

Die SoftWiki-Ontologie: Grundlage für semantische Interoperabilität im Requirements Engineering

Thomas Riechert¹, Steffen Lohmann²

¹Agile Knowledge Engineering and Semantic Web
Institut für Informatik, Universität Leipzig, 04009 Leipzig
rieichert@informatik.uni-leipzig.de

²Interaktive Systeme und Interaktionsdesign, Universität Duisburg-Essen
Lotharstraße 65, 47057 Duisburg
steffen.lohmann@uni-due.de

Kurzfassung: Eine gemeinsame, konzeptuelle Grundlage ist wesentlich für erfolgreiches Requirements Engineering mit einer großen Anzahl verteilter Stakeholder. Die SoftWiki-Ontologie definiert zentrale Konzepte des Requirements Engineering in standardisierter Semantic-Web-Syntax und schafft damit das Fundament für ein gemeinsames Verständnis, semantische Interoperabilität, einfache Portabilität sowie automatisierte Analysen und Konsistenzprüfungen. Bei der Entwicklung der Ontologie wurde auf ein hohes Alignment zu bereits bestehenden Vokabularen und Ontologien geachtet. Teile der Ontologie finden darüber hinaus eine unmittelbare Entsprechung in der Benutzeroberfläche der kollaborativen SoftWiki-Webplattform.

1 Einführung

Die Entwicklung eines gemeinsamen Verständnisses aller Beteiligten über die Ziele, Szenarien und Anforderungen eines geplanten Softwaresystems ist eine wesentliche Voraussetzung für erfolgreiches Requirements Engineering. Damit diese Voraussetzung auch im Falle einer großen Anzahl verteilter Stakeholder erfüllt werden kann, folgt der SoftWiki-Ansatz den Prinzipien des kollaborativen Content-Managements, das auf einem semantischen, datenorientierten Wiki als Kollaborationsplattform aufsetzt (vgl. Beitrag 5 des Buches). Die Grundlage hierfür bildet die SoftWiki-Ontologie¹, die wesentliche Konzepte des Requirements Engineering auf Basis von Semantic-Web-Standards in strukturierter und formal konsistenter Form beschreibt. Sie liefert nicht nur das semantische Fundament für den SoftWiki-Methoden- und -Werkzeugverbund, sondern ermöglicht zudem eine hohe Interoperabilität und Portabilität und vereinfacht automatisierte Analysen und Konsistenzprüfungen auf der gesamten Datenbasis.

¹Die vollständige, englische Bezeichnung lautet „SoftWiki Ontology for Requirements Engineering“ und wird mit „SWORE“ abgekürzt.

Dieser Beitrag stellt die SoftWiki-Ontologie im Detail vor. Nach einer kurzen Einführung in die Beschreibungssprachen des Semantic Web in Abschnitt 2, wird in Abschnitt 3 auf die Kernkonzepte der SoftWiki-Ontologie eingegangen. Hierbei werden insbesondere auch die Verbindungen mit bestehenden Ontologien und Vokabularen beschrieben. Außerdem wird dargestellt, in welcher Form sich die Ontologie-Konzepte in der Benutzeroberfläche der SoftWiki-Webplattform (vgl. Beitrag 6 des Buches) wiederfinden. Der Beitrag schließt mit einem Fazit und Ausblick auf zukünftige Erweiterungsmöglichkeiten der SoftWiki-Ontologie in Abschnitt 4.

2 Semantische Repräsentation mittels OWL

Unter dem Begriff „Ontologie“ wird in der Informatik üblicherweise eine „Spezifikation einer Konzeptualisierung“ [Gru93] verstanden. Eine Ontologie definiert eine gemeinsame Sichtweise auf eine Domäne, indem sie Konzepte und Beziehungen dieser Domäne auf konsistente Weise beschreibt und logische Schlussfolgerungen ermöglicht. Ein wesentliches Ziel der Semantic-Web-Initiative² ist es, Konzeptstrukturen erprobter Wissensbereiche in Form von Ontologien zu repräsentieren und mittels standardisierter Beschreibungssprachen über das Web weltweit verfügbar zu machen. Im Zuge dieser Entwicklungen sind eine Reihe von Ontologien und standardisierter Vokabulare entstanden, die zunehmend eingesetzt werden, um Ressourcen und Objekte semantisch eindeutig auszuzeichnen.

