Leveraging Semantic Data Wikis for Distributed Requirements Elicitation

Thomas Riechert[1] and Thorsten Berger[1,2]

[1]Chair of Business Information Systems
University of Leipzig, Germany
riechert@informatik.uni-leipzig.de

[2]Generative Software Development Lab
University of Waterloo, Canada
tberger@swen.uwaterloo.ca

Abstract

Using Wikis for the collaborative creation of structured textual content has gained increasing importance in the past decade. As Wikis facilitate the involvement of large user groups to create content in an easy way, their application in large, spatially distributed software development efforts seems to be very promising. In this context, we present a classification of Wiki-based approaches to Requirements Engineering (RE) and discuss their suitability. Next, we introduce the ontology of an approach that aims at supporting the collaboration of stakeholders with regard to the RE process. This approach enables large stakeholder groups to elicit, semantically structure and classify requirements in the very early and creative RE phases.

Instead of leveraging text-based Wikis, the approach is based on our semantic data Wiki OntoWiki, which focuses on the structuring and management of fine-grained data by employing Semantic Web technologies. OntoWiki enables the intuitive authoring of Knowledge Bases and facilitates the application of Knowledge Management methods.

1. Introduction

The successful accomplishment of the Requirements Engineering (RE) process is considered a major success factor in almost all software development projects. Each of the stakeholders involved defines goals, scenarios and requirements for the software product under the supervision of a requirements analyst. Thereafter, the elicited requirements have to be documented, structured, analyzed and probably refined in order to be utilizable for implementing the product. These activities are supported by a wide range of comprehensive Requirements Management (RM) tools. Well-known ones are, for example, IRQA, Doors, RequisitePro, MKS or CaliberRM. In smaller or agile projects, however, we can frequently observe a tradeoff between the application of a heavyweight RE process with the RM tools mentioned and a lightweight approach using general-purpose documentation tools. Especially text-based Wikis have been successfully applied for documenting requirements, as they facilitate the involvement of non-technical users in an easy way.

However, in large projects with a significant number of stakeholders involved, it is crucial not only to jointly elicit, but also to structure all requirements in an efficient way. This is even more important when the project is spatially distributed. The capability to adequately structure, analyze and collaboratively refine these requirements becomes a key success factor in the early phases of such projects (cf. Globalization of RE in [1]). Although some RM tools already allow multiple users to elicit requirements by providing clients with checkin/checkout-style access to a repository with versioning support, these clients focus on the requirements architect and are unlikely to scale for large user groups.

In this paper, we first present a classification of Wiki approaches for RE and, subsequently, the ontology of our approach to eliciting requirements by leveraging a semantic data Wiki. In doing so, we take recourse to our earlier work in the field of semantic-based RE. In particular, we build upon an earlier version of the ontology [2] and outline various improvements that have been made toward a better alignment with existing domain-specific vocabularies in order to foster requirements reuse and shared understanding between stakeholders.

Accordingly, the remainder of the paper is structured as follows: In section 2, we elaborate on a classification of various Wiki approaches for the RE process. In section 3, we introduce OntoWiki, emphasizing the specific semantic capabilities used. In section 4, we present SWORE, the extensible ontology for requirements structuring and its realization in

OntoWiki. Thereafter, we discuss some related work in section 5. Finally, in section 6, we conclude our contribution by discussing some early experiences and giving an outlook on our future work.

2. Classification of Wiki Approaches to Requirements Engineering

Wikis can be characterized as easy-to-use and open authoring environments for the creation and maintenance of textual content. As it has already been discussed in the literature [3-5], they are also suitable for the elicitation and documentation of requirements in many software development efforts. By facilitating collaborative content creation, they are especially useful in projects where numerous stakeholders are involved in the RE process.

However, just using plain Wikis for the documentation of requirements unveils many limitations. First, we cannot adequately model semantic relations between requirements, such as conflicts or various kinds of dependencies (“details”, “entails”, etc.). Second, we cannot derive individual views on the requirements by incorporating semantic filters. Wikis usually only provide a consolidated overall view on the text base, which is independent from the current author. Individual views are rather difficult to establish, for example by imposing personal tagging. Third, more sophisticated functionalities of RM tools such as change management, baselining or branching are not supported.

