ReqWiki: A Semantic System for Collaborative Software Requirements Engineering

Bahar Sateli
Semantic Software Lab
Department of Computer Science and Software Engineering.
Concordia University,
Montréal, QC, Canada
sateli@semanticsoftware.info

Srinivasan Sembakkam Rajivelu
Department of Computer Science and Software Engineering.
Concordia University,
Montréal, QC, Canada
s_sembak@encs.concordia.ca

Elian Angius
Semantic Software Lab
Department of Computer Science and Software Engineering.
Concordia University,
Montréal, QC, Canada
e_angiu@encs.concordia.ca

René Witte
Semantic Software Lab
Department of Computer Science and Software Engineering.
Concordia University,
Montréal, QC, Canada
witte@semanticsoftware.info

Copyright is held by the author/owner(s).
WikiSym '12, Aug 27–29, 2012, Linz, Austria.
ACM 978-1-4503-1605-7/12/08.

Abstract
The requirements engineering phase within a software project is a heavily knowledge-driven, collaborative process that typically involves the analysis and creation of a large number of textual artifacts. We know that requirements engineering has a large impact on the success of a project, yet sophisticated tool support, especially for small to mid-size enterprises, is still lacking. We present *ReqWiki*, a novel open source web-based approach based on a semantic wiki that includes Natural Language Processing (NLP) *assistants*, which work collaboratively with humans on the requirements specification documents.

Author Keywords
Software Requirements Engineering; Quality Assurance; Verification and Validation; Natural Language Processing; Semantic Wiki

ACM Classification Keywords
D.2.1 [Software Engineering]: Requirements/Specifications; H.3.1 [Content Analysis and Indexing]: Abstracting methods, Indexing methods, Linguistic processing; H.5.2 [User Interfaces]: Natural language, User-centered design; I.2.1 [Applications and Expert Systems]: Natural language interfaces; I.2.7 [Natural Language Processing]: Text analysis

Motivation

Requirements Engineering (RE) is one of the most important phases of a software project: We know that the success or failure is highly dependent on successful requirements engineering [11]. Natural language (NL) specifications continue to be the most commonly used form (as opposed to formal models, based on a logical framework), accounting for up to 90% of all specifications [6]. However, NL specifications are prone to a number of errors and flaws, in particular due to the ambiguity inherent in natural language [11].

The cost of finding and fixing these defects increases from each engineering stage to the next (requirements, design, implementation, testing, deployment, maintenance) [5]. In ReqWiki, this is addressed by requirements quality assurance (QA), also known as requirements validation [11]. The goal is to verify the requirements artifacts produced during elicitation, evaluation, and specification, both in terms of quality and objectives.

Introduction

Software requirements engineering (RE) encompasses the tasks related to capturing, determining and recording the needs of various stakeholders of a software project. Wikis, as an affordable and lightweight documentation and distributed collaboration platform, have demonstrated their capabilities in distributed requirements elicitation [2] and documentation [10]. While the integrity of software requirements specifications (SRS) documents inside a wiki can be enforced through the use of templates and wiki bots on a syntactic level, semantic SRS defects, such as ambiguity or inconsistency, require human revision. We show how a combination of various state-of-the-art techniques from the Natural Language Processing (NLP) and Semantic Computing domains can help to improve the quality of SRS documents contained in a wiki platform, by providing guidance and automatically detecting SRS defects.

The developed ideas have been implemented in *ReqWiki*[1] – a novel semantic wiki system that is customized for requirements engineering purposes. It is the first open source RE tool that combines wiki technology for collaborative use and semantic knowledge representation for formal queries and reasoning with natural language processing assistants that work collaboratively together with the human users, within a single, cohesive interface. Based entirely on open source and open standards, ReqWiki is targeted for Use Case-driven requirements engineering [5], following the Unified Process (UP) methodology [3].

Application

The core of ReqWiki is a Semantic MediaWiki engine, shown in Figure 1, which provides a collaborative environment for SRS development. However, unadorned wikis can not fulfil all requirements, such as consistency management, semantic support, and natural language processing in the context of RE. For this, we added two substantial features to the wiki engine underlying ReqWiki, detailed below: a semantic data model and NLP web services.

Semantic Model

To ensure consistency and compliance with a standard methodology, both within and between requirements artifacts, we formally model our system entities and their relationships by defining an ontology written in OWL[2] and use the ReqWiki semantic wiki engine to connect textual descriptions with this formal ontology, thereby providing further semantic reasoning and querying capabilities. This ontology provides for a semantic description of the concepts stored in the wiki (e.g., actors, goals, use cases, test cases, features), as well as the relationships between them. The ReqWiki *Semantic Forms*[3] used to add content to the wiki automatically connect these artifacts with the ontology. We leverage this ontology in ReqWiki to support the automatic creation of three forms of traceability [11] links: semantic links inside the wiki, query-based links, and revision links.

