

Maariwa – ontologiebasiertes Webpublishing mit einem Semantic Wiki

Rico Landefeld, Harald Sack
Friedrich-Schiller-Universität Jena,
D-07743 Jena
rico.landefeld@takwa.de, sack@minet.uni-jena.de

Abstract: Semantic Wikis dienen sowohl der kollaborativen Erstellung von Wissensrepräsentationen als auch der semantischen Annotation textueller Wiki-Inhalte. In dieser Arbeit wird ein Semantic Wiki Ansatz vorgestellt, der ein vereinfachtes, auf den Einsatz in einem Wiki zugeschnittenes Ontologiemetamodell mit einem auf aktuellen Browsertechnologien basierenden Bedienkonzept und einer einfachen semantischen Abfragesprache kombiniert. Textfragmente einer Wiki-Seite können interaktiv mit einer maschinenverwertbaren Semantik in einer für den Nutzer leicht verständlichen Art und Weise verknüpft werden, wobei der zusätzliche zur Annotierung zu leistende Aufwand minimiert wird. Durch ein einfaches und intuitives Bedienkonzept soll die Leistungsfähigkeit eines Semantic Wiki Systems auch für Anwenderkreise ohne Expertenwissen erschlossen werden.

1 Einführung

Der erste Browser des World Wide Webs (WWW) hatte eine Funktion, die schnell wieder in Vergessenheit geriet: Webseiten konnten nicht nur gelesen, sondern auch direkt verändert (geschrieben) werden. Wikis [LC01] haben diese Idee vor einigen Jahren wieder aufgegriffen und geben jedem Besucher einer Wiki-Webseite die Möglichkeit, diese auch zu verändern. Wikis sind schlanke Content Management Systeme, die HTML-Dokumente verwalten, welche vom Besucher eines Wikis unter ausschließlicher Verwendung eines Web-Browsers erstellt oder geändert werden können. Die Dokumente eines Wikis werden auf diese Weise kollaborativ von der Gemeinschaft aller Nutzer, einer so genannten „Community“, erstellt und gepflegt. Wikis machen ihren Benutzern keine formalen Vorgaben, auf welche Weise der Inhalt aufzubauen und zu strukturieren ist. Diese Freiheit begünstigte ein rasches Wachstum, wie das Beispiel der freien Enzyklopädie Wikipedia¹ zeigt, erfordert aber häufiges Refaktorisieren, wenn ein Wiki mit wachsender Seitenzahl noch übersichtlich und navigierbar bleiben soll. Klassische Wikis bieten nur eingeschränkt Funktionen zur Strukturierung des Inhalts. Meistens werden von den Benutzern manuell spezielle Übersichtsseiten und Kategoriensysteme angelegt, nach denen die Dokumente des Wikis geordnet werden. Dieses Vorgehen ist mit wachsender Größe des Wikis mit einem hohen Pflegeaufwand verbunden. In der Wikipedia ist als Resultat zu beobachten, dass das Kategoriensystem missbräuchlich verwendet wird, indem z.B. neue Kategorien für Artikel

¹<http://www.wikipedia.org>

geschaffen wurden, die lediglich eine einzige bestimmte gemeinsame Eigenschaft teilen [VKV⁺06]. Ähnliche Probleme werden auch für Intranet Wikis berichtet [BG06]. Die Probleme des regulären Wikiansatzes lassen sich darauf zurückführen, dass die in einem Wiki enthaltenen Informationen in der Regel in der Markupsprache HTML kodiert sind und daher nicht, oder nur eingeschränkt, automatisch aggregiert und wiederverwendet werden können.

Semantic Wikis stellen eine Symbiose des Wikiansatzes mit semantischen Technologien eingebettet in das im Entstehen begriffene Semantic Web dar [BLHL01]. Sie verknüpfen textuellen Inhalt mit einem Wissensmodell, das in Wiki-Seiten beschriebenes Wissen in einer formalen Sprache festhält und damit maschinell verarbeitbar und verstehbar machen soll. Semantic Wikis stellen einen möglichen Weg dar, die Probleme klassischer Wikis zu lösen und gleichzeitig die kollaborative Erstellung und Pflege von formalen Wissensrepräsentationen (Ontologien) zu ermöglichen. Damit ein Semantic Wiki breiten Anwenderkreisen und nicht nur Experten offen steht, darf allerdings das „ease of use“-Prinzip klassischer Wikis nicht verletzt werden.