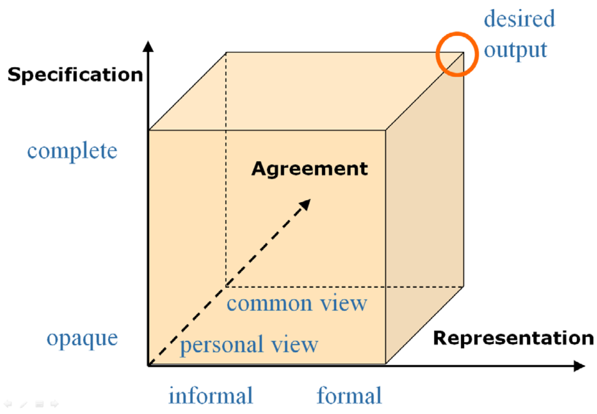


Figure 1. Three dimensions of RE [6]

As some of these limitations stem from the lack of semantic capabilities in traditional Wikis, the usage of semantic Wikis has been proposed in the literature. Most notably, *Semantic Mediawiki* was adapted for RE purposes [4,5] that deal with document-grained textual content. Typically, specific templates were developed in order to guide the requirements elicitation. However, without further discussing the advantages of these approaches, we argue that more formal and comprehensive structuring facilities for fine-grained requirements are needed. Such structuring facilities should expose more explicitly the requirements semantics, should conform to a mature semantic schema and should also be extensible with standard vocabularies in order to capture metadata or additional domain-specific knowledge.

We therefore examine a third kind of Wikis: semantic data-based Wikis which focus on managing fine-grained (but still textual) data. Such Wikis are usually based on *RDF Triple Stores* and directly represent its content in an ontology based on RDF triples.

In summary, we distinguish three different types of Wikis applicable for RE purposes:

a) **Classical document**-based Wikis (e.g. MediaWiki, DokuWiki);

b) **Semantic document**-based Wikis (e.g. Semantic MediaWiki, IkeWiki);

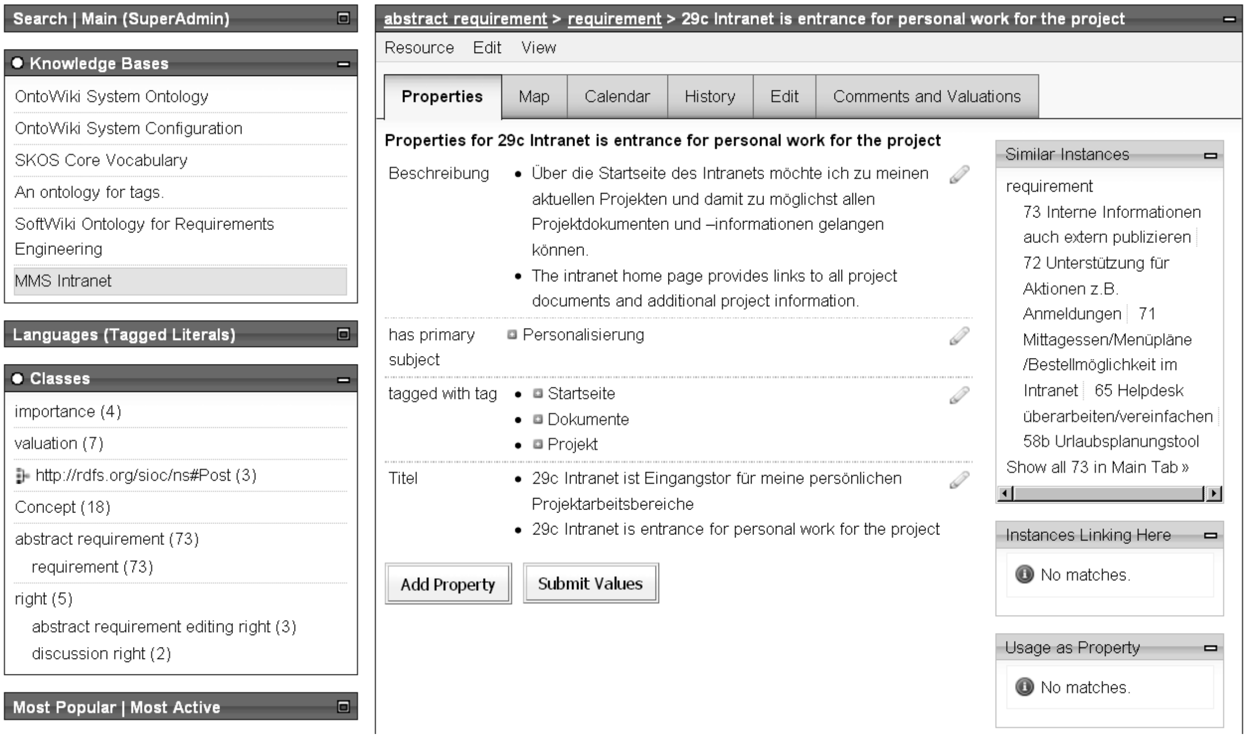
c) **Semantic data**-based Wikis (e.g. OntoWiki, Freebase.com).

