Semantic Need: Guiding Metadata Annotations by Questions People #ask

Hans-Jörg Happel

FZI Research Center for Information Technology, Karlsruhe, Germany,
happel@fzi.de

Abstract. In its core, the Semantic Web is about the creation, collection and interlinking of metadata on which agents can perform tasks for human users. While many tools and approaches support either the creation *or* usage of semantic metadata, there is neither a proper notion of metadata need, nor a related theory of guidance *which* metadata should be created. In this paper, we propose to analyze structured queries to help identifying missing metadata. We conduct a study on Semantic MediaWiki (SMW), one of the most popular Semantic Web applications to date, analyzing structured "ask"-queries in public SMW instances. Based on that, we describe *Semantic Need*, an extension for SMW which guides contributors to provide semantic annotations, and summarize feedback from an online survey among 30 experienced SMW users.

1 Introduction

Berners-Lee et al. [3] envisioned a *Semantic Web* populated by machine-understandable metadata based on which agents can reason and act to fulfill tasks for human users. Accordingly, one can distinguish two different roles: the *users* and the *providers* of semantic metadata.

Semantic Web research has addressed both roles and their corresponding work processes to a considerable extent. The usage of semantic metadata is supported by various tools ranging from semantic web service frameworks to ontology-based information retrieval systems. The creation and provision of semantic metadata has been studied in terms of manual and (semi-)automatic annotation systems (e.g. [8]) and with respect to exposing existing structured content on the Semantic Web (e.g. [4]). Surprisingly, only few research has studied topics such as incentives or methods for guiding the creation of semantic metadata so far. Since the provision of semantic metadata remains a costly process, several authors thus call for better means to "support users in the creation of metadata" [6][p. 148] and "to create incentives for annotations" [8][p. 198].

In this paper we propose to guide metadata provision by actual *metadata needs*. In previous research [11], we coined the term *Need-driven Knowledge Sharing* (NKS) to outline a framework connecting the usage and provision of information. We describe how NKS can be applied on the Semantic Web, taking *Semantic MediaWiki* (SMW) as a concrete example.

After introducing NKS, we present two heuristics for identifying missing annotations in SMW and describe their application in an exploratory empirical study with SMW installations running on the public internet. Then we present the implementation of *Semantic Need*, an extension for SMW which uses structured queries to guide users in contributing semantic metadata. In a second study, we asked 30 experienced SMW administrators to provide feedback on our general concept and on Semantic Need in particular. Based on the analysis of this data, we discuss further improvements and application scenarios of our approach.

2 Need-driven Knowledge Sharing

Ultimately, the Semantic Web can be seen as a specialized system for sharing codified information. As when sharing texts and documents, users and providers of information are separated due to the asynchrony of the technology, resulting in reduced motivation and contribution [12]. To address this, we developed the concept of *Need-driven Knowledge Sharing* (NKS) [11].

It is based on the assumption that information needs re-occur over time and across different information seekers, and can thus be used as a means to guide the creation and improvement of information. NKS rejects viewing information sharing as a linear process where all information has to be created prior to any request. In turn, it embraces that an information repository is never 100% *complete*, but grows and evolves over time. This perspective acknowledges the real world experience that individual requests might even fail to deliver any appropriate result, if some information is not yet known to the repository [13].

In a similar fashion, the logical formalisms underlying the Semantic Web share that "information [...] is in general viewed as being incomplete" [2, p. 68] and thus make a so-called *Open World Assumption* (OWA). In opposite to "closed world"-systems such as relational databases, facts that cannot be derived are not considered false but (yet) unknown under the OWA. Thus, a semantic knowledge base (*KB*) usually describes only a limited subset of what is considered true in a domain (see Fig. 1) and might grow over time.

A *KB* can be generally considered as a set of logical statements or axioms[1]. Such axioms might be used to state so-called terminological knowledge which describes classes and properties of the domain (i.e. "Professor is a subclass of Teacher") or about named individuals (i.e. "Rudi Studer is a Professor")[2]. If a *KB* cannot answer a request that is considered to have true results, this can either be due to missing assertions [2, p. 68] but also due to an incomplete specification of the terminology.

Although this evolutionary nature of captured knowledge is a fundamental principle underlying the Semantic Web, there do not exist appropriate methods providing guidance on how a knowledge base should evolve – i.e. which

[1] In RDF these axioms are called triples [16]

[2] The terminological and assertional part of a *KB* are usually referred to as TBox, respectively ABox [2, p. 46].

axioms should be added to satisfy information needs. We thus propose to use structured queries for this purpose. While there is no universal definition of structured queries, we consider so-called *conjunctive queries*[3] [14, p. 294] which are composed of conjunctive query atoms. These atoms may contain variables (i.e. “$Professor(x) \wedge worksAt(x, y)$”) which will be assigned concrete instance values from the KB if suitable results can be derived from the axioms in KB. Formally, a query q can be *satisfied* by a knowledge base KB, if $\exists\mu : KB \models \mu(q)$. The function μ maps every variable of the query to the name of an individual, ensuring that only known individuals are returned by a query [14, p. 295].