Existierende Ansätze haben verschiedene Wege gewählt, um formale Wissensmodelle in einem Wiki nutzbar zu machen. Eine entscheidende Bedeutung hat unserer Meinung nach das Verhältnis einerseits aus kognitiven und faktischen Aufwand, der nötig ist, um semantische Annotationen zu erstellen und andererseits dem praktischen Nutzen für die Anwender eines Wikis, der durch Funktionen entsteht, die diese Annotationen nutzen. Der notwendige Aufwand der Annotierung wird maßgeblich durch das Bedienkonzept und das verwendete Ontologiemetamodell des Wikis bestimmt. Gleichzeitig beeinflusst die semantische Ausdrucksstärke der Annotationen die Mächtigkeit der Funktionen, die darauf aufbauen. Deshalb muss das Ontologiemetamodell eines Semantic Wikis einen Kompromiss zwischen Komplexität und Ausdrucksstärke darstellen. Wir sind außerdem der Meinung, dass die Integration semantischer Annotationen in Wikis neue Bedienkonzepte erfordern, die den entstehenden Aufwand minimieren helfen. Bestehende Semantic Wikis verwenden entweder Ontologiemetamodelle, die Sprachelemente verwendeter Sprachen zur Wissensrepräsentation (WR-Sprachen) direkt abbilden, Verknüpfungen im vorhandenen Wissensmodell schwer wiederverwendbar machen oder eine schwache Semantik besitzen, welche die Ausnutzung semantischer Annotationen durch Wikifunktionen einschränkt. Größtenteils wird direkt auf klassischen Wiki-Systemen aufgesetzt oder diese nachgebildet und damit auch deren Bedienkonzepte vererbt. Wir schlagen deshalb einen Ansatz vor, der folgende drei Konzepte vereint: ein vereinfachtes, für den Einsatz in einem Wiki zugeschnittenes Ontologiemetamodell, einen WYSIWYG-Editor, der ein integriertes Bedienkonzept für Text- und Ontologiebearbeitung realisiert, und eine eigene, möglichst einfache semantische Abfragesprache. Dieser Ansatz wurde in Form des von uns entwickelten Semantic Wikis *Maariwa*² erfolgreich implementiert.

Die vorliegende Arbeit ist wie folgt gegliedert: Kapitel 2 widmet sich verwandten Arbeiten, deren Ontologiemetamodelle und Bedienkonzepte näher betrachtet werden. In Kapitel 3 wird das Semantic Wiki Projekt *Maariwa* vorgestellt. Das letzte Kapitel fasst die Ergebnisse der Arbeit zusammen und diskutiert mögliche Erweiterungen.

²<http://ipc755.inf-nf.uni-jena.de:8081/Maariwa>

2 Verwandte Arbeiten und Ansätze

Semantic Wikis haben ihre Wurzeln nicht nur in klassischen Wikiansätzen, sondern auch in Arbeiten, die das gruppenbasierte Erstellen und Pflegen von Ontologien über das WWW untersuchten. Erste Arbeiten zu diesem Thema erschienen in der Mitte der neunziger Jahre, noch vor dem Start der Semantic Web Initiative des W3C (vgl. [FFR97, SRKK97, ACFLGP01]). Mit der Etablierung der vom W3C geförderten Semantic Web Technologien und der wachsenden Popularität der Wikis wurde dieses Thema erneut aufgegriffen und mit Wikikonzepten verbunden. Je nach Anwendungsfokus stehen dabei entweder der textuelle oder der formalisierte Inhalt des Wikis im Vordergrund.

Auf der einen Seite beschreibt das Wikitology Paradigma [DRR⁺05] Wiki-Systeme, die als Benutzerschnittstelle zu einer Ontologie fungieren, in denen jede Seite des Wikis ein Konzept und jeder Link zwischen den Seiten eine Beziehung in der zugrunde liegenden Ontologie repräsentiert. Das Wiki bildet das Werkzeug, mit dem die Ontologie bearbeitet wird. Steht auf der anderen Seite der textuelle Inhalt bei einem Semantic Wiki im Vordergrund, wird das Wissensmodell verwendet, um die Navigation, die Durchsuchbarkeit und die Wiederverwendbarkeit der im Wiki enthaltenen Informationen zu verbessern. Wir sprechen bei solchen Ansätzen von ontologiegestützten Wikis. Ein weiteres Unterscheidungsmerkmal unterschiedlicher Semantic Wiki Implementationen bildet die Art und Weise, wie das Ontologiemetamodell für die Anwendung in einem Wiki angepasst wurde und wie stark damit die zugrundeliegenden WR-Sprachen verdeckt werden. Das Ontologiemetamodell eines Semantic Wikis definiert eine Abbildung von Elementen der verwendeten WR-Sprachen auf Elemente des Anwendungsmodells. Darüberhinaus legt das Ontologiemetamodell die semantische Ausdrucksstärke der Annotationen fest und bietet gleichzeitig eine Basis für die Abfrage von Informationen. Die Form der Integration der Ontologieelemente in das Wiki legt fest, ob semantische Annotationen zusammen oder separat vom Text der Seiten gepflegt werden können [ODM⁺06]. Im folgenden werden die wichtigsten Semantic Wiki Implementierungen vorgestellt und anhand ihrer Ontologiemetamodelle miteinander verglichen.

Platypus Wiki [CCT04] ist eine der frühesten Semantic Wiki Implementationen und verwendet einen Ansatz, der ein Wiki Web auf einen RDF Graphen [MM04] abbildet, indem jeder Textseite des Wikis eine RDF-Ressource und jedem Link ein RDF-Property zugeordnet wird. Semantische Annotationen werden parallel zum Seitentext in einem eigenem Eingabefeld als RDF(S) [BG04] oder OWL [MvH03] in XML-Serialisierung eingegeben.