In order to classify these three types of Wikis according to their suitability for managing requirements, we take Pohl’s characterization of the RE process from [6]. Pohl identifies three major dimensions of RE: *specification*, *agreement* and *representation*, see Figure 1. The *specification* dimension denotes the “degree of requirements understanding at a given time” and refers to the improvement of requirements by repeated analysis and refinement. The *agreement* dimension deals with “the degree of agreement reached on a specification” and refers to finding a consensus among the stakeholders. Last, the *representation* dimension copes with the different levels of documentation and ranges from informal descriptions (or illustrations) to more formal versions.

Table 1. Classification of Wiki types

	Specification	Agreement	Representation
a)	1	2	4
b)	1	3	5
c)	1	3	6

1 between vague and complete
2 consolidated view only
3 both individual and consolidated view
4 informal
5 according to templates
6 according to ontology model


Search | Main (SuperAdmin)
Knowledge Bases
OntoWiki System Ontology
OntoWiki System Configuration
SKOS Core Vocabulary
An ontology for tags.
SoftWiki Ontology for Requirements Engineering
MMS Intranet
Languages (Tagged Literals)
Classes
importance (4)
valuation (7)
http://rdfs.org/sioc/ns#Post (3)
Concept (18)
abstract requirement (73)
requirement (73)
right (5)
abstract requirement editing right (3)
discussion right (2)
Most Popular | Most Active
abstract requirement > requirement > 29c Intranet is entrance for personal work for the project
Resource
Edit
View
Properties
Map
Calendar
History
Edit
Comments and Valuations
Similar Instances
requirement
73 Interne Informationen auch extern publizieren
72 Unterstützung für Aktionen z.B. Anmeldungen
71 Mittagessen/Menüpläne /Bestellmöglichkeit im Intranet
65 Helpdesk überarbeiten/vereinfachen
58b Urlaubsplanungstool
Show all 73 in Main Tab »
Instances Linking Here
No matches.
Usage as Property
No matches.
Add Property
Submit Values

Properties for 29c Intranet is entrance for personal work for the project

Beschreibung	Über die Startseite des Intranets möchte ich zu meinen aktuellen Projekten und damit zu möglichst allen Projektdokumenten und –informationen gelangen können. The intranet home page provides links to all project documents and additional project information.
has primary subject	Personalisierung
tagged with tag	Startseite Dokumente Projekt
Titel	29c Intranet ist Eingangstor für meine persönlichen Projektarbeitsbereiche 29c Intranet is entrance for personal work for the project

Figure 2. OntoWiki screenshot showing a requirements instance

In the following, we elaborate on the classification of the three Wiki types by discussing their coordinates in Pohl's RE dimensions. Table 1 contains the result of our classification.

Specification: Although it is very difficult to determine the completeness of elicited requirements, all three Wiki types facilitate at least content acquisition from many stakeholders in an easy way. Keeping in mind that active stakeholder participation increases requirements quality [3] and that Wikis follow the authoring philosophy of "making it easy to fix mistakes instead of making it hard to make them" [7], Wikis can foster the acquisition of a more complete set of requirements.

Agreement: All three Wiki types provide a consolidated general view on all documented requirements. However, with a large number of stakeholders involved, it is inevitable to define individual views. Semantic Wikis allow to define such views by supporting semantic filtering or by exposing semantic query interfaces like SPARQL[1] in OntoWiki or inline queries with #ask[2] in Semantic MediaWiki.