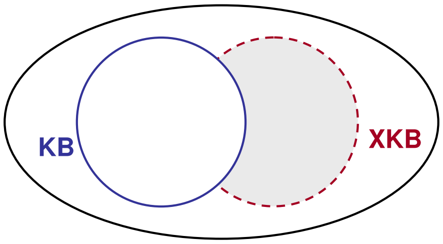


Fig. 1. KB denotes the set of all axioms in the knowledge base. XKB denotes the set of all axioms which have to be added to the KB to satisfy all structured queries.

We choose $QBox$ as the set of all structured queries that have been formulated against a knowledge base KB. Due to the inherent incompleteness of the KB, we expect that there is a set of *unsatisfied* queries UQ ($UQ \subseteq QBox$) for which holds: $\neg\exists\mu : \forall q \in UQ : KB \models \mu(q)$. UQ' is the subset of UQ, for which true results can be assumed[4]. We thus choose a set of logical axioms XKB such that $\exists\mu : \forall q \in UQ' : XKB \cup KB \models \mu(q)$. We assume that KB and $KB \cup XKB$ are *consistent* knowledge bases. Note that XKB thus loosely corresponds to the set of axioms filling the “semantic gap between supply and demand on the Semantic Web” as described by [15].

Finally, we choose a set of *partially unsatisfied* queries PUQ' ($PUQ' \subseteq UQ'$) by requiring that $\exists\mu : \forall q \in PUQ' : \exists atom \in q : KB \models \mu(atom)$. We consider PUQ' a particularly relevant subset of the $QBox$, since in opposite to queries in $UQ \setminus PUQ'$, queries in PUQ' (“PUQs”) have at least one query atom that can be satisfied from KB. Using PUQs, axioms contributing yet missing knowledge can thus be related to existing KB statements.

[3] In particular, we only consider the case of “DL-safe” conjunctive queries in this paper – i.e. we do not allow for non-distinguished variables in query atoms.

[4] For instance, a query for “All volcanoes in Karlsruhe” would not be contained in UQ' since there cannot be any true result (at least if we consider the real world as our domain).

3 Semantic Need Heuristics

We now want to investigate if "semantic gaps" as described in the previous section really occur on the Semantic Web. Since query data is not widely available for analysis, we decided to analyze Semantic MediaWiki (SMW) installations, since they contain persisted structured queries on Wiki pages (so called "inline queries").

In the following section, we thus give a brief introduction to SMW. We then apply the NKS concept to SMW, describing *incomplete* and *sparse result sets* as two heuristics for identifying PUQs. Finally, we apply these heuristics to structured queries extracted from public SMW installations.

3.1 Semantic MediaWiki

Semantic MediaWiki (SMW)[5] is an extension to the widespread MediaWiki engine[6]. It allows users to semantically annotate content on Wiki pages such that data can be exported and queried in a structured way.

- A list of conditions (basically categories and property values but also named instances) which should be matched against the knowledge base to constrain the result set
- A list of printout statements from which values should be contained in the result set

PUQs, as introduced in section 2, may either lead to incomplete or sparse result sets for queries in Semantic MediaWiki. We will now elaborate both cases in more detail and introduce heuristics for identifying axioms that should be added to the *KB* to satisfy PUQs.

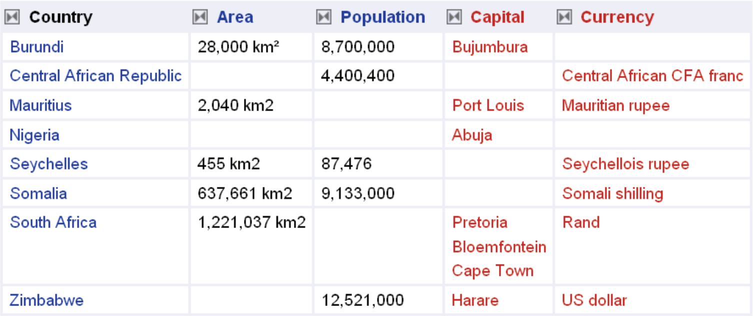
3.2 Incomplete Result Set

An incomplete result set denotes the situation that an *expected* result is not returned by a structured query. There are several reasons why this can be the case. First, a result instance might no yet be captured – e.g. a query for all instances of the class employee (`[[Category:Employee]]`) would not yield employees that are not yet known to the system. Second, instance annotations might be incomplete – e.g. a query for all employees with a salary >40.000 (`[[Category:Employee]][[salary::>40000]]`) would not yield `Employee` instances that lack any information about their salary.