Rhizome [Sou04] unterstützt semantische Annotationen mit Hilfe einer speziellen Wiki Markup Language (WikiML). Der gesamte Inhalt des Wikis, Text, Struktur und Metadaten, wird intern über RDF kodiert und ist mit Hilfe eines geeigneten RDF-Editors auch separat editierbar. Im Gegensatz zu klassischen Wikiansätzen verwendet Rhizome ein feingranulares Sicherheitsmodell. Rhizome unterstützt zwar das Anlegen von Metadaten, bietet aber keine zusätzlichen Funktionen, die diese Metadaten ausnutzen.

Semantic Media Wiki (SMW) [VKV⁺06, KVV05] baut auf dem weit verbreiteten Media Wiki³ auf. Das bekannteste Anwendungsbeispiel des Media Wikis ist die freie Enzy-

³<http://www.mediawiki.org/>

klopädie Wikipedia. Das Entwicklungsziel des SMWs ist es, im Rahmen der Wikipedia die Durchsuchbarkeit und Strukturierbarkeit durch den Einsatz semantischer Technologien zu verbessern. Die zusätzlichen Funktionen sind deshalb so angelegt, dass sie für einen breiten Anwenderkreis wie die Wikipedia-Community anwendbar bleiben. SMW erweitert die WikiML-Syntax des Media Wiki um Konstrukte für Attribute, Typen und Beziehungen. Klassen werden durch die im Media Wiki bereits vorhandenen Kategorien repräsentiert.

Makna Wiki [DST06] verwendet einen RDF-Triple orientierten Ansatz zur Annotation von Wiki-Seiten. Die aus Subjekt, Prädikat und Objekt bestehenden Triples können entweder während der Bearbeitung einer Wiki-Seite im Editor mit WikiML-Konstrukten oder seitenübergreifend formularbasiert hinzugefügt werden. Mit Makna Wiki können nur Instanzen erzeugt und bearbeitet werden.

IkeWiki [SGW05] ist für eine Community konzipiert, bei der Fachexperten und WR-Experten zusammenarbeiten, um die textuellen und formalen Inhalte des Wikis zu erstellen. In der Benutzeroberfläche spiegelt sich diese Herangehensweise in einer Trennung der Bearbeitungsfunktionen für den Seitentext und der Annotationen wieder. Der Editor für Annotationen erlaubt es, Wiki-Seiten mit Klassen und Hyperlinks mit Beziehungen zu typisieren. Der Editor unterstützt den Anwender mit Autovervollständigungsfunktionen für Metadaten.

SweetWiki [BG06] verbindet kollaborative Annotation (social Tagging) [GH06] mit semantischen Technologien zu einem als *semantic Tagging* bezeichnetem Konzept. Wiki-Seiten werden von Anwendern mit Schlüsselwörtern (Tags) versehen, die zusammen nicht nur einen kollektiv erstellten Index, eine so genannte *Folksonomie* [Van05] bilden, sondern eine formale Ontologie. Dies wird erreicht, indem jedes Tag als Konzept einer Ontologie aufgefasst wird. Die Beziehungen zwischen den Konzepten und deren Hierarchie wird nicht von den Anwendern, sondern von einem gesondertem Expertenkreis festgelegt. Der Anwender kommt mit der Ontologie nicht direkt in Berührung, er kann lediglich Tags anlegen und zuordnen. Für den Anwender findet keine Unterscheidung in Instanzen- und Schemadaten statt, da Tags Konzepte repräsentieren, die sowohl Instanzen als auch Klassen sein können.

Neben den vorgestellten Arbeiten existieren noch zahlreiche weitere Semantic Wiki Implementierungen, die sich ebenfalls in den durch die genannten Beispiele aufgespannten Rahmen einordnen lassen. Konzeptionell bewegen sich alle Implementierungen zwischen Wikitologies und ontologiestützten Wikis. Neben dem textuellen Inhalt können Wissensmodelle aufgebaut werden, die für zusätzliche Funktionen genutzt werden können. Frühe Ansätze wie Platypus trennen die Eingabe der textuellen von den formalen Inhalten, während spätere Ansätze mit Hilfe einer erweiterten WikiML Syntax beides zusammenführen. IkeWiki und Makna Wiki unterstützen den Autor bei der semantischen Annotation darüber hinaus mit dynamischen Eingabehilfen.

Wenn die Sprachelemente von RDF(S) bzw. OWL im Ontologiemetamodell direkt verwendet werden, existieren für die Abfrage keine Vereinfachungen, so dass ohne Brüche im Anwendungsmodell eine Abfragesprache wie SparQL[PS07] verwendet werden kann, die für RDF und darauf aufbauende Sprachen konzipiert wurde.

