Wikipedia and the Semantic Web
The Missing Links*

Markus Krötzsch, Denny Vrandečić, Max Völkel

Institute AIFB, University of Karlsruhe, Germany
{kroetzsch,vrandecic,voelkel}@aifb.uni-karlsruhe.de

Abstract. Wikipedia is the biggest collaboratively created source of encyclopaedic knowledge. Growing beyond the borders of any traditional encyclopaedia, it is facing new problems of knowledge management: The current excessive usage of article lists and categories witnesses the fact that 19th century content organization technologies like inter-article references and indices are no longer sufficient for today's needs.
Rather, it is necessary to allow knowledge processing in a computer assisted way, for example to intelligently query the knowledge base. To this end, we propose the introduction of *typed links* as an extremely simple and unintrusive way for rendering large parts of Wikipedia machine readable. We provide a detailed plan on how to achieve this goal in a way that hardly impacts usability and performance, propose an implementation plan, and discuss possible difficulties on Wikipedia's way to the semantic future of the World Wide Web. The possible gains of this endeavor are huge; we sketch them by considering some immediate applications that semantic technologies can provide to enhance browsing, searching, and editing Wikipedia.

1 Introduction

An important aspect of Wikipedia's utility is its use of modern technologies. Besides the obvious advantages of the Wiki principle for creating the knowledge base, one of the most important aspects for the daily usage of Wikipedia is the strong interconnection of its articles via hyperlinks. The ubiquity of such links in Wikipedia is one of the key features for finding desired information.

The reason for this is that – despite of its revolutionary editing mechanism and organization – Wikipedia's dedicated facilities for searching information are surprisingly primitive. Users often rely on full text search, article name or links for finding information. So it became common to create pages with the sole purpose of collecting links (*lists* of articles). A more structured approach with a similar functionality is Wikipedia's category system.

To illustrate the shortcomings of the current mechanisms for searching and browsing, assume we were looking for all articles about *James Bond movies from*

* This research was partially supported by the European Commission under contract IST-2003-506826 *Semantically Enabled Knowledge Technologies* (SEKT) and FP6-507482 (KnowledgeWeb). The expressed content is solely the view of the authors.

the 60s that were not starring Sean Connery. Clearly this information is present in Wikipedia – the problem is that it cannot be retrieved automatically. The user thus has to search through all articles of related topics, maybe finding a list of all possible movies and then reading each of the articles to find what she was looking for. It is not feasible to create lists of articles to answer even a fraction of all interesting queries, and consequently it is not likely that Wikipedia will ever treat such queries based on the current mechanisms of information organization[1].

Wikipedians already have come up with several ideas to further structure the vast amount of information available through Wikipedia, which lead to such diverse projects as Wikidata[2], Wikipedia DTD, Person DTD, Metalingo and WikiSpecies. We suggest a similar approach like Wikidata but more dynamic and – even more important – built upon mature data exchange *standards*.

The technique proposed in this paper aims at providing an extremely simple and low-tech way of augmenting Wikipedia with machine readable information that allows one to (internally or externally) implement all kinds of query answering systems to solve the above problem. We first give a short overview of the required basic semantic technologies in Section 2. Then we present our approach in Section 3. There we also focus on usability and performance, both of which we consider vital for the feasibility of any extension to Wikipedia. In Section 4 we present an implementation plan for gradually introducing the proposed functions to Wikipedia. Furthermore, the proposed extension of Wikipedia can also be exploited for the creation of new tools to browse, author, check for consistency, and visualize Wikipedia's content. Details on this are discussed in Section 5. Before we conclude our treatment with Section 7, we review related work in Section 6.

2 A jump start introduction to semantic technologies

Problems similar to the above have already been identified in the even larger context of the World Wide Web, motivating intensive research on appropriate semantic technologies. A major ideal of these developments has been the creation of the *Semantic Web*, "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [3].

Standards for the representation of data in the Semantic Web have been created, like the W3C's *RDF* [9] and *RDF Schema* [4]. RDF is a graph-centric approach which builds on the intuition that the fundamental aspects of a domain of interest can be modelled as binary relationships between objects. For example, one can state that "Sean Connery is an actor in the movie *Goldfinger*," where Connery and the movie are objects between which the relationship of being an actor holds.

[1] However, there already is a tendency to add lists and categories for all kinds of query-like combinations of features, including some that do only contain very few articles.