Clearly, it is not obvious to decide if a given result set is incomplete. One option could be to leverage ontological background knowledge such as cardinality statements on properties. However, such statements are only possible in more feature-rich formalisms such as OWL, but not in RDF or SMW. Thus, another option is to heuristically infer missing results. In the following we present one particular heuristic for this purpose.

[5] http://www.semantic-mediawiki.org

[6] http://www.mediawiki.org



Country	Area	Population	Capital	Currency
Burundi	28,000 km^2	8,700,000	Bujumbura	
Central African Republic		4,400,400		Central African CFA franc
Mauritius	2,040 km2		Port Louis	Mauritian rupee
Nigeria			Abuja	
Seychelles	455 km2	87,476		Seychellois rupee
Somalia	637,661 km2	9,133,000		Somali shilling
South Africa	1,221,037 km2		Pretoria Bloemfontein Cape Town	Rand
Zimbabwe		12,521,000	Harare	US dollar

Fig. 2. An example of a sparse result set

Near matches As stated before, structured queries often contain multiple conditions to select particular subsets of an ontology class. The previously mentioned query for employees with a certain salary is an example for this. We define *near matches* as instances in the knowledge base which are potentially relevant results for a given query, but which do not appear in its result set due to missing semantic metadata.

To identify such cases, we only consider queries with at least two conditions. Technically, a candidate "near match" has to match at least one condition of a query and must not match at least one other condition, for which it lacks any annotation. This is to avoid considering instances which are properly described (e.g. an employee with a salary of 30.000, which does not match the query by purpose).

Near matches can thus help indicating missing annotations that prevent instances from appearing as a query result. The underlying assumption is, that these instances potentially could match the information need if metadata would have been properly annotated. Accordingly, we consider them *near matches* and assume that this might offer valuable insights on required metadata to people contributing to a knowledge base.

3.3 Sparse Result Set

In SMW-QL, a cell in the result set will remain empty by default if there is no appropriate binding for that variable (see Fig. 2)[7]. We define *sparse result set* as a case, when at least one cell in a result set remains empty.

Missing Result Values For SMW queries, empty cells can be considered an unsatisfied information need, since the query requests a variable binding which can

[7] Note that e.g. in SPARQL the default behaviour will not show the entire result set tuple, if at least one variable can not be bound. This default behaviour can be changed using the $OPTIONAL$ modifier [16]. However, in this case, we would end up with an *incomplete result set* as discussed before.

not be satisfied from the knowledge base. Thus, we define *missing result values* as a heuristic to infer missing annotations. They can easily be derived by simply counting empty cells in query result sets. For the maintainer of a Wiki page it might be interesting to know which printout statements are missing on a particular page in order to help delivering additional information for queries.

3.4 Public SMW Analysis

We now investigate if “missing result values” and “near matches” can be useful heuristics to identify PUQs in real-world settings. Since SMW is a popular MediaWiki extension, there exists a large number of publicly accessible installations which we could use for this purpose.

Design To check our heuristics we follow the basic research interests how many *missing result values* respectively *near matches* exist for real world structured queries. In terms of information need indicators, we will rely on the analysis of inline queries, since these are the only information needs in SMW which currently have a persistent representation. To select public SMW instances for analysis, we derived an initial list by consulting overview pages and search engines. By dismissing Wikis with only few semantic data (less than 3 queries and 250 annotations), we cut down our list from around 200 to 100. We then ruled out Wikis which were not accessible via a public API or difficult to crawl due to connection problems during the test runs of our evaluation tooling. Out of these, we randomly selected 26 Wikis, which we crawled. Due to the massive amount of data we decided to carry out deeper investigations on eight Wikis described in Table 1.

Sitename	Pages	ANN[8]	PG_{ANN}[9]	IQ[10]	IQ_{EC}	IQ_{ECPO}	IQ_{ECCJ}
CS Wiki (CS)	195	1.591	67	5	5	5	4
Eroge Wiki (ER)	340	1.853	182	3	1	0	0
HAR2009 (HA)	2.892	3.468	940	38	0	0	0
Historiographus (HI)	998	2.724	390	19	14	10	8
Mount Wiki (MN)	2.662	1.422	833	199	0	0	0
Protege Wiki (PR)	1.545	253	367	11	10	6	4
Sharing Buttons (SH)	122	590	18	7	0	0	0
territoile (TR)	1.801	3.135	502	3	1	1	1
Σ	10.564	15.036	3.299	285	31	22	17

Table 1. Overview of surveyed SMW installations

[8] Overall number of semantic annotations

[9] Number of pages containing at least one semantic annotation

[10] Number of inline queries. Further columns indicate subsets of IQ constrained by evaluation conditions as described in the text.