2.1 Grundlegendes zur SoftWiki-Ontologie

Die SoftWiki-Ontologie bietet eine solche, semantisch eindeutige Repräsentation für wesentliche Konzepte des Requirements Engineering. Bei der Entwicklung der Ontologie wurde darauf geachtet, verfügbare und etablierte Vokabulare und Ontologien zu integrieren, die sich sinnvoll zur Beschreibung von Teilbereichen des Requirements Engineering einsetzen lassen (wie z.B. Dublin Core, SKOS oder FOAF, s.u.). Hierdurch soll vermieden werden, dass Konzeptstrukturen in Teilbereichen, die bereits an anderer Stelle semantisch eindeutig beschrieben sind, in einer weiteren Ontologie abgebildet werden.

Die SoftWiki-Ontologie ist in der XML-basierten Beschreibungssprache OWL³ repräsentiert. Hierbei handelt es sich um einen Standard des World Wide Web Consortium (W3C)⁴, der die Beschreibungssprachen RDF⁵ und RDFS⁶ um zusätzliche Ausdrucksmöglichkeiten erweitert. Eine ausführliche Einführung in die Beschreibungssprachen RDF, RDFS und OWL findet sich beispielsweise in [HKSR07].

²W3C Semantic Web Activity, <http://www.w3.org/2001/sw/>

³Web Ontology Language (OWL), <http://www.w3.org/2004/OWL/>

⁴World Wide Web Consortium (W3C), <http://www.w3.org/>

⁵Resource Description Framework (RDF), <http://www.w3.org/RDF/>

⁶RDF-Schema, <http://www.w3.org/TR/rdf-schema/>

Wesentliche OWL-Konstrukte, die bei der Modellierung der SoftWiki-Ontologie verwendet wurden, sind Klassen (`owl:Class`), Eigenschaften (wie `owl:ObjectProperty`, `owl:DatatypeProperty`) und Beziehungen (wie `owl:inverseOf`, `owl:disjointWith`). Abbildung 1 veranschaulicht die Verwendung dieser Konstrukte anhand der grafischen Darstellung eines Ausschnitts der SoftWiki-Ontologie.

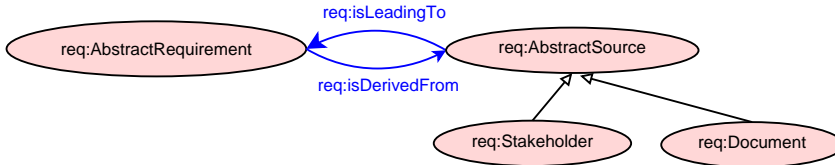


Abbildung 1: Ausschnitt aus der SoftWiki-Ontologie

Die Klassen der SoftWiki-Ontologie sind überwiegend in einer hierarchischen Struktur angeordnet. Da Stakeholder und Dokumente beispielsweise typische Informationsquellen sind, die bei der Anforderungserhebung berücksichtigt werden müssen, bilden `req:Stakeholder` und `req:Document` Unterklassen von `req:AbstractSource`. Die OWL-Repräsentation der modellierten Beziehungen zwischen den Klassen `req:AbstractSource` und `req:AbstractRequirement` gibt Listing 1 wieder.

```
<owl:ObjectProperty rdf:about="isDerivedFrom">
  <rdfs:range rdf:resource="AbstractSource"/>
  <rdfs:label xml:lang="de">ist abgeleitet von</rdfs:label>
  <rdfs:domain rdf:resource="AbstractRequirement"/>
  <owl:inverseOf rdf:resource="isLeadingTo"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="isLeadingTo">
  <rdfs:domain rdf:resource="AbstractSource"/>
  <rdfs:range rdf:resource="AbstractRequirement"/>
  <rdfs:label xml:lang="de">führt zu</rdfs:label>
</owl:ObjectProperty>
```

Listing 1: OWL-Repräsentation der Beziehungen zwischen `req:AbstractSource` und `req:AbstractRequirement`

Über die erste „Object Property“ wird definiert, aus welcher Informationsquelle eine Anforderung abgeleitet wurde. Entsprechend führt die Informationsquelle zu einer Anforderung, so wie es die zweite „Object Property“ beschreibt. Da es sich somit um inverse Beziehungen handelt, wird dies explizit durch `owl:inverseOf` angegeben. Die Wertebereiche, auf die sich die jeweilige „Object Property“ bezieht, werden in RDFS mittels `rdfs:domain` und `rdfs:range` deklariert.

3 Kernkonzepte der SoftWiki-Ontologie

Abbildung 2 zeigt eine grafische Darstellung der Kernkonzepte der SoftWiki-Ontologie, wie sie in der derzeitigen Version 1.03 modelliert sind. Durch die Integration bestehender Ontologien und Vokabulare werden eine Reihe bereits dokumentierter und etablierter Konzepte importiert. Diese sind durch die entsprechenden Namensraum-Präfixe gekennzeichnet (foaf, sioc, skos, tag). Alle eigenständigen Konzepte der SoftWiki-Ontologie hingegen besitzen das Namensraum-Präfix req. Dieser Namensraum verweist auf den *Uniform Resource Identifier (URI)*, unter dem die SoftWiki-Ontologie zu finden ist⁷.