For the first type, semantic concepts defined in wiki pages are presented as hyperlinks, so that users can navigate to their corresponding page and view its content (Figure 2). When defining these semantic concepts, the forms themselves already constrain field input to non-empty existing ontology instances of the expected field type. Thus, this form of traceability enforces link correctness

[1] ReqWiki, http://www.semanticsoftware.info/reqwiki

[2] Web Ontology Language (OWL), http://www.w3.org/2004/OWL/

[3] Semantic Forms, http://www.mediawiki.org/wiki/Extension:Semantic_Forms

and simplifies verification and validation phases, since dead links and unlinkable artifacts are detectable through special wiki pages.

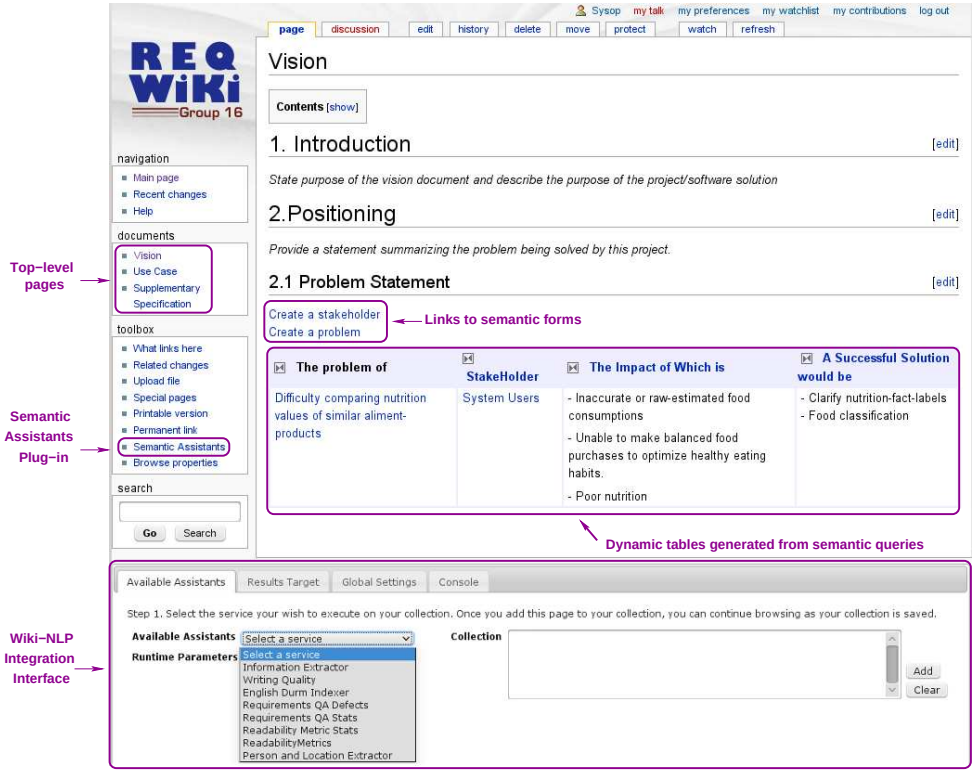

Figure 1: The ReqWiki User Interface

For the second type, using the same semantic metadata in our ontology, we can insert in-line queries in wiki pages, e.g., the top-level pages (Figure 1), to create dynamic tables from the ontological metadata, such as the Need-Feature [5] table shown in Figure 2.

A third form of traceability available in ReqWiki is the revision control capability, recording changes to pages. This not only indicates which author changed what content at which times, but also allows reverting to previous revisions in case of erroneous modifications.

```
1 {{#ask:
2 [[Category:Features]]
3 | ?BelongsTo=Need
4 | ? = Feature
5 | format= table
6 }}
```

User Needs versus Features

Need	Feature
Modify policy detail information	Alter policy information
Modify policy detail information	Query the status of policy information
Alteration of unit link product	Input conversion to unit investment
Alteration of unit link product	Modify conversion to unit investment
Alteration of unit link product	Query unit investment
Alteration of unit link product	Query unit price

Figure 2: Automatic traceability links in ReqWiki: Semantic MediaWiki query and resulting Need-Feature Table