Process In order to retrieve data for our analysis, we wrote a crawler[11] which accesses the MediaWiki API. It extracts all semantic annotations and structured queries from the pages and stores it into a database. After retrieving the data, we applied further processing in order to restrict the number of queries for analysis. First, we chose an evaluation condition ("EC") which selects queries that a) are "ask"-queries (ruling out "show" queries) and that b) have either "table" (=default) or "broadtable" as output format (ruling out, e.g., RSS exports of query results). The number of queries satisfying the evaluation condition is shown in Table 1 as IQ_{EC}. In order to further align the set of queries to our analysis, we applied a final selection step. For the analysis of missing result values, we selected those queries that actually contain printout statements (IQ_{ECPO}). Accordingly, we selected only conjunctive queries for the analysis of near matches (IQ_{ECCJ}).

Overall, this processing resulted in 22 queries satisfying IQ_{ECPO} and 17 queries satisfying IQ_{ECCJ}. Due to overlaps of both sets (see Table 2 and 3) this results in 25 distinct queries[12]. As a first step of analysis, we derived the number of results for all queries. Since many queries were located on Template pages, the corresponding fields in Table 2 and 3 denote "n.a.", since the number of results would depend on the page embedding the template. Instead, we computed the number of instances for the `[[Category:]]` part of the query ($Results_{CAT}$).

Results

Missing result values Table 2 summarizes the analysis of the IQ_{ECPO} query set. We computed the number of missing result values (e.g. empty cells) in the result set. For queries on normal pages, this is the actual number. For queries on template pages, we summarized the number of missing result values across all instances ($Results_{CAT}$).

As it can be seen from the results, all queries on normal pages provide a complete result set. However, for queries on template pages, up to 63% of cells in the query result set were empty. To estimate if these empty cells were really due to missing information (instead of consciously ommitted), we manually investigated three empty cells for each of five different queries. It turned out, that only two of the 15 empty cells could not be considered missing information. This shows that queries lack result values to a considerable extent. In average, 16% of cells remained empty across all queries surveyed.

Near matches For the conjunctive queries, we first observed that all 17 queries under consideration consisted of exactly two conjunctions. In most cases, this is a category stametement combined with a restriction on one property (e.g. PR2: `[[Category:Plugin]] [[For Application::PAGENAME]]`). In order to derive near matches, we computed the number of instances which completly lack the annotation of the restricted property. The rationale behind this is, that these

[11] Available at http://www.teamweaver.org/wiki/index.php/MediaWikiTools

[12] See http://www.teamweaver.org/downloads/data/sneed/sneed-smw-queries.pdf

[13] Number of instances for the `[[Category:]]` part of the query

ID	Results	$Results_{CAT}$[13]	Empty cells	Printout requests	% Empty cells
CS1	n.a.	8	19	4	59%
CS2	n.a.	7	0	3	0%
CS3	n.a.	1	0	1	0%
CS4	n.a.	16	0	2	0%
CS5	7	7	0	4	0%
HI1	1	18	0	3	0%
HI2	28	65	0	2	0%
HI4	n.a.	18	27	3	50%
HI5	n.a.	65	22	2	17%
HI7	n.a.	24	60	4	63%
HI8	n.a.	4	1	3	8%
HI9	n.a.	35	6	4	4%
HI10	n.a.	15	3	4	5%
HI11	n.a.	14	2	4	4%
HI12	n.a.	15	9	4	15%
PR1	72	80	0	1	0%
PR2	n.a.	80	13	1	16%
PR3	n.a.	91	1	1	1%
PR5	n.a.	91	75	2	41%
PR6	n.a.	91	1	1	1%
PR4	n.a.	91	57	1	63%
TR1	70	102	0	1	0%
		Σ 938	Σ 296	Ø2,5	Ø16%

Table 2. Empty/missing result values for the surveyed queries

instances might qualify to appear in the query result set, once a correct value for the property is annotated.

As described in the last column of Table 3, up to 94% of instances lacked the annotation on the selection property in extreme cases. Again, we performed a deeper investigated on three near matches for each of five different queries. Out of these 15, five turned out be “false positives” - i.e. were lacking annotations by purpose. While near matches might thus not be a strict indicator for “missing” annotations, they are nevertheless a strong hint. On average, across all queries, a value of 22% turns out. This is a rather high number, considered that this rules out the instances from appearing in the results of the surveyed queries.

Although our analysis is based on a rather small set of queries, this selection can already help to identify up to 296 missing printout statements and up to 147 missing selection properties within the surveyed Wikis. Given the fact that we only analyzed around 9% of the overall inline queries (due to our evaluation conditions), this stresses the potential for using “missing result values” and “near matches” as heuristics for guiding semantic annotations.

ID	Results	$Results_{CAT}$[13]	Missing selection property	% Missing selection property
CS1	n.a.	8	6	75%
CS2	n.a.	7	0	0%
CS3	n.a.	1	0	0%
CS4	n.a.	16	4	25%
HI1	1	18	17	94%
HI2	28	65	10	15%
HI3	1	3	1	33%
HI4	n.a.	18	17	94%
HI5	n.a.	65	10	15%
HI6	n.a.	3	0	0%
HI7	n.a.	24	13	54%
HI8	n.a.	4	2	50%
PR1	72	80	8	10%
PR2	n.a.	80	9	11%
PR3	n.a.	91	0	0%
PR4	n.a.	91	18	20%
TR1	70	102	32	31%
		Σ 676	Σ 147	Ø22%

Table 3. Near matches for the surveyed queries

4 Semantic Need Implementation

In this section, we first discuss how semantic gaps in a knowledge base can be resolved. Then we present our prototypical implementation of Semantic Need as an extension for SMW and summarize results of a survey conducted among 30 SMW administrators.