Of course, a prerequisite for view derivation is the definition of a dynamic classification scheme. The most basic form can be the application of personal tagging as known from the Folksonomies concept [8]. Furthermore, this concept fosters the utilization of a consensus terminology. In combination with standard Wiki features like discussions and annotations, we can assume an increase in the agreement about all requirements.

Representation: Structuring requirements in classical and document-based semantic Wikis is primarily accomplished with proprietary templates which define a basic editing schema. In contrast, semantic data-based Wikis facilitate the usage of ontology schema languages like RDFS or OWL to define a more expressive and formal structure. Such an approach is a prerequisite for more sophisticated knowledge management techniques in RE and allows consistency checks via reasoning capabilities. Furthermore, creating links between requirements and instances of other standardized Semantic Web vocabularies such as SIOC[3] (Semantically-Interlinked Online Communities), FOAF[4] (Friend of a Friend) or SKOS[5] (Simple Knowledge Organization System) becomes possible.

[1] http://www.w3.org/TR/rdf-sparql-query/
[2] http://semantic-mediawiki.org/wiki/Help:Inline_queries
[3] http://sioc-project.org/
[4] http://www.foaf-project.org/
[5] http://www.w3.org/2004/02/skos/

In summary, we can conclude that semantic data-based Wikis bear the potential to support RE in all three dimensions of Pohl's characterization. However, it is obvious that Wikis lack the comprehensive authoring workflows and analysis processes found full-featured RM tools. Instead, semantic data-based Wikis should be used in the very early and creative phases in RE and should provide semantically enriched content that is to be further managed in RM tools.

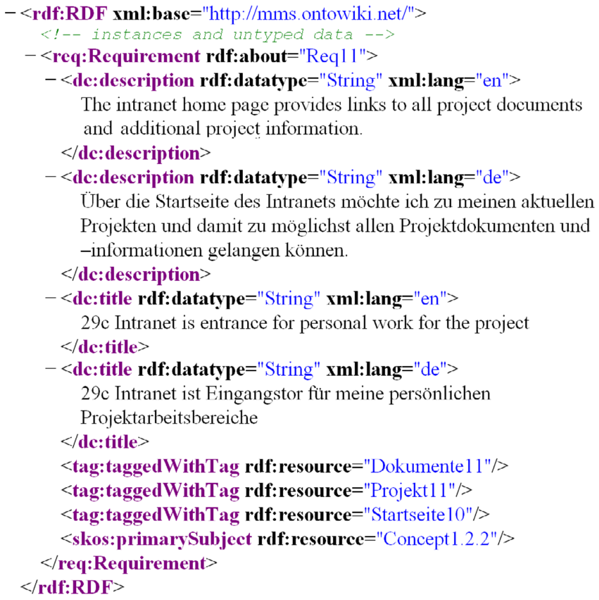
3. The Semantic Data Wiki OntoWiki

One of the main interests of our research group comprises the field of Knowledge Management in combination with Semantic Web technologies. In this context, we have been developing the tool OntoWiki [9] which is focused on supporting distributed knowledge acquisition scenarios by taking advantage of semantic Wiki principles. OntoWiki provides knowledge base authoring and management features which directly operate on ontologies that are based on RDF triples. These authored ontologies can be defined via the schema description languages RDFS and OWL. Furthermore, OntoWiki provides a plugin extension mechanism that can be used to add domain-specific views or user interfaces for knowledge bases (like the SoftWiki plugin to be discussed later).

The core functionality of OntoWiki comprises:

- *intuitive authoring and visualization* of knowledge bases with internationalization support;
- *versioning facilities* to track, review and selectively roll-back changes;
- *community support* for discussing and annotating information chunks (literal data);
- *searching* in knowledge bases with filtering and sorting support (using semantic relations);
- *rendering of class hierarchies* and presentation of the related instance data;
- *access control* on the ontology level;
- *inline-editing* of property values and relations of instances;
- *predefined views* on the knowledge base such as Properties, Map, Calendar, History, etc. (cf. Figure 2).