Projekt	Annotierungsform	Ontologiemetamodell	Verdeckung von WR-Sprachelementen	semantische Suche
Platypus	Text und Annotierung getrennt	RDF	nein	nein
Rhizome	WikiML	RDF	nein	nein
SMW	WikiML	OWL-DL	ja (Kategorien, Attribute, Beziehungen)	ja
Makna Wiki	WikiML und formularbasiert	RDF	nein (Verwendung von Triples)	ja, eingeschränkt
IkeWiki	WikiML und WYSIWYG	RDF(S) und OWL	nein	nein
Sweet Wiki	WYSIWYG, Zuordnung von Tags	OWL	ja (Tags)	nein

Tabelle 1: Vergleich verschiedener Semantic Wiki Konzepte

SMW implementiert ein vereinfachtes, auf OWL-DL basierendes Ontologiemetamodell, dessen Elemente in einer eigenen, im Vergleich zu SparQL wesentlich einfacheren Abfragesprache verwendet werden. Bei diesem Ansatz bleibt die Verknüpfung zwischen Klassen und ihren Attributen und Beziehungen verborgen. Bei der Annotierung wird direkt keine Hilfestellung durch schon erstellte Verknüpfungen geboten. SweetWikis Ontologiemetamodell unterscheidet nicht zwischen Klassen, Instanzen, Datentyp- oder Objekttypeneigenschaften - alles wird auf Tags abgebildet. Dies hat die Konsequenz, dass Informationen ebenfalls nur über Tag-Suche oder direkt über SparQL abgerufen werden können. Informationen, die über mehrere Seiten verteilt sind, können so nicht abgefragt werden. In Tabelle 1 wird der Vergleich der verschiedenen Semantic Wiki Projekte kurz zusammengefasst.

Das von uns vorgeschlagene Semantic Wiki *Maariwa* verwendet ein Ontologiemetamodell, das die zugrundeliegende WR-Sprache OWL-Light verbirgt, die Wiederverwendung erstellter Verknüpfungen im vorhandenen Wissensmodell ermöglicht sowie die Voraussetzung für eine seitenübergreifende semantische Suche liefert. Im nächsten Kapitel stellen wir die grundlegenden Konzepte Maariwas vor und skizzieren dessen Implementation..

3 Maariwa - Konzeption, Architektur und Implementierung

Maariwa ist ein Semantic Wiki Projekt der FSU Jena mit dem Ziel, ein erweitertes Wiki-konzept zu implementieren, das es erlaubt neben Wiki-Seiten parallel Ontologien aufzubauen, auf deren Basis verbesserte Navigations- und Suchfunktionen realisiert werden können, so dass die automatische Wiederverwendung und Zusammenführung der im Wiki enthaltenen Informationen ermöglicht wird. Die Ontologie soll den textuellen Inhalt des Wikis strukturieren, indem sie die „Kristallisationspunkte“ des im Wiki gespeicherten Wissens abbildet und damit Zugriffspfade zu den Inhalten des Wikis bietet. Die Zugriffspfade können mit MarQL, einer einfachen semantischen Abfragesprache, adressiert werden, um

den Inhalt des Wikis semantisch zu durchsuchen. Dieses Konzept bezeichnen wir deshalb als *ontologiebasiertes Webpublishing*. Die Zielgruppe Maariwas sind Communities ohne Expertise in formaler Wissensrepräsentation. Zusätzlich erleichtert Maariwa den Zugang durch die Verwendung eines WYSIWYG-Editors, der Textverarbeitung und Ontologiebearbeitung integriert. Ein vereinfachtes, auf die Anwendung in einem Wiki zugeschnittenes Ontologiemetamodell dient dabei als Ausgangspunkt der Implementierung.

3.1 Aufbau und Integration des Ontologiemetamodells

Das in einem Maariwa Wiki abgebildete Wissensmodell, soll eine einfache und effiziente Durchsuchbarkeit gewährleisten, wie sie z.B. in den folgenden Anfragen zum Ausdruck kommt:

- Welcher Wissenschaftler hat die Energie-Masse Äquivalenz entdeckt?
- Welche berühmten Physiker lebten im 19. Jahrhundert?
- Welche deutschen Städte haben mehr als 100.000 Einwohner?

Zu diesen Zweck muss das Ontologiemetamodell entsprechende Konstrukte bieten, auf deren Basis diese Anfragen formuliert werden kann. Zugleich müssen diese Konstrukte aber möglichst einfach strukturiert sein, um auch von Anwendern ohne WR-Kenntnisse verstanden werden zu können. Die genannten Anfragen lassen sich durch in Tabelle 2 abgebildete Konstrukte formulieren, die eine Teilmenge des OWL-Light Sprachumfanges bilden. Alle anderen Sprachelemente des OWL-Light Sprachumfanges werden vernachlässigt, da es in unserem Sinne keine Anfragen gibt, die diese ausnutzen. Abweichend

OWL Konstrukte	Maariwa Konstrukt
Klasse	Klasse
Datentypeigenschaft	Attribut
Objekteigenschaft	Beziehung
Klasseninstanz	Seite, die eine Klasse repräsentiert und von Individuen-Seiten instanziiert wird
subClassOf	Super- und Subklassenbildung, z.B. Säugetier (Superklasse) - Mensch (Subklasse)
Individual	Seite, die ein Individuum beschreibt und eine Instanz von ein oder mehreren Klassen bildet
instanziierte Datentypeigenschaft	Attributwert einer Individuum-Seite
instanziierte Objekteigenschaft	Beziehungswert einer Individuum-Seite

Tabelle 2: Abbildung der OWL Konstrukte im Maariwa Ontologiemetamodell

von OWL-Light, werden Datentypeigenschaften und Objekttypeigenschaften als Attribute und Beziehungen bezeichnet. Außerdem sind Attribute und Beziehungen keine globalen Objekte, sondern sie sind nur innerhalb der Klasse eindeutig, für die sie angelegt wurden.