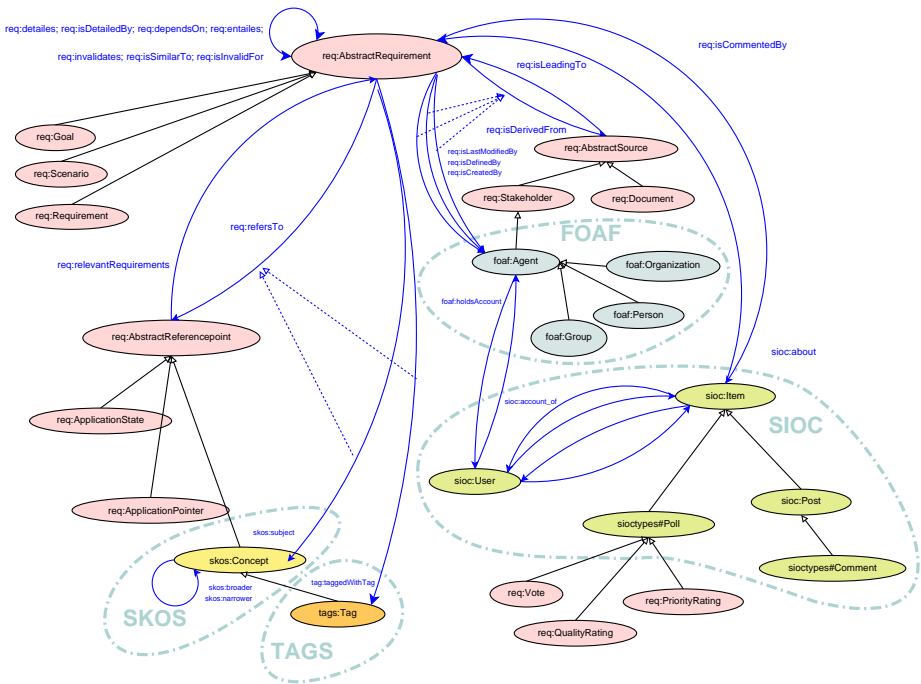


Abbildung 2: Grafische Darstellung der Kernkonzepte der SoftWiki-Ontologie

3.1 Anforderungen und Anforderungsbeziehungen

Wie bereits im vorigen Abschnitt erwähnt, wurden in der SoftWiki-Ontologie die beiden zentralen Klassen req:AbstractRequirement und req:AbstractSource definiert (vgl. auch Abb. 1). Von der Klasse req:AbstractRequirement wurden entsprechend einer Unterteilung nach [Poh07] die Sub-Klassen req:Goal, req:Scenario und

⁷Die URI der SoftWiki-Ontologie ist <http://ns.softwiki.de/req/>

req:Requirement abgeleitet (vgl. Beitrag 3 des Buches), wobei konkrete Anforderungen als Instanzen der Klasse req:Requirement dokumentiert werden. Anforders- titel und -beschreibung werden durch die im *Dublin-Core-Vokabular*⁸ definierten Ei- genschaften dc:title und dc:description abgebildet. Auch der Erstellungs- und Änderungszeitpunkt einer Anforderung werden in Dublin Core über die Eigenschaften dcterms:created und dcterms:modified festgehalten. Durch Verwendung von Du- blin Core sind die Metadaten zu den Anforderungen semantisch interoperabel beschrie- ben, d.h. sie können auch von anderen Anwendungen eingelesen und interpretiert werden, sofern diese Anwendungen den Dublin-Core-Standard unterstützen.

Abbildung 3 zeigt die Benutzeroberfläche zur Bearbeitung einer Anforderung innerhalb der SoftWiki-Webplattform (vgl. Beitrag 6 des Buches). Außerdem ist dargestellt, wie die eingegebenen Anforderungsdaten auf Dublin-Core und weitere Vokabulare (s.u.) ab- gebildet werden. Listing 2 gibt zusätzlich die OWL-Repräsentation der in Abbildung 3 darstellten Instanz wieder.