We present a screenshot of OntoWiki's web interface in Figure 2. It shows the properties view of an instance of the class `requirement`. The menus on the left hand side are used for navigating the knowledge bases and their containing classes or instances. As we can further see in the screenshot, an ontology for requirements labeled "MMS Intranet" has been selected in the "Knowledge Bases" box. Beneath, the "Classes" box reveals the class hierarchy as well as the number of related instances to each class. The shown instance of the class `requirement` labeled "29c Intranet is entrance for personal work for the project", was selected in a list of instances not shown by the screenshot.



```
– <rdf:RDF xml:base="http://mms.ontowiki.net/">
    <!-- instances and untyped data -->
  – <req:Requirement rdf:about="Req11">
    – <dc:description rdf:datatype="String" xml:lang="en">
        The intranet home page provides links to all project documents
        and additional project information.
      </dc:description>
    – <dc:description rdf:datatype="String" xml:lang="de">
        Über die Startseite des Intranets möchte ich zu meinen aktuellen
        Projekten und damit zu möglichst allen Projektdokumenten und
        –informationen gelangen können.
      </dc:description>
    – <dc:title rdf:datatype="String" xml:lang="en">
        29c Intranet is entrance for personal work for the project
      </dc:title>
    – <dc:title rdf:datatype="String" xml:lang="de">
        29c Intranet ist Eingangstor für meine persönlichen
        Projektarbeitsbereiche
      </dc:title>
      <tag:taggedWithTag rdf:resource="Dokumente11"/>
      <tag:taggedWithTag rdf:resource="Projekt11"/>
      <tag:taggedWithTag rdf:resource="Startseite10"/>
      <skos:primarySubject rdf:resource="Concept1.2.2"/>
    </req:Requirement>
  </rdf:RDF>
```

Figure 3. Sample RDF source of a requirement

The representation of instances is, according to the RDF paradigm, a collection of triples – each consisting of a Subject, a *Predicate* and an *Object*. According to Figure 2, which shows the properties view in the center of the screen, the Subject is the currently selected instance (which is a resource of class `requirement`, in this case). Its properties are denoted by all *Predicates* that occur in triples with the instance as the *Subject*. They are shown on the left hand side of the properties view. The properties' values (i.e. the *Objects* in the related triples) are shown on the right hand side, respectively. If these values are *Literals*, they can be edited directly, but if they refer to another instance (*Resource*), OntoWiki provides navigation facilities for further traversion. Take a look at the plus and the edit signs in the screenshot as well as at Figure 3, which shows the corresponding RDF source for further illustration.

4. Realization

The central element of our approach constitutes the ontology schema *SWORE*[6] – *SoftWiki Ontology for Requirements Engineering*. SoftWiki refers to our current R&D project, within which we developed the ontology schema. Basically, SWORE supplies the semantic structure used to hold the requirements-relevant data and to facilitate links to other domain-specific vocabularies. The full version of SWORE is freely available for download[7].

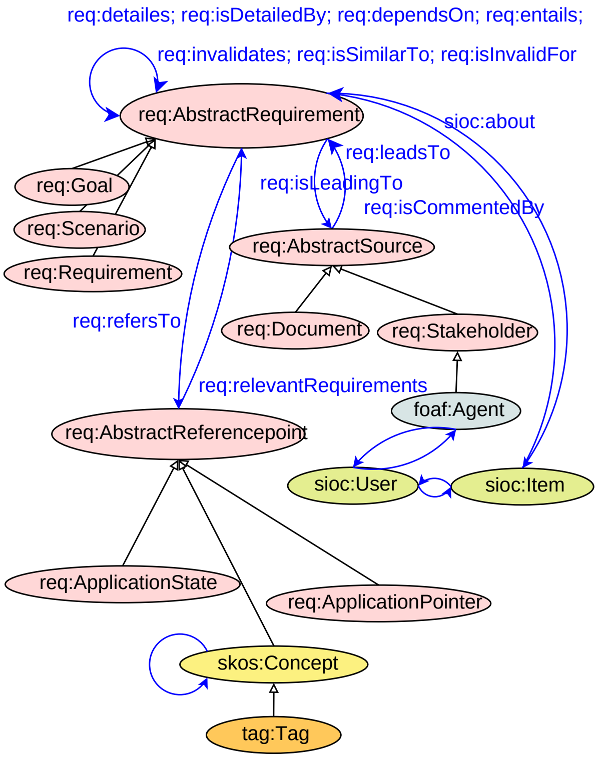


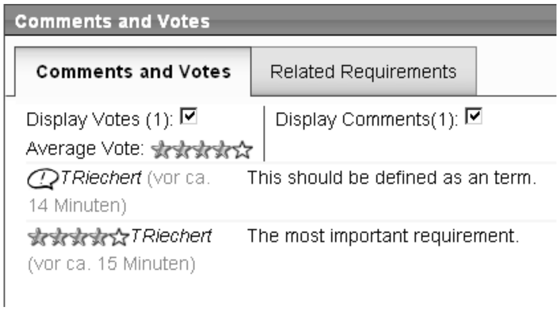
Figure 4. Core of the SoftWiki Ontology for RE

Figure 4 illustrates the core of the SWORE ontology in its most recent version 1.01. Designing the ontology was accompanied by our project partners from the RE domain. Furthermore, the following established RE information models were taken into account: [10,11,2].