Das ermöglicht gleichnamige Attribute und Beziehungen in unterschiedlichen, nicht verwandten Klassen und verhindert die für einen nicht WR-Experten schwer verständlichen Namenskonflikte. Attribute besitzen einen Datentyp und optional eine Einheit. Es werden nur drei Datentypen unterschieden: Text, Zahlen und Datumsangaben. Für Zahlen und Datumsangaben ist die Angabe eines entsprechenden Datentyps notwendig.

Konzeptionell wurde das Ontologiemetamodell Maariwas so in das Wikimodell integriert, dass WikiSeiten mit Konzepten der Ontologie annotiert werden können, um Seiteninhalte damit teilweise zu formalisieren. Klassen, Individuen und Mengen von Individuen werden jeweils durch eine Wiki-Seite beschrieben. Einer Seite, die eine Klasse beschreibt, werden Attribute, Beziehungen und Superklassen zugeordnet. Seiten die Individuen beschreiben, werden hingegen mit mindestens einer Klassen verbunden, deren Attribute und Beziehungen mit Werten gefüllt werden können. Um eine Menge von Individuen beschreiben zu können, muss eine Seite mit einem MarQL Ausdruck (vgl. 3.4) verknüpft werden. Links zwischen Seiten können mit Beziehungen typisiert werden. Klassen können für Individuenseiten auch als Tags verwendet werden, indem sie nur flach, ohne Hierarchie und ohne Beziehungen oder Attribute angelegt werden.

3.2 Integration semantischer Annotationen in die Anzeige und Seitenbearbeitung

Die Oberfläche des Editors in Maariwa ist so konzipiert, dass der zusätzliche Aufwand minimiert wird, der durch die Pflege der Ontologie entsteht. Doppelte Eingaben von Informationen als Text- und (Ontologie-)Konzepte werden weitestgehend verhindert,

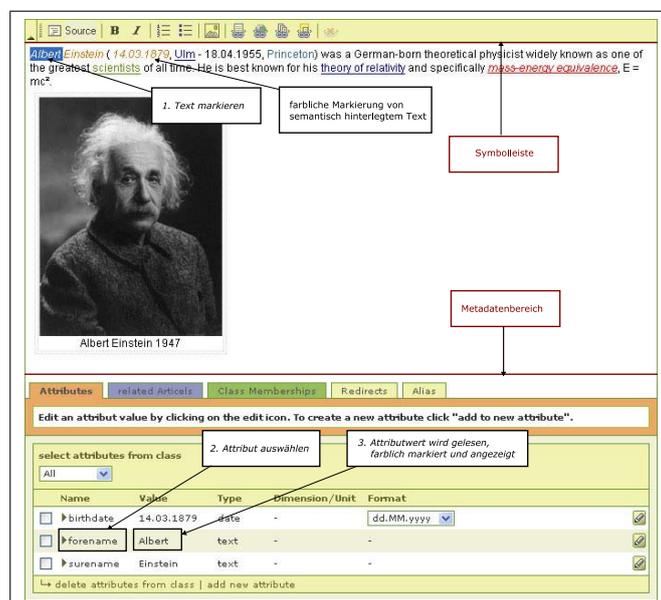


Abbildung 1: WYSIWYG Bearbeitung einer Maariwa Wiki-Seite mit Annotationen

indem Konzepte in vielen Fällen direkt aus dem Seitentext übernommen werden können. Andernfalls können Konzepte dialogbasiert angelegt werden. Damit wird es ermöglicht, textuelle Inhalte und Ontologien parallel zu entwickeln. Folgende zwei Varianten der Annotierung sind mit Maariwa möglich: Einerseits können Konzepte einer Ontologie aus dem textuellen Inhalt einer Wiki-Seite heraus erstellt werden und andererseits können zuerst Konzepte zugeordnet werden, die anschließend im Text der Wiki-Seite beschrieben werden. Schema- und Instanzdaten können immer parallel bearbeitet werden, ohne dass die Bearbeitung einer Seite unterbrochen werden muss (vgl. Abb. 1).

Die in Maariwa verwalteten Annotationen werden direkt in der Wiki-Seite visualisiert (vgl. Abb. 2). Unterschiedliche Farbmarkierungen visualisieren dabei die Semantik einer Verknüpfung. Sowohl Text, der Attributwerte enthält, als auch Links, die Beziehungen repräsentieren, werden farblich hervorgehoben.