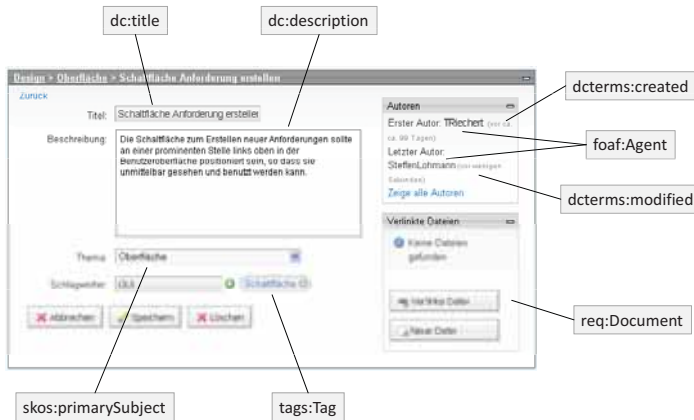


Abbildung 3: Dialog zur Bearbeitung einer Anforderung und verwendete Datentypen

Die Softwiki-Ontologie berücksichtigt außerdem, dass Anforderungen häufig in verschie- denartigen Abhängigkeiten zueinander stehen. Über die Eigenschaften req:details bzw. req:isDetailedBy können beispielsweise Detaillierungsbeziehungen ausgedrückt werden. Ebenso lassen sich req:dependsOn für Abhängigkeiten, req:entails für Ableitungen, req:invalidates und req:isInvalidFor für Konflikte sowie req:isSimilarTo für inhaltliche Ähnlichkeiten zwischen Anforderungen verwenden. Aus diesen vordefinierten Beziehungstypen kann der Nutzer auswählen, wenn er zwei Anforderungen miteinander verknüpft (vgl. Beitrag 5 des Buches). Bei Bedarf können darüber hinaus eigene Beziehungstypen definiert und auf einfache Weise in die SoftWiki-Ontologie eingebunden werden.

⁸Bei Dublin Core (DC) handelt es sich um ein standardisiertes Vokabular zur Beschreibung von Dokumenten. Weitere Informationen unter: <http://dublincore.org/>

```
<req:Requirement rdf:about="http://softwiki.de/demo/deutsch/Req2"
  dc:title="Schaltfläche Anforderung erstellen"
  dcterms:created="2009-04-16T12:12:41"
  dcterms:modified="2009-07-24T17:30:12" >
  <dc:description>Die Schaltfläche zum Erstellen neuer
    Anforderungen sollte an einer prominenten Stelle links
    oben in der Benutzeroberfläche positioniert sein, so
    dass sie unmittelbar gesehen und benutzt werden kann
  .</dc:description>
  <skos:primarySubject rdf:resource="UserInterface" />
  <req:isCreatedBy rdf:resource="http://softwiki.de/demo/
    deutsch/OntoWiki/Config/TRiechert" />
  <req:isLastModifiedBy rdf:resource="http://softwiki.de/
    demo/deutsch/OntoWiki/Config/SteffenLohmann" />
  <req:isDefinedBy rdf:resource="http://softwiki.de/demo/
    deutsch/OntoWiki/Config/TRiechert" />
  <tags:taggedWithTag rdf:resource="http://softwiki.de/demo/
    deutsch/Schaltfläche" />
  <tags:taggedWithTag rdf:resource="http://softwiki.de/demo/
    deutsch/GUI" />
  <req:dependsOn rdf:resource="http://softwiki.de/demo/
    deutsch/Req3" />
</req:Requirement>
```

Listing 2: OWL-Repräsentation einer Instanz der Klasse req:Requirement

3.2 Klassifizierung: Themenbaum, Tags und Glossar

Die SoftWiki-Ontologie unterstützt verschiedene Ansätze zur Klassifizierung von Anforderungen, die in ihrer Gesamtheit über die Klasse req:AbstractReferencepoint realisiert sind. Zum einen werden die Anforderungen in den Themenbaum des Projekts eingeordnet (siehe Abb. 4), indem sie einem Thema zugewiesen werden. Im Falle des in Abbildung 3 dargestellten Beispiels wurde die Anforderung beispielsweise dem Thema „Oberfläche“ zugeordnet, welches wiederum ein Unterthema von „Design“ ist.

Die Themenstruktur wird mit Hilfe des *Simple Knowledge Organization System (SKOS)*⁹ definiert. Über die SKOS-Eigenschaft skos:broader (bzw. die hierzu inverse Eigenschaft skos:narrower) können hierarchische Beziehungen innerhalb der Themenstruktur ausgedrückt werden. Die Zuweisung eines Themas zu einer Anforderung erfolgt mittels skos:primarySubject (eine Subklasse von skos:subject).