The classes `AbstractSource` and `AbstractRequirement` as well as the properties `isDetailedBy` and `leadsTo` are most important for the vocabulary. `AbstractRequirement` has the sub-classes `Goal`, `Scenario` and `Requirement.`

In order to model the actual source of a requirement, we use one of `AbstractSource`'s subclasses: either `Document` (if it stems from a document; such as legals and laws, for example) or `Stakeholder` (if the requirement has been directly defined by a stakeholder). Furthermore, stakeholders are able to detail requirements (`isDetailedBy`) or set them into relation to other requirements by using `dependsOn`, `entails`, `invalidates`, `isSimilarTo` or `isInvalidFor`. These relations enable the specification of requirements in different granularities.

Concerning the collaboration aspects of RE, we put the focus on the integration of stakeholder discussions and the ability to vote for requirements. Both features were realized by integrating the commonly used Semantic Web vocabularies SIOC and FOAF.



Display Votes (1): ☑
Average Vote: ☆☆☆☆☆
Display Comments(1): ☑

TRiechert (vor ca. 14 Minuten) This should be defined as an term.

☆☆☆☆☆ TRiechert (vor ca. 15 Minuten) The most important requirement.

Figure 5. Comments and votes

The facility to interlink requirements with later phases of the software engineering process [12] is realized by the property `refersTo` and the class `AbstractReferencePoint`. Such reference points are structured via the vocabulary SKOS and basic tagging.

Finally, with the designed SWORE, we extended OntoWiki by implementing the SoftWiki plugin that provides a custom user interface for stakeholders. Figure 5 shows a part of the generic user interface realized for discussion and voting. In order to make the whole SoftWiki tool available for a broader stakeholder community, we are currently integrating it into the open source development platform *Cofundos.org*. Starting in April 2009, we will support Requirements Elicitation for some selected open source projects.

Nonetheless, we have already prototypically evaluated the approach in smaller (educational) software development projects and the results were very promising. However, the small-scale evaluations also revealed some obstacles that we will have to deal with in the near future. One issue was the performance and responsiveness of the tool, which seems to be owed to the higher overhead of the semantic

[6] http://ns.softwiki.de/req/
[7] http://softwiki.de/SWORE/100

technologies used. The higher abstraction and, thus, the additional mapping of RDF-based data onto more efficient persistence mechanisms (i.e. relational databases) decrease the overall performance. Likewise, the reasoning and consistency checking activities seem to imply higher response times.

5. Related Work

As already mentioned in section 2, Wikis have been successfully applied to foster knowledge sharing in software engineering efforts. This knowledge ranges from domain knowledge such as requirements, legals and laws or specifications to very detailed implementation knowledge such as architectural models, technical documentations or even test cases. When focusing directly on the application of Wikis in Requirements Engineering, we primarily find position or vision papers.

There seems to be consensus that one of the major aims of Wikis in RE is the incorporation of (non-technical) stakeholders. For example, Decker et al. [3] elaborate on the applicability of text-based Wikis in such setups and propose a meta-model for requirements structuring. In another publication, Decker et al. explicitly discuss the application of semantic Wikis and sketch their vision of "Wikitology" [13].

Furthermore, Ferreira and Da Silva [4] propose the usage of semantic Wikis in combination with Natural Language Processing (NLP) techniques in order to elicit and validate requirements in natural language. Abeti et al. [5] leverage the Semantic MediaWiki in order to capture several kinds of requirements for business processes and to facilitate discussions about them. The elicited requirements in their WikiReq tool can also be exported to Eclipse EMF for visualization purposes.