4.1 Resolving Semantic Gaps

Semantic gaps in the knowledge base, as indicated in the previous paragraphs, can either be resolved by capturing or by sharing knowledge.

Capturing is necessary, if knowledge is not yet formalized at all. This can involve both, schema-level knowledge or data/annotations. Concerning annotations, "near matches" and "missing result values" can help identifying concrete properties which are not yet annotated for a knowledge base instance. Thus, users can be provided with an interface denoting all missing properties for a given instance as derived by these heuristics. Similarly, one can try to identify if "near matches" and "missing result values" are due to missing schema mappings. This denotes the case if query atoms do not correspond to existing categories or properties in the Wiki. This can either imply that parts of the domain knowledge are missing in the ontology of the Wiki, or it can be an indictor of synonyms – e.g. if a user asks for `[[Category:Worker]]` instead of `[[Category:Employee]]`. Thus, the system might assist users in finding candidate mappings to improve the ontology schema and thus help satisfying information needs.

Sharing knowledge can be done if information *is* already formally captured, but not available at query time, since it is hidden in a yet unknown or not accessible knowledge base. Information needs might thus be satisfied by either sharing (i.e. copying) semantic information into the queried knowledge base, or by introducing suitable mappings, which allow the query engine to retrieve semantic information from distributed spaces.

4.2 Semantic Need for MediaWiki

Our current implementation adresses the *capturing* of semantic *annotations*, while the *sharing* of semantic information and the provision *schema-level knowledge* are foreseen in the system design, but not yet realized. We also currently focus on so-called "inline queries" embedded in Wiki pages. We consider them the most relevant, since many end users might not be able or willing to formulate ad hoc structured queries on their own. Basic information about inline queries is stored in a "semantic query log" which includes the conditions and printout statements of the query. Due to space restrictions, we will skip details on the storage by now[14]. Based on the query log, a so-called *Need API* offers *metadata need* information such as "near matches" and "missing result values".

One consumer of such need information is the *Capturing UI*, a special user interface which allows *knowledge engineers, domain experts* or *end users* to contribute potentially missing facts to the knowledge base. We realized two different types of implementation so far. First, we provide "global" overview pages which list all queries – in particular those without results – and a Wiki-wide overview of pages and their missing annotations. Second, the same feature is applied to individual pages, resulting in an overview of missing annotations for a specific Wiki page. This can be considered a semantic counterpart to the MediaWiki page `Special:WhatLinksHere`, which helps users to find out how a Wiki page is *syntactically* embedded (i.e. linked).

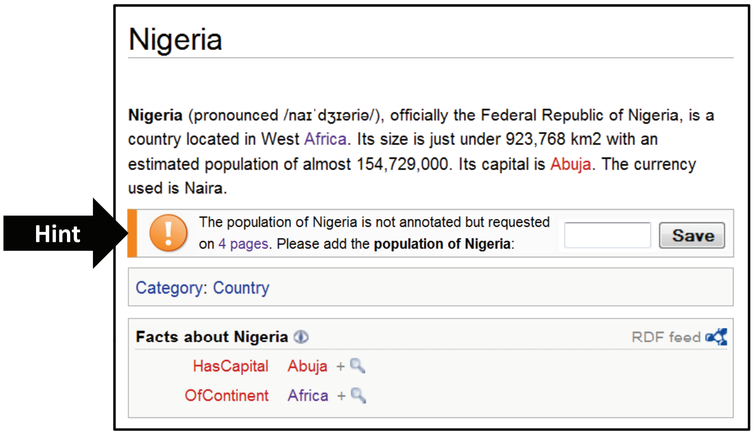
Although usability issues are not a core focus of this paper, we also thought about how to address actual end users who might contribute to the Wiki more directly (see Fig. 3). Besides this, several other ways to inform users about contribution possibilities can be imagined – including integration in Java-Script based annotation UIs[15], game-based interfaces (e.g. [17]) or identifying and approaching potential contributors directly (e.g., by E-Mail).

4.3 Semantic Need Survey

While an initial implementation of the Semantic Need extension is already available, it is not yet robust enough for an evaluation in the field. We thus decided to evaluate the current version based on an expert survey among experienced SMW administrators, which we describe in the following.

[14] Initial (but more general) ideas have been presented in [9].

