S., Selvin, A., Sierhuis, M., Conklin, J., Haley, C., Nuseibeh, B.: Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC. In Dutoit, A., McCall, R., Mistrk, I., Paech, B., eds.: *Rationale Management in Software Engineering*. Springer (2006) 111–132
8. Buckingham Shum, S., Motta, E., Domingue, J.: Augmenting design deliberation with compendium: The case of collaborative ontology design. In: Workshop on Facilitating Hypertext-Augmented Collaborative Modeling at the ACM Hypertext Conference. (2002)