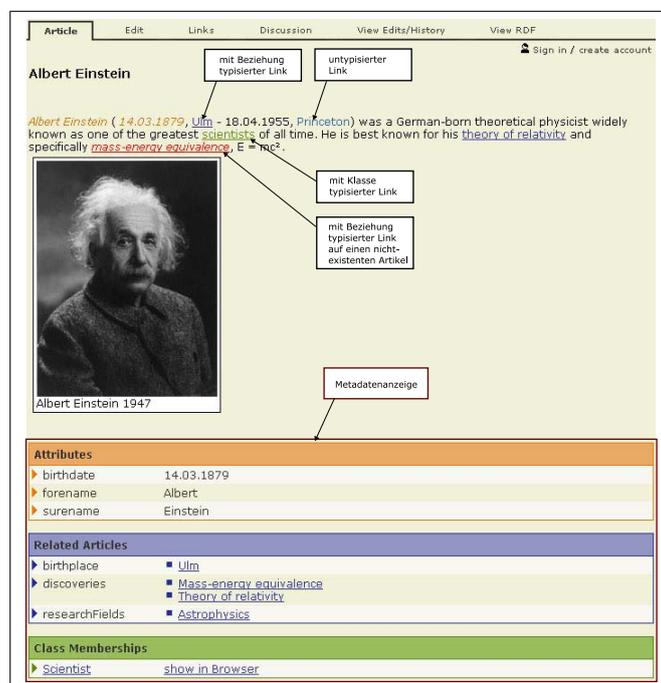


Abbildung 2: Ansicht einer semantisch annotierten Wiki-Seite in Maariwa

Zusätzlich zeigen Tooltips (eingblendete Infoboxen) die Semantik eines annotierten Textfragmentes. Ein einblendbarer Klassenbrowser zeigt, an welcher Stelle eine verknüpfte Klasse in der Ontologie formal eingeordnet wurde und ermöglicht Navigation und einfachen Zugriff auf Instanzen. Für jede Wiki-Seite kann über eine eigene URL die RDF/XML Repräsentation aller verknüpften Ontologiekonzepte aufgerufen werden. Auf diese Weise sind RDF-Daten einzeln exportierbar. Export und Import vollständiger Ontologien wird ebenfalls unterstützt.

3.3 Ontologieversionierung

Einem möglichen Schaden, der durch Missbrauch eines ungeschützten Schreibzugriffs auf Wiki-Seiten entstehen kann, wird in üblicher Weise durch eine Versionierung der Wiki-Seiten vorgebeugt. Maariwa überträgt dieses Prinzip auch auf die verwalteten Ontologien. Die Versionierung verfügt über zwei Ebenen: Instanzenebene und Schemaebene. Die Schemaebene umfasst alle Änderungen die an Klassen, Beziehungen und Attributen vorgenommen wurden, während die Instanzenebene alle Änderungen an Individuen mit ihren Attribut- und Beziehungswerten einschließt. Diese Ebenen sind miteinander gekoppelt, weil jede Schemaänderung Auswirkungen auf Instanzen haben kann. Aus diesem Grund umschließt eine Schemaversion zusätzlich alle Änderungen auf der Instanzenebene, die seit der letzten Schemaänderung auftraten. Jede Version einer Seite umfasst neben den Änderungen am Seitentext auch die Änderungen an den zugeordneten Konzepten.

3.4 MarQL - eine semantische Abfragesprache

Die Syntax der RDF-Abfragesprache SparQL [PS07] spiegelt die Charakteristik des RDF-Datenmodells wieder: RDF-Daten werden in Form von Triples repräsentiert und können zusammen im Dokument als Graph interpretiert werden. SparQL traversiert diesen Graphen und liefert als Ergebnis diejenigen Knoten zurück, die die in der Abfrage gestellten Bedingungen erfüllen. Weil die semantisch ausdrucksstärkeren Sprachen RDF(S) und OWL auf dem RDF-Datenmodell aufsetzen, kann SparQL auch für diese Sprachen als Abfragesprache verwendet werden, jedoch ohne deren Konstrukte gesondert zu unterstützen. MarQL ist eine spezialisierte Abfragesprache für Ontologien, die auf dem von Maariwa verwendeten Metamodell aufbauen. Die MarQL Syntax basiert nicht auf RDF-Tripeln, sondern setzt direkt auf der höheren Abstraktionsebene von Ontologieelementen wie z.B. Klassen, Attributen und Beziehungen auf. Die RDF-Kodierung der dabei abgefragten Daten bleibt verborgen. Die MarQL Syntax ist wesentlich kompakter als SparQL, aber auch unflexibler, da sie nur für bestimmte Anfragearten definiert wurde. Ein MarQL Ausdruck liefert stets eine Menge von Wiki-Seiten zurück, die Individuen beschreiben, für die der Ausdruck gültig ist.