[15] Such as http://smwforum.ontoprise.com/smwforum/index.php/Help:Introduction_to_Advanced_Annotation_Mode



Nigeria

Nigeria (pronounced /naɪˈdʒɪəriə/), officially the Federal Republic of Nigeria, is a country located in West Africa. Its size is just under 923,768 km2 with an estimated population of almost 154,729,000. Its capital is Abuja. The currency used is Naira.

The population of Nigeria is not annotated but requested on 4 pages. Please add the **population of Nigeria**:

Category: Country

Facts about Nigeria RDF feed

HasCapital	Abuja +
OfContinent	Africa +

Fig. 3. In-page display and input form for missing annotations

Design and Process The main goal of the survey was to gather feedback on our current concept and its realization. We thus decided to include a small example scenario with screenshots of SMW and our extension. Since this requires a) prior knowledge of SMW, b) a holistic view of an existing SMW installation and its usage and c) results in a rather large questionnaire, our main target group consists of experienced SMW administrators rather than end users.

The questionnaire consists of five major components. Two parts address the problems of a *sparse* and *incomplete result set*, asking respondents about the frequency and severity of these issues. Another part deals with semantic annotation practices. People are asked how they find out missing annotations in a standard SMW. Afterwards, screenshots of Semantic Need are shown (including Fig. 3) and people are asked if they agree that Semantic Need might be effective to a) generally help maintaining annotations, b) focus annotation effort and c) motivate users to provide contributions. Two other parts of the survey address the usage context of SMW. We asked about the knowledge domain captured in the Wiki and the structure and content of the knowledge base.

The final questionnaire has 34 questions[16]. It was pre-tested by 5 persons resulting in some minor modifications and clarifications. To gather participants for the survey, we followed two strategies. Since we were interested in frequent SMW users, we advertised our survey on the official SMW user and developer mailinglists. Furthermore, we directly contacted 15 persons which are known to drive own SMW projects.

Results We received 30 complete answers. A majority of 15 answers came from Germany, 7 from the US while the remaining participants are scattered across eight different (mostly European) countries. Concerning their experience with

[16] See http://www.teamweaver.org/downloads/data/sneed/sneed-survey.pdf

SMW, 15 respondents describe themselves as "intermediate", 11 as "expert" and 4 as "novice". On average, they are using SMW for 2.3 years.

The knowledge domain captured in SMW is characterized as "fixed/standardized" in 8 cases, as a "generally open domain without many predetermined entities and properties" in 6 cases and as a mix of both options in 15 cases. Accordingly, the semantic data model is largely prescribed by Semantic Forms/Templates in 19 cases. Only 7 SMWs have an equal level of prescribed and ad hoc structure and another 4 rely mostly on free-form annotations. None of the Wikis surveyed do *not* use Semantic Forms/Templates at all. 12 people answered that no particular methodologies, practices or tools are used to maintain the semantic data, while 5 people claim to follow simple informal practices and 7 people implement changes based on more advanced measures such as scripts, documentation and team decisions. In 7 cases, the data stored in SMW is driven by the structure from external data and systems.

The problem of *sparse result set* was observed "often" or "sometimes" by 18 people, while 12 indicated "rarely" or "never". 15 people rate the issue as "not problematic" while 12 answered "somehow problematic". No one rated query result sparseness as "very problematic". In their free text justification, people made the point that the application context (4 answers) and the nature of the data itself (5) have an impact on if query result sparseness is an actual problem.

For *incomplete result sets*, 19 people answered to have observed the issue "often" or "sometimes" while 9 observed it "rarely" or "never". Furthermore, only 5 people consider the issue "not problematic", while 18 answered "somehow problematic" and 5 even "very problematic". This is stressed by the free text justifications in which 16 respondents repeated that query result incompleteness is a problematic issue. Key aspects are the "invisiblility" of the issue (which makes it worse than sparse query results) which is quantified if the dataset grows large: *"due to the nature of our wiki (IT company) it is hard to know when a query is incomplete. For example, there are hundreds of pages on servers so impossible to know when one or several are missing."*

We also asked how people would deal with finding out missing annotations for a particular Wiki page and clustered the free-form answers in four main categories. 6 answers suggest to make a comparison with annotations on similar Wiki pages. Related to that, 7 people would check the schema (i.e. properties) and forms related to that page. Another 4 people would do an analysis of the page text to identify additional content that could be formalized. Finally, 10 answers suggested to create specific ask-queries for this purpose. It turns out that *decisions* are a core part of this process – as one answer puts it: *"Write down a list of all the quantifiable data on the page. - Then decide if any of these are excessive in depth for most users. - In this case I would add part of africa, size, population, and currency."*

The global overview about Wiki pages and their missing annotations is generally appreciated in the survey. On a 5-point scale ranging from "strongly diagree" to "strongly agree", most respondents agree that this feature can be effective to maintain semantic annotations in SMW (8/18/2/2/0[17]). The agreement is slightly less on if it can help to guide annotation efforts towards most crucial information needs (5/18/6/1/0) and on if it can motivate users to provide missing annotations (9/13/5/2/1). The page-specific features of Semantic Need are even more appreciated. 15 respondents strongly agree that it can be effective to maintain semantic annotations in SMW (15/11/2/2/0). Concerning annotation guidance and user motivation, 26 respondents at least chose "agree" in both cases (12/14/3/1/0). Finally, 20 participants (66%) are interested in using the Semantic Need extension on their own Wiki.