Neben diesem hierarchischen Klassifizierungsansatz über die Themenstruktur des Projekts ermöglicht die SoftWiki-Ontologie auch eine Klassifizierung durch frei gewählte Schlag-

⁹Bei SKOS handelt es sich um eine Beschreibungssprache zur Repräsentation von konzeptuellen Strukturen. Mehr Informationen unter: <http://www.w3.org/2004/02/skos/>

wörter (sog. *Tags*). Hierzu wurde die Klasse `tag:Tag` der *Tag-Ontology*¹⁰ importiert, die dort als Subklasse von `skos:Concept` definiert ist. Der Name eines Tags wird durch `tag:name` angegeben und über `tag:taggedWithTag` mit der Anforderung verknüpft (vgl. Listing 2). Zusätzlich können Tags über die RDFS-Eigenschaft `rdfs:comment` mit Definitionen versehen werden. Alle Instanzen der Klasse `tag:Tag` werden als *Tag Cloud* [LZT09] visualisiert in der Benutzeroberfläche dargestellt (siehe Abb. 4), wobei Tags, die über eine Definition verfügen, farblich hervorgehoben werden. Die Tag Cloud bietet damit eine weitere Zugriffsform auf die Anforderungen in Ergänzung zum Themenbaum.

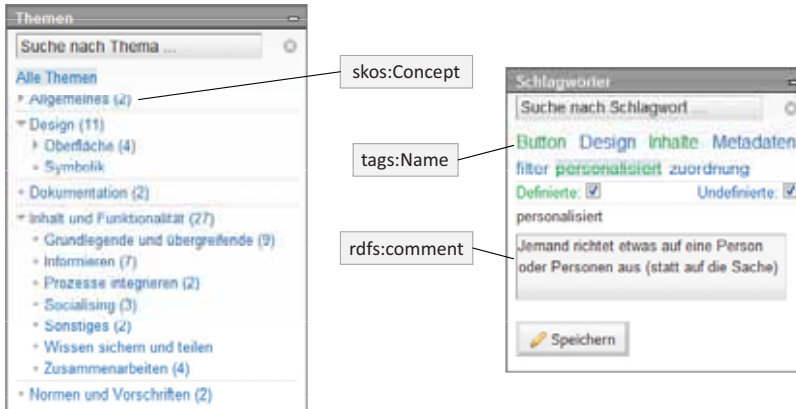


Abbildung 4: Themenbaum und Tag Cloud (zu einem Unterthema) eines Projekt

Als zusätzliche Referenzpunkte wurden in der SoftWiki-Ontologie die Klassen `req:ApplicationState` und `req:ApplicationPointer` definiert. Sie können beispielsweise bei der integrierten und kontextualisierten Erfassung von Nutzeranforderungen verwendet werden, um explizite Verweise auf den Systemzustand und Systemteile abzubilden (vgl. Beitrag 18 des Buches).

3.3 Anforderungsquellen: Stakeholder und Dokumente

Wie bereits erwähnt, können auch Informationsquellen, aus denen Anforderungen abgeleitet werden, in der SoftWiki-Ontologie repräsentiert werden. Hierzu wurde die abstrakte Klasse `req:AbstractSource` definiert, der verschiedenste Informationsquellen zugeordnet werden können. Für die beiden wesentlichen Anforderungsquellen „Stakeholder“ und „Dokumente“ wurden bereits entsprechende Subklassen definiert. Die Klasse `req:Stakeholder` wurde hierbei über die Eigenschaft `owl:sameClassAs` mit der Klas-

¹⁰<http://www.holygoat.co.uk/owl/redwood/0.1/tags/>

se `foaf:Agent` der FOAF-Ontologie¹¹ gleichgesetzt. Durch die Integration von FOAF-Konzepten wird es möglich, diesen Teil der Datenbasis als soziales Netz zu repräsentieren und entsprechend zu analysieren. Hierzu kann prinzipiell jede Anwendung genutzt werden, die FOAF-Dateien lesen und verarbeiten kann. Genauso können FOAF-Datensätze, die bereits zu Stakeholdern existieren, auf einfache Weise integriert werden.

In die SoftWiki-Ontologie wurden zusätzlich zur Klasse `foaf:Agent` auch deren Subklassen `foaf:Person`, `foaf:Group` und `foaf:Organization` importiert. Hierdurch ist berücksichtigt, dass ein Stakeholder nicht zwangsläufig eine einzelne Person sein muss, sondern auch eine Gruppe oder Organisation repräsentieren kann [Poh07].