NLP Services for RE

The second major component of our ReqWiki system is the idea of semantic NLP *assistants* that support users in developing the requirements specification. Our goal is to offer these analysis capabilities directly within the wiki, to avoid forcing users to switch to a different text analysis application. This is achieved through our Wiki-NLP integration [9]. This integration provides a service-oriented architecture, where new analysis services can be deployed and dynamically discovered by the users. This allows to setup an installation with custom NLP services, e.g., for domain-specific analysis pipelines or custom (company/organization-specific) quality rules. The integration is based on our Semantic Assistants framework, which can broker any deployed NLP pipeline as a standard web service [12]. The semantic NLP services provided by the Semantic Assistants architecture also aid users in developing the SRS by automatically extracting metadata from wiki content and embedding it in pages, as shown in Figure 3. Within our initial experiments, we made a number of both general and

Example NLP Services in ReqWiki

Writing Quality Assessment is a service that integrates the After The Deadline [7] tool and helps ReqWiki users find spelling and grammatical mistakes.

Readability Metrics is a service that provides an overall readability score of the content that users have produced, indicating how hard to read their text is for other stakeholders.

Requirements Quality Assurance is a service developed based on the NASA requirements quality metrics [4] that detects SRS defects like *Options, Directives* or *Weak Phrases* in a document.

Document Indexer creates a back-of-the-book style index of the wiki content and stores it in the wiki as a page.

Information Extractor automatically extracts named entities, such as persons, organizations or locations, from wiki pages, based on the ANNIE system [1]. This service is especially useful in analysing existing domain documents during the requirements analysis phase.

requirements-specific NLP services available to ReqWiki users, as shown in the box on the left.

Pre-Conditions	The manager must be identified and authenticated in the application
Success end condition	The tasks is created and assigned to the technicians with status Assigned.

Readability Metrics on UC/Manage_Tasks (View)

Content	Type	Start	End	Features
The tasks is created and assigned to the technicians with status Assigned.	Passive Voice	686	760	▪ The sentence has been detected as passive and can be improved by changing the verb phrase

Writing Quality on UC/Manage_Tasks (View)

Content	Type	Start	End	Features
The tasks is	Grammar	686	698	▪ problem: Wrong Auxiliary Verb ▪ suggestion: The task is

Figure 3: Annotations generated by different NLP services, stored inside a ReqWiki page

Summary

Requirements engineering is one of the most important phases in the software development process, with more than 50% of failed projects attributed to poor RE. Despite these facts, many projects, in particular in small and mid-size companies, still rely on office tools without built-in support for RE. In this research work, we showed how modern semantic techniques can be combined with NLP services in a collaborative web-based wiki platform to improve RE for all involved stakeholders.

We have conducted a user study on the usability and effectiveness of ReqWiki [8]. The results corroborate our hypotheses that NLP support significantly improves a SRS and that users do not require prior background knowledge in NLP in order to make use of sophisticated semantic support, provided that it is offered through an intuitive interface, like the one we propose here.

References

[1] H. Cunningham et al. *Text Processing with GATE (Version 6)*. University of Sheffield, Department of Computer Science, 2011.

[2] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth. Wiki-Based stakeholder participation in requirements engineering. *IEEE Software*, 24(2):28–35, 2007.

[3] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, 3rd edition, 2003.

[4] P. A. Laplante. *Requirements Engineering for Software and Systems*. Auerbach Publications, 1st edition, 2009.

[5] D. Leffingwell and D. Widrig. *Managing Software Requirements: A Use Case Approach*. Pearson Education, 2nd edition, 2003.

[6] M. Luisa, F. Mariangela, and N. Pierluigi. Market research for requirements analysis using linguistic tools. *Requirements Engineering*, 9:40–56, 2004.

[7] R. Mudge. The Design of a Proofreading Software Service. In *Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids, CL&W*, 2010.

[8] B. Sateli. A General Architecture to Enhance Wiki Systems with Natural Language Processing Techniques. Master's thesis, Concordia University, Montréal, QC, 2012.

[9] B. Sateli and R. Witte. Natural Language Processing for MediaWiki: The Semantic Assistants Approach. In *Proceedings of the 8th International Symposium on Wikis and Open Collaboration (WikiSym '12)*. ACM, 2012.

[10] C. Silveira, J. P. Faria, A. Aguiar, and R. Vidal. Wiki-Based Requirements Documentation of Generic Software Products. In *Proceedings of the 10th Australian Workshop on Requirements Engineering (AWRE05)*, 2005.

[11] A. van Lamsweerde. *Requirements Engineering*. Wiley, 2009.

[12] R. Witte and T. Gitzinger. Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In *3rd Asian Semantic Web Conference (ASWC)*, LNCS vol. 5367, pages 360–374. Springer, 2008.