Die Struktur der MarQL Abfragesprache soll an einigen einfachen Beispielen erläutert werden. Der MarQL Ausdruck *Scientist.institution.location.country = Germany* referenziert Wiki-Seiten über Wissenschaftler, die an einer in Deutschland gelegenen Institution arbeiten. In diesem Beispiel ist *Scientist* eine Klasse mit einer Beziehung *institution*. Beziehungen lassen sich in einem Pfad (dem linken Teil des Ausdrucks) durch einen Punkt getrennt, kaskadieren. So sind die Beziehungen *institution* und *location* miteinander verknüpft. Das bedeutet, dass es eine Klasse geben muß, die gleichzeitig die Zielklasse von *institution* ist und selbst eine Beziehung *location* besitzt. Nach dem gleichen Prinzip sind *location* und *country* verknüpft, mit dem Unterschied, dass *country* entweder ein Attribut oder eine Beziehung sein kann und *Germany* damit entweder ein Attributwert oder ein Individual ist. MarQL unterstützt neben logischen Operatoren auch String- und Vergleichsoperatoren. So liefert die Abfrage *City.population ≥ 100.000* eine Liste aller

Wiki-Seiten, die Städte mit mehr als 100.000 Einwohner beschreiben. Genauso kann mit $Scientist.birthdate < 1.1.1900 \text{ AND } Scientist.birthdate \geq 1.1.1800$ eine Liste aller Wissenschaftler des 19. Jh. erstellt werden. Der kleinste MarQL Ausdruck ist ein einfacher Klassenname. Ein solcher Ausdruck liefert eine Liste aller Instanzen einer Klasse und entspricht damit einem einfachen *Tag*.

3.5 Implementierung

Maariwa setzt nicht auf etablierten Wikilösungen auf, sondern stellt eine vollständige, auf Java basierende Eigenentwicklung dar. Den Anwendungskern Maariwas bildet eine Serviceschicht, die Wikifunktionen in Form von lose gekoppelten Diensten verwirklicht (vgl. Abb. 3). Neben der Datenverwaltung und Versionierung der Wiki-Seiten und den mit ihnen verbundenen Ontologiekonzepten, sind die Stichwortsuche und die semantische Suche ebenfalls als Dienste realisiert. Die MarQL Anfragen werden intern in SparQL Anfragen übersetzt. Als Datenspeicher kommt ein relationales DBMS zum Einsatz. Die Serviceschicht speichert Objekte, die ein Ontologiekonzept darstellen in verschiedenen Triplestores und alle anderen Objekte in eigenen Datenbanktabellen.

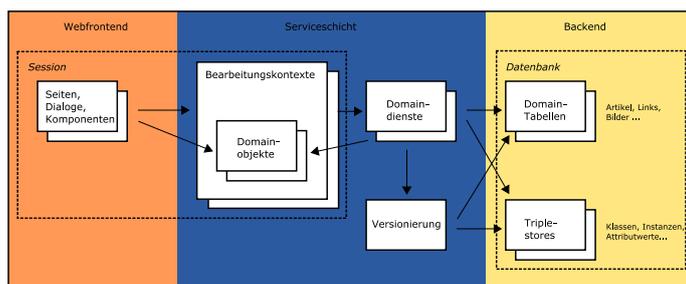


Abbildung 3: Maariwa Architektur

Die Serviceschicht wird von den Komponenten des Webfrontends verwendet, die zusammen die Benutzeroberfläche Maariwas bilden. Webserver und Browserclient kommunizieren teilweise asynchron über AJAX [Gar05], um eine flüssigere und interaktivere Bedienbarkeit zu gewährleisten.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Semantic Wiki Ansatz vorgestellt, der ein vereinfachtes, auf den Einsatz in einem Wiki zugeschnittenes Ontologiemetamodell mit einem auf aktuellen Browsertechnologien basierendem Bedienkonzept und einer einfachen semantischen Abfragesprache kombiniert, um Anwenderkreisen ohne Expertenwissen die Leistungsfähigkeit eines Semantic Wikis zu erschließen. Im Zentrum des Ansatzes stehen ein intuitives

Bedienkonzept und die Minimierung des Annotationsaufwands. Im Gegensatz zu bestehenden Ansätzen, können mit Maariwa Wikiseiten samt Annotationen und Links nach dem WYSIWYG Prinzip bearbeitet werden. Das Ontologiemetamodell erlaubt neben der Formulierung von Zugriffspfaden auf Wiki-Seiten auch die Wiederverwendung von bereits erstellten Verknüpfungen im Wissensmodell. Dies beschleunigt den Aufbau des Wissensmodells, senkt den Annotationsaufwand und es entstehen Vorlagen, die die Annotation neuer Seiten erleichtern. Die einfache Abfragesprache MarQL nutzt die in Maariwa erstellten Annotationen und die darin enthaltenen Zugriffspfade zur Realisierung einer semantischen Suche. Die Syntax von MarQL ist sehr kompakt. Semantische MarQL-Anfragen können so mit einfachen Schlüsselwortanfragen konkurrieren und bieten daher eine echte Alternative zu diesen. Das Ontologiemetamodell und die Abfragesprache sind so aufeinander abgestimmt, dass sie Annotationen mit unterschiedlicher semantischer Stärke unterstützen. Einfache Stichwörter ohne Hierarchie (Tags) sind damit genauso möglich wie komplexe Ontologien mit Beziehungen und Attributen. Abhängig vom Einsatzgebiet des Wikis und dessen inhaltlichen Strukturierbarkeit, können auf diese Weise Annotationen von unterschiedlich starker Semantik verwendet werden.