Die Beziehungen `req:isCreatedBy`, `req:isDefinedBy` und `req:isLastModifiedBy` bilden ab, welche Personen (`foaf:Agent`) an der Definition einer Anforderung mitgewirkt haben. Dies ließe sich zwar prinzipiell auch über die Dublin-Core-Eigenschaft `dc:creator` abbilden, doch wurde an dieser Stelle aus Gründen einer besseren Nachvollziehbarkeit (Traceability) im kollaborativen Requirements Engineering eine feinere Unterteilung gewählt. `req:isCreatedBy` definiert den Stakeholder, der die Anforderung erstellt hat und somit typischerweise verantwortlich hierfür ist. `req:isDefinedBy` bildet alle weiteren beteiligten Stakeholder ab, die mit ihrer Bearbeitung zur Weiterentwicklung der Anforderung beigetragen haben. Und über `req:isLastModifiedBy` wird derjenige Stakeholder festgehalten, der die letzte Änderung vorgenommen hat.

Da die SoftWiki-Webplattform auf den Prinzipien eines semantischen, datenorientierten Wikis basiert (vgl. Beitrag 4 des Buches), werden im Sinne einer umfassenden Revisionskontrolle auch noch weitere Daten zu den Änderungszuständen (wie die komplette Änderungshistorie und Unterschiede in den Versionen einer Anforderung) mitprotokolliert. Allerdings werden diese Daten systemintern gepflegt und nicht mittels der SoftWiki-Ontologie repräsentiert, um die Ontologie und den dazugehörigen Datensatz möglichst 'schlank' zu halten. Für die meisten Anwendungsfälle sollte eine Repräsentation der oben genannten Daten jedoch ausreichen.

Die Klasse `req:Document` ist durch die SoftWiki-Ontologie nicht weiter spezifiziert, da bei der Definition von Anforderungen potenziell Dokumente verschiedener Datei- und Medienformate herangezogen werden können: von unterschiedlich formatierten und strukturierten Textdokumenten bis hin zu Grafiken und multimedialen Ressourcen. Auch `req:Document` findet eine Entsprechung in der SoftWiki-Webplattform, wie Abbildung 3 zeigt. Um eine hohe Rückverfolgbarkeit zu gewährleisten, sollten alle Dokumente über eine URI verfügen. Bestenfalls ist dies die URI, die den Dokumenten automatisch zugewiesen wird, sobald sie beispielsweise in die SoftWiki-Webplattform eingestellt werden. Alternativ kann ein Dokument jedoch auch über die URL, unter der es im Internet oder Intranet zu finden ist, identifizierbar sein.

¹¹ FOAF (Friend of a Friend) wurde entwickelt, um Personen und deren Beziehungen innerhalb eines sozialen Netzwerks in Semantic-Web-Syntax abzubilden. Mehr Informationen unter: <http://xmlns.com/foaf/spec/>

3.4 Diskussion: Kommentierung und Bewertung

Die Diskussion von Anforderungen kann im kollaborativen Requirements Engineering einen hohen Stellenwert einnehmen. Dies ist in der SoftWiki-Ontologie berücksichtigt, indem die SIOC-Ontologie importiert wurde¹². Alle Kommentare und Bewertungen werden als `sioic:Item` repräsentiert und über `sioic:about` (bzw. die als invers definierte Eigenschaft `req:isCommentedBy`) mit der zugehörigen Anforderung verknüpft. Als zwei wesentliche Subklassen für kollaboratives Requirements Engineering wurden `sioic:Post` und `sioicotypes#Poll` importiert. `sioic:Post` (bzw. die Subklasse `sioicotypes#Comment`) repräsentiert alle Kommentare, die Nutzer zu einer Anforderung äußern (vgl. Abb. 5). Über `sioicotypes#Poll` werden Bewertungen zu den Anforderungen abgebildet. Die SoftWiki-Ontologie unterscheidet hierbei zwischen drei Arten von Bewertungen: `req:Vote` bildet die Entscheidung für oder gegen eine Anforderung ab, wohingegen `req:PriorityRating` die Priorität für die Realisierung einer Anforderung wiedergibt; über `req:QualityRating` wiederum wird die allgemeine Qualität einer Anforderung ausgedrückt.