Summarizing we can observe that SMW usage differs largely – ranging from prescribed data structures to more open, Semantic Web-inspired scenarios. While the first group argues that data quality and completeness is crucial in their case and thus considers missing annotations a serious problem, others stress the evolving nature of Semantic Web applications: *"I don't see this is a 'problem' - it's the way things are, always in flux, always perfecting, always coming to stasis. Law of Thermodynamics."* Semantic Need however, was considered helpful by both groups – either to help raising data quality or to provide guidance in less predefined settings.

5 Design Implications

In this section we reflect on our overall approach, the Semantic Need prototype and the data we have gathered to validate it. We identified a number of design parameters which we consider useful for our own future work but also for other people developing Semantic Web applications.

The need for need specification: Surprisingly the Semantic Web, which is all about expressing knowledge in a formal way, has not yet done much in terms of *expressing information needs*. We thus consider an ontology which helps users to characterize their information needs more precisely (e.g., duration or urgency) helpful. This should be complemented by appropriate *semantic query log* standards and storage mechanisms.

Data quality modeling: Several people in our survey argued that some of the identified "problems" might just be intended states: *"I have seen instances in which sparseness was intentional, i.e., a query is created specifically to show the absence of data - it can be useful in the right circumstances."*. Thus, the precision of information need heuristics depends on assumptions and background knowledge of the application domain. While some people suggested to specify properties e.g. as *mandatory*, these features are either not part of the knowledge representation formalism[18] or hardly used (such as cardinality constraints).

[17] Amount of answers stating: strongly agree/agree/neutral/disagree/strongly disagree

[18] In the case of SMW, some are artificially enforced by the Semantic Forms extension

A scattered Semantic Web: While we focus on single SMW instances in this paper, we think that our concepts are also useful on a larger scale. The Semantic Web is decentralized and heterogeneous and so are the “semantic intranets” of some of our survey partcipants: *“because my data comes from ExternalData, I wouldn’t ENTER those properties on the Wiki itself, but this extension would help us to go back to the source and add it.”.* A Semantic Need-enabled SMW could thus pull data (and information needs) from external systems, capture mappings and share data in external places. While an interconnected set of SMW instances would be a straightforward idea, we also think that our approach could be implemented in other Semantic Web applications, given a set of standards for information need description and exchange.

Ontology evolution vs. maturing: Much research on ontology dynamics has a technical spin under the label of *ontology evolution.* However, our survey results show that data integrity is not the only concern in this field. While the Semantic Forms extension, which helps to “freeze” parts of the data structure, has been quickly adopted by many SMW administrators, the process for dealing with emerging entities is not yet well adressed. We thus argue that methodological considerations such as the *ontology maturing* concept [5] should be given more attention in the design of Semantic Web applications.

6 Related Work

Since our main goal is to guide the creation of semantic metadata, work in the area of *semantic annotation* is partly relevant for us. However, most systems, such as CREAM [8] are inspired by the linear perspective of the information foodchain [7] and thus drive the annotation process by the pre-defined ontology structure. While guidance and incentives for annotation are considered major open issues ([8][p. 198], [6][p. 148]), we are not aware of other approaches considering queries for guiding the annotation process.

The probably most directly related work to ours is a recent study by Mika et al. [15]. Similar to our NKS framework, they contrast and connect the perspectives of semantic metadata provision and usage. However, they use a slightly different approach by taking keyword queries from Yahoo query logs and mapping those to entity/property pairs, which they compare to actual semantic metadata from DBPedia. Thus, the work is primarily of descriptive nature and does not suggest actual technical solutions. The evaluation track of the SemSearch workshop[19] and the evaluation campaign of the SEALS project [18] are recently emerging initiatives to capture and analyze structured query data.

The work presented in this paper might also be considered related to approaches for maintaining or *gardening* semantic knowledge bases. A particular example is the Semantic Gardening Extension[20] for SMW. However, it is focused

[19] http://km.aifb.kit.edu/ws/semsearch10/

[20] http://smwforum.ontoprise.com/smwforum/index.php/Help:Semantic_Gardening_Extension

on knowledge base instances without properly defined ontology classes or ontology classes without instances. A *need dimension*, taking into account the actual usage of semantic data is currently not part of this work.

As for the core idea of driving knowledge sharing by user requests, the seminal Answer Garden system [1] deserves credit. While Answer Garden uses experts to filter and answer requests, so called "Collaborative Question Answer Systems" (such as Yahoo Answers[21]) and our Woogle system [10] embrace all users as potential contributors.