Maariwas Konzept ist an vielen Stellen ausbaufähig. Zu den derzeit bearbeiteten Erweiterungen zählen die direkte Integration von bestehenden Metadaten in den Seitentext, z.B. in Form von Tabellen, um sich selbst aktualisierende Übersichten und Zusammenfassungen zu realisieren. Eine stufenweise Skalierung der semantischen Ausdrucksstärke des Ontologiemetamodells ist ebenfalls in Arbeit, damit Anwender semantische Annotationen abhängig von ihren Erfahrungen und Kenntnissen einsetzen können. Formale Wissensmodelle können in Semantic Wikis sinnvoll aufgebaut und eingesetzt werden, um textuelle Inhalte zu strukturieren und das darin enthaltene Wissen maschinenverwertbar abzubilden. Für eine Community, die ein Semantic Wiki einsetzt, ist von entscheidender Bedeutung, wie das Aufwand/Nutzen Verhältnis für die Erstellung semantischer Annotationen ausfällt. Dabei ist nicht die Frage ob der formale oder der textuelle Inhalt im Vordergrund steht, sondern wie die Erstellung textueller und formaler Inhalte am effizientesten verzahnt werden können, so dass diese Unterscheidung obsolet wird.

Literatur

- [ACFLGP01] J. C. Arpírez, O. Corcho, M. Fernández-López und A. Gómez-Pérez. WebODE: a Scalable Workbench for Ontological Engineering. In *1st Int. Conf. on Knowledge Capture (KCAP01)*. Victoria, Canada, Oktober 2001.
- [BG04] D. Brickley und R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3c recommendation, RDF Core Working Group, W3C, 2004.
- [BG06] M. Buffa und F. Gandon. SweetWiki: semantic web enabled technologies in Wiki. In *Proc. of the 2006 international symposium on Wikis*, Seiten 69–78, 2006.
- [BLHL01] T. Berners-Lee, J. Hendler und O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, Mai 2001.

- [CCT04] S. E. Campanini, P. Castagna und R. Tazzoli. Platypus Wiki: a Semantic Wiki Wiki Web. In *Semantic Web Applications and Perspectives, Proc. of 1st Italian Semantic Web Workshop*, Dezember 2004.
- [DRR⁺05] B. Decker, J. Rech, E. Ras, B. Klein und C. Hoecht. Self-organized Reuse of Software Engineering Knowledge supported by Semantic Wikis. In *Workshop on Semantic Web Enabled Software Engineering (SWESE), at ISWC 2005*, 2005.
- [DST06] Karsten Dello, Elena Paslaru Bontas Simperl und Robert Tolksdorf. Creating and using Semantic Web information with Makna. In *Proc. of the 1st Workshop on Semantic Wikis – From Wiki To Semantics*. ESWC2006, June 2006.
- [FFR97] A. Farquhar, R. Fikes und J. Rice. The Ontolingua server: A tool for collaborative ontology construction. *Int. Journal of Human-Computer Studies*, 46(6):707–727, 1997.
- [Gar05] J. J. Garrett. Ajax: A New Approach to Web Applications, 2005. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- [GH06] Scott Golder und Bernardo A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Sciences*, 32(2):198–208, April 2006.
- [KVV05] M. Krötzsch, D. Vrandečić und M. Völkel. Wikipedia and the semantic web - The missing Links. In *Proc. of the 1st Int. Wikimedia Conf., Wikimania*, 2005.
- [LC01] Bo Leuf und Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.
- [MM04] Frank Manola und Eric Miller. RDF Primer. W3C recommendation, W3C, Februar 2004. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
- [MvH03] Deborah L. McGuinness und Frank van Harmelen. OWL Web Ontology Language: Overview. Working draft, World Wide Web Consortium, March 2003.
- [ODM⁺06] Eyal Oren, Renaud Delbru, Knud Möller, Max Völkel und Siegfried Handschuh. Annotation and Navigation in Semantic Wikis. In *Proc. of the 1st Workshop on Semantic Wikis – From Wiki*, Workshop on Semantic Wikis. ESWC2006, June 2006.
- [PS07] Eric Prud'hommeaux und Andy Seaborne. SPARQL Query Language for RDF (Working Draft). Bericht, W3C, March 2007.
- [SGW05] S. Schaffert, A. Gruber und R. Westenthaler. A Semantic Wiki for Collaborative Knowledge Formation. In *Semantics 2005, Vienna, Austria*, November 2005.
- [Sou04] Adam Souzis. Rhizome Position Paper. In *Proc. of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, September 2004.
- [SRKK97] B. Swartout, P. Ramesh und T. Russ K. Knight. Toward Distributed Use of Large-Scale Ontologies. In *Symp. on Ontological Engineering of AAAI. (Stanford, California, March)*, 1997.
- [Van05] Thomas Vander Wal. Folksonomy Explanations, 2005. <http://www.vanderwal.net/random/entrysel.php?blog=1622>.
- [VKV⁺06] M. Völkel, Markus Krötzsch, D. Vrandečić, H. Haller und R. Studer. Semantic Wikipedia. In *Proc. of the 15th int. conf. on World Wide Web, WWW 2006, Edinburgh, Scotland*, Mai 2006.