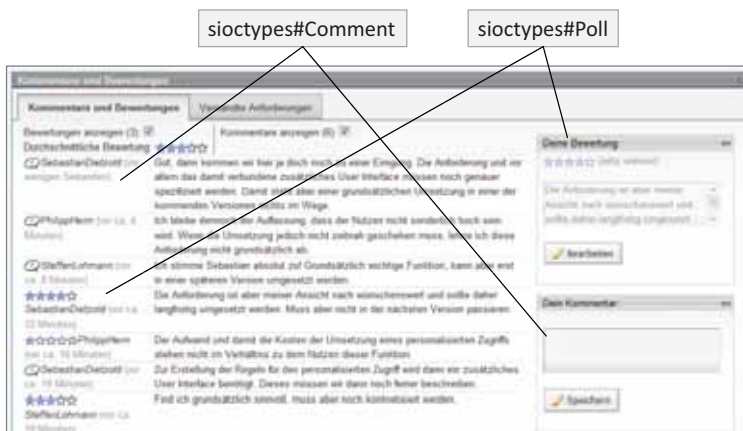


Abbildung 5: Kommentierung und Bewertung einer Anforderung

Die Anzahl der Kommentare (`sioicotypes#Comment`) pro Anforderung und Nutzer ist durch die SoftWiki-Ontologie prinzipiell nicht begrenzt. Ein Nutzer kann sich also mehrmals zu einer Anforderung äußern, was im Verlauf einer Diskussion der Anforderung natürlich auch möglich sein sollte. Bei der Anzahl der Bewertungen hingegen erlaubt die SoftWiki-Ontologie für jede Art von Bewertung (`req:Vote`, `req:PriorityRating`, `req:QualityRating`) maximal eine Bewertung pro Nutzer und Anforderung. Hierdurch wird sichergestellt, dass ein Nutzer nicht mehrmals für eine Anforderung abstimmen und dadurch die Durchschnittsbewertung zu seinen Gunsten beeinflussen kann.

¹²SIOC (Semantically-Interlinked Online Communities) ermöglicht es, Inhalte von Nutzern auf Community-orientierten Webplattformen (wie Foren, Wikis oder Weblogs) zu repräsentieren und zu verknüpfen.

4 Fazit und Ausblick

In diesem Beitrag wurde die SoftWiki-Ontologie vorgestellt und ihre Funktion im Requirements Engineering erläutert. Außerdem wurde gezeigt, in welcher Form sich die Konzepte der Ontologie in der Benutzeroberfläche der kollaborativen SoftWiki-Webplattform wiederfinden.

Die Ontologie wurde insbesondere mit Blick auf ein agiles und kollaboratives Requirements Engineering sowie Softwareprojekte mit einer großen Anzahl verteilter Stakeholder entwickelt, was sich auch in der Integration von Ontologien wie FOAF, SIOC oder der Tag-Ontology ausdrückt. Darüber hinaus gewährleistet die Verwendung der Beschreibungssprache OWL eine hohe Ausdrucksmächtigkeit, die kaum mit Beschränkungen hinsichtlich Projekt-spezifischer Erweiterungen oder allgemeiner Weiterentwicklungen der SoftWiki-Ontologie einhergeht. Zusätzlich ermöglicht OWL ein Reasoning auf der Datenbasis, beispielsweise zur automatischen Validierung und Konsistenzprüfung. Die Auszeichnung der Daten mit eindeutigen URIs gewährleistet darüber hinaus eine hohe Nachvollziehbarkeit (Traceability) der Anforderungen über Werkzeuge und Projekte hinaus.

Alles in allem bildet die SoftWiki-Ontologie ein semantisch konsistentes Repräsentationsformat für die Dokumentation von Anforderungen und hiermit verbundener Daten im SoftWiki-Methoden- und -Werkzeugverbund. So greifen beispielsweise auch die Visualisierungskomponente des SoftWiki-Ansatzes (vgl. Beitrag 20 des Buches) oder das Webbrowser-Plugin zur integrierten und kontextualisierten Erfassung von Nutzeranforderungen (vgl. Beitrag 10 des Buches) auf die SoftWiki-Ontologie zu. Darüber hinaus wird die semantische Interoperabilität und allgemeine Portabilität der Ontologie durch ein hohes Alignment zu etablierten Vokabularen und Ontologien zusätzlich gesteigert. Teilbereiche der Datenbasis können somit vergleichsweise einfach und bestenfalls ohne Informationsverlust oder zusätzlichen Transformationsaufwand in andere Werkzeuge importiert und dort analysiert und bearbeitet werden. Beispielsweise kann mit FOAF-kompatiblen Werkzeugen eine Analyse des sozialen Netzwerks [PW05] der Stakeholder durchgeführt oder über SIOC-kompatible Werkzeuge der Diskussionsverlauf zu Anforderungen genauer analysiert werden.

Für zukünftige Versionen der SoftWiki-Ontologie ist geplant, weitere Konzepte des Requirements Engineering sowie zusätzliche Ontologien (wie DOAP¹³) zu integrieren. Außerdem soll die Kompatibilität zu existierenden Austauschformaten für Anforderungsdaten (wie RIF¹⁴) untersucht und – wo sinnvoll – erhöht werden.