6. Conclusions and Future Work

In this paper, we elaborated and discussed a classification of Wiki approaches to RE. We also presented an improved version of the core ontology of our approach to support the requirements elicitation, structuring and documentation process by leveraging a semantic data Wiki. The major improvement was the alignment of the ontology with existing domain-specific vocabularies such as SKOS, SIOC and FOAF. Our approach to RE can be characterized especially by the following three points:

- We focus on a semantic data-based Wiki with comprehensive fine-grained structuring and reasoning capabilities and do not rely on document templates.
- We foster the extensibility of the ontology schema and enable the interlinking of requirements with other domain data as described in commonly used Semantic Web vocabularies. Linking requirements to domain knowledge further sharpens the semantics of requirements.
- We do not compete with the functionality of full-featured RM tools; instead we focus on the very early and creative RE phases.

Concerning the last point, we have developed an export filter that is able to translate our RDF-based requirements into the Requirements Interchange Format [14]. As such, the requirements can be imported into common RM tools in turn.

Since the results of the first prototypical applications of the approach in smaller projects are very promising, we now target on adopting the tool in larger projects. One example will be the adoption of the tool for the aforementioned open source platform *Cofundos.org*.

Furthermore, since one of the advantages of the approach is the ability to link requirements with further domain knowledge and since another one of our research directions is the Product Line Engineering, we will also investigate on the integration of a variability ontology into the SoftWiki tool.

In summary, we can conclude that semantic data-based Wikis bear the ability to improve the agreement between stakeholders by improving the domain understanding and the successive creation of a shared terminology.

7. Acknowledgements

We wish to thank our project partners in the national R&D project SoftWiki[8], which is funded by the German Federal Ministry of Education and Research.

8. References

[1] B.H.C. Cheng and J.M. Atlee, "Research Directions in Requirements Engineering," *2007 Future of Software Engineering*, IEEE Computer Society, 2007, pp. 285-303.

[2] A.V. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," *Proceedings of the 5th*

[8] http://www.softwiki.de

IEEE International Symposium on Requirements Engineering, IEEE Computer Society, 2001, p. 249.

[3] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, “Wiki-Based Stakeholder Participation in Requirements Engineering,” *IEEE Software*, vol. 24, 2007, pp. 28-35.

[4] D. Ferreira and A.R. da Silva, “Wiki Supported Collaborative Requirements Engineering,” *Wikis for Software Engineering (Wikis4SE) Workshop at Wiki Symposion (WikiSym'08),* Orlando, Florida: 2008.

[5] L. Abeti, P. Ciancarini, and R. Moretti, “Wiki-based Requirements Management for Business Process Reengineering,” *Wikis for Software Engineering (Wikis4SE) Workshop at Wiki Symposion (WikiSym'08)*, Orlando, Florida: 2008.

[6] K. Pohl, “The three dimensions of requirements engineering,” *Advanced Information Systems Engineering*, 1993, pp. 275-292.

[7] B. Leuf and W. Cunningham, *The Wiki way: quick collaboration on the Web*, Addison-Wesley Longman Publishing Co., Inc., 2001.

[8] L. Specia and E. Motta, “Integrating Folksonomies with the Semantic Web,” *The Semantic Web: Research and Applications*, Springer, 2007, pp. 624-639.

[9] S. Auer, S. Dietzold, and T. Riechert, “OntoWiki – A Tool for Social, Semantic Collaboration,” *The Semantic Web - ISWC 2006*, 2006, pp. 736-749.

[10] K. Pohl, *Process-Centered Requirements Engineering*, John Wiley & Sons, Inc., 1996.

[11] P. Haumer, K. Pohl, and K. Weidenhaupt, “Requirements Elicitation and Validation with Real World Scenes,” *IEEE Trans. Softw. Eng.*, vol. 24, 1998, pp. 1036-1054.

[12] T. Riechert and S. Lohmann, “Mapping Cognitive Models to Social Semantic Spaces–Collaborative Development of Project Ontologies,” 2007.

[13] B. Klein, C. Hoecht, and B. Decker, “Beyond Capturing and Maintaining Software Engineering Knowledge—“Wikitology” as Shared Semantics,” *Workshop on Knowledge Engineering and Software Engineering at the Conference on Artificial Intelligence*, Koblenz: 2005.

[14] Automotive HIS, “Requirements Interchange Format, version 1.1a,” *http://www.automotive-his.de/rif/doku.php (February 16, 2009)*, 2007.