7 Summary

This paper has described three major contributions. First, we have argued for considering information needs – and in particular structured queries – as drivers for the process of creating semantic metadata. To this end we introduced the *Semantic Need* approach which guides contributors to create metadata which is of the most value for other users in the Semantic Web. Second, we introduced an extension for SMW as a proof-of-concept realization of this approach. While this stresses the general feasibility of our ideas, we think that a realization within a larger Semantic Web scope is possible as well (see also [9]).

Third, we conducted two empirical studies to validate our claims. Our analysis of public SMW installations shows, that the current application areas of Semantic Need – *missing result values* and *near matches* – occur in the surveyed dataset to a considerable extent and are thus of practical relevance. This is also stressed by the result of an expert survey among 30 experienced SMW administrators. Their feedback provides initial evidence that Semantic Need can be an effective tool to support the guided growth of semantic knowledge bases.

Beyond that, our framework and our empirical data will enable us to pursue further studies of that kind. Obvious directions would be to guide ontology schema evolution and mapping or query refinements based on information needs.

Acknowledgements

This work was partially supported by the THESEUS project, which is funded by the German Federal Ministry of Economics (BMWi) under grant 01MQ07019, and the GlobaliSE project, which is funded by the Baden-Württemberg Stiftung. Thanks go to Andreas Abecker, Markus Krötzsch, Sebastian Rudolph, Stephan Grimm, Athanasios Mazarakis and Heiko Haller for helpful feedback and to Paul Hübner and Hristo Valev for their implementation of the MediaWiki crawler and the Semantic Need extension.

References

1. Ackerman, M.S., Malone, T.W.: Answer garden: a tool for growing organizational memory. In: Proceedings of the ACM SIGOIS conference on Office information systems. pp. 31–39. ACM, New York, NY, USA (1990)

[21] http://answers.yahoo.com/

2. Baader, F., Nutt, W.: Basic description logics. In: Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.) Description Logic Handbook. pp. 43–95. Cambridge University Press (2003)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic Web. Scientific American 284(5), 34–43 (May 2001)
4. Bizer, C., Cyganiak, R.: D2r server-publishing relational databases on the semantic web (poster). In: International Semantic Web Conference (2006)
5. Braun, S., Schmidt, A., Walter, A., Nagypal, G., Zacharias, V.: Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In: Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007). CEUR Workshop Proceedings, vol. 273 (2007)
6. Decker, S.: Semantic web methods for knowledge management. Ph.D. thesis, University of Karlsruhe (2002)
7. Decker, S., Jannink, J., Melnik, S., Mitra, P., Staab, S., Studer, R., Wiederhold, G.: An information food chain for advanced applications on the www. In: Borbinha, J.L., Baker, T. (eds.) ECDL. Lecture Notes in Computer Science, vol. 1923, pp. 490–493. Springer (2000)
8. Handschuh, S.: Creating ontology-based metadata by annotation for the semantic web. Ph.D. thesis, University of Karlsruhe (2005)
9. Happel, H.J.: Growing the semantic web with inverse semantic search. In: 1st Workshop on Incentives for the Semantic Web (INSEMTIVE '08). pp. 1–12 (2008)
10. Happel, H.J.: Social search and need-driven knowledge sharing in wikis with woogle. In: WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration. pp. 1–10. ACM, New York, NY, USA (2009)
11. Happel, H.J.: Towards need-driven knowledge sharing in distributed teams. In: Proceedings of the 9th International Conference on Knowledge Management. pp. 128–139. JUCS (2009)
12. Happel, H.J.: Semantic need: An approach for guiding users contributing metadata to the semantic web. Int. J. Knowledge Engineering and Data Mining (to appear) (2010)
13. Happel, H.J., Mazarakis, A.: Considering information providers in social search. In: Proceedings of the 2nd International Workshop on Collaborative Information Seeking (CIS '10). pp. 1–5 (2010)
14. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
15. Mika, P., Meij, E., Zaragoza, H.: Investigating the semantic gap through query log analysis. In: International Semantic Web Conference. Lecture Notes in Computer Science, vol. 5823, pp. 441–455. Springer (2009)
16. Prud'Hommeaux, E., Seaborne, A.: SPARQL query language for RDF. World Wide Web Consortium, Recommendation REC-rdf-sparql-query-20080115 (Jan 2008)
17. Siorpaes, K., Hepp, M.: Ontogame: weaving the semantic web by online games. In: ESWC'08: Proceedings of the 5th European semantic web conference. pp. 751–766. Springer-Verlag, Berlin, Heidelberg (2008)
18. Wrigley, S.N., Reinhard, D., Elbedweihy, K., Bernstein, A., Ciravegna, F.: Methodology and Campaign Design for the Evaluation of Semantic Search Tools. In: Proceedings of the Semantic Search 2010 Workshop (SemSearch 2010) (2010)