Literatur

[Gru93] Tom R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

¹³Description of a Project (DOAP), usefulinc.com/doap

¹⁴Requirements Interchange Format (RIF), <http://www.automotive-his.de/rif/>

- [HKSR07] Pascal Hitzler, Markus Krötzsch, York Sure und Sebastian Rudolph. *Semantic Web*. eXamen.press. Springer, 2007.
- [LZT09] Steffen Lohmann, Jürgen Ziegler und Lena Tetzlaff. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. In *Human-Computer Interaction - INTERACT 2009*, Jgg. 5726 of LNCS, Seiten 392–404. Springer, 2009.
- [Poh07] Klaus Pohl. *Requirements Engineering. Grundlagen, Prinzipien, Techniken*. Dpunkt, 2007.
- [PW05] John C. Paolillo und Elijah Wright. *Social Network Analysis on the Semantic Web: Techniques and Challenges for Visualizing FOAF*, Seiten 229–242. Springer, 2005.

Inhaltsverzeichnis

Kapitel 1: Motivation, Begriffe und Typologien

Sören Auer

1 Das Projekt SoftWiki: Methoden- und Softwareunterstützung für agiles, endanwender-getriebenes Requirements Engineering3

Martin Friedel, Dirk Fritsch, Kim Lauenroth

2 Stand der Forschung und Technik zum Einsatz von Wiki-Systemen im verteiltem Requirements Engineering mit großen Stakeholdergruppen.....9

Kim Lauenroth

3 Einführung in das Requirements Engineering und Nutzen von Wiki-Systemen für das Requirements Engineering17

Thorsten Berger, Sebastian Dietzold, Thomas Riechert

4 Der Einsatz semantischer Daten-Wikis in frühen Phasen des Requirements Engineering.....27

Kapitel 2: Vorgehen, Methoden und Werkzeuge

Kim Lauenroth, Thomas Riechert

5 Der SoftWiki-Ansatz für verteiltes Requirements Engineering mit großen Stakeholdergruppen39

Sebastian Dietzold, Thomas Riechert

6 Realisierung einer Web-basierten Plattform für das verteilte Requirements Engineering auf Basis des Application-Frameworks OntoWiki49

Thomas Riechert, Steffen Lohmann

7 Die SoftWiki-Ontologie: Formale Grundlage für semantische Interoperabilität im Requirements Engineering.....59

Haiko Cyriaks, Christian Raether, Horst Stolz, Veli Velioglu

8 Ein Ansatz zur Erkennung ähnlicher Anforderungen auf Basis semantischer Textanalyse71

Philipp Heim, Timo Stegemann, Steffen Lohmann, Jürgen Ziegler, Haiko Cyriaks, Horst Stolz

9 InteractiveExtractor: Durchgängige Unterstützung bei der Extraktion von anforderungsrelevanten Informationen81

Steffen Lohmann, Jürgen Ziegler
10 Integrierte und kontextualisierte Erfassung von Nutzeranforderungen.....89

Thorsten Berger, Raimund S. Utz, Dirk Fritsch
11 Integration des SoftWiki-Systems mit konventionellen Requirements Engineering
Tools97

Thomas Riechert, Fan Zhang, Sören Auer
12 Integration des SoftWiki-Ansatzes in das Cofundos-Entwicklungsmodell..... 109

Kapitel 3: Ausgewählte Anwendungsfälle und Evaluierung

Martin Friedel, Kim Lauenroth
13 Vorgehensweise und Evaluationskriterien 119

Bettina Widmann, Raimund S. Utz
14 Einsatz der SoftWiki-Plattform für den Innovationsprozess..... 133

Anita Wenzel
15 Einsatz von SoftWiki zur Prototypentwicklung im E-Procurement..... 137

Peter Bormann
16 Einsatz von SoftWiki zur Einführung eines Enterprise-Content-Management-
Systems (ECM)..... 141

Bettina Widmann
17 Ergebnisse und Auswertung der industriellen Evaluierung von SoftWiki..... 145

Kapitel 4: Ausblick und weiterführende Arbeiten

Anita Wenzel, Dirk Fritsch
18 Durchgängige Informationsrückflüsse aus dem Software-Entwicklungsprozess
in das SoftWiki-System 155

Sebastian Schneider
19 Geschäfts- und Betreibermodelle 161

Philipp Heim, Jürgen Ziegler
20 Graph-basierter facetierter Zugriff auf semantisch annotiertes
Anforderungswissen 173

Asarnusch Rashid, Andreas Oberweis

21 OpenProposal – Ein Werkzeug zur Verbesserung der Anwenderbeteiligung in
Softwareprojekten 179

Martin Böhringer, Peter Gluchowski, Christian Kurze

22 Spezifika von Microblogging-Anwendungen als Semantic-Web-Applikationen
am Beispiel von Twitter 193

Philipp Heim, Thomas Schlegel, Thomas Ertl

23 Vom Gezwitscher zur umfassenden und aktuellen Situationseinschätzung 199