[2] `http://meta.wikimedia.org/wiki/Wikidata`

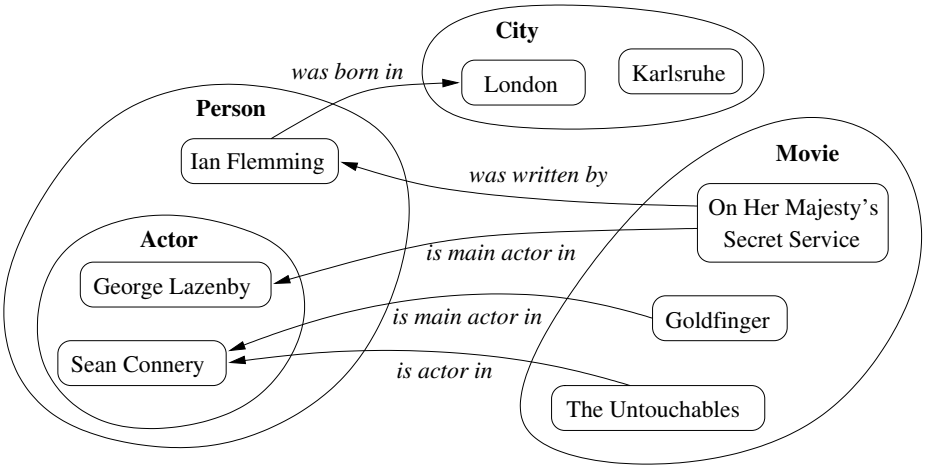

Fig. 1. Graphical representations of a RDF(S) specification are easily understood by humans. Individuals and their relationships (RDF) are denoted as nodes (rounded boxes) and arrows, while classes of individuals (RDFS) are displayed as closed areas.

In order to denote objects and relations as in the above example, one needs globally unique identifiers (URIs) for these entities. Luckily, objects in the Wikipedia already have such identifiers in form of their URL. Statements in RDF can then be expressed as triples of URIs: the first URI denotes the *Subject* ("Sean Connery"), the second gives the *Predicate* ("is actor in"), and the third specifies the object ("Goldfinger (movie)"). Novices to RDF are often confused by its unreadable RDF/XML syntax and overlook the simplicity of the RDF data model.[3] Finally, RDF also suggests visualization of specifications as labelled graphs of objects (nodes) and relations (edges). An example of this is given in Figure 1.

To allow for a more structured description of knowledge, RDF Schema was introduced as a simple *ontology language* on top of RDF. The RDF Schema type system knows classes for typing entities, and properties for typing relations between entities. Classes can be compared to Wikipedia's categories: they describe collections of objects and can be organized in a hierarchy. Figure 1 displays various RDFS classes and their relationships (e.g. *Actor* is a subclass of *Person*). As in Wikipedia, multiple inheritance and even cycles are allowed in the class hierarchy. A similar mechanism is available for typing relationships. We will explain the ideas behind this in Section 3.

While RDF and RDF Schema provide an important foundation for structuring knowledge, there are many interesting statements that cannot be expressed

[3] Yet, we admit that our introduction is somewhat oversimplified as so called blank nodes and more complex constructs like lists, reification and language tags were omitted from the description. For a complete overview of RDF the reader is referred to [9]

in these languages. For example, one cannot describe the class of all objects that belong to the classes "person" and "actor" and that are in relation "has played in" with an object of the category "James Bond movies". More examples of such advanced statements are discussed with respect to their utility for Wikipedia in Section 3. In order to overcome these limitations in a way that still allows for effective machine processing, the *Web Ontology Language OWL* [12] has been created. OWL introduces further constructs for describing ontologies more specifically, e.g. inverse relationships, conjunctions of classes, or number restrictions on roles. OWL is based on Description Logics [2], which yields a sound theoretical background for implementing correct software and ensures semantic interoperability between the existing tools.

The purpose of this paper is to propose a practical solution for making the powerful specification languages mentioned above available to the average user. It is evident that the required mechanisms for describing content already exist, but that Wikipedia can only profit from such technologies if they are adjusted to the requirements of practical usage in a global community.

3 Design

We now present the details of our proposal, including in-depth discussions of its impact on Wikipedia's usability and performance. As mentioned in Section 1, hyperlinks between articles are a central feature of Wikipedia. The problem is that they do not lend themselves to automatic processing, which in turn is needed to implement features such as query answering. The reason for this shortcoming of hyperlinks is that their meaning is documented in natural language only – a representation that is still mostly incomprehensible for computer programs. For example, the article about the movie "On Her Majesty's Secret Service" contains links to the articles on "George Lazenby" and "Sean Connery". The important information which of these (if any) has been an actor in this movie is not available without understanding the whole text.

We propose to add typed links as a means of overcoming this deficiency in a simple and unintrusive way. The idea thus is to introduce a system of link types that are to hyperlinks what categories are to articles. For example, there could be a type "is main actor of" which is used for links between movies and their main actors. Such types can be generated by users, much in the same way that new categories are created for articles. Other suggestive examples of such types could be "has capital," "is mother of," or "is year of birth."[4]

Now one can specify a type for the hyperlinks that appear in an article, using a simple extension of Wikipedia's link syntax. For example, the article "On Her Majesty's Secret Service" could contain the sentence "In 1969, it was produced as [...] the first and only film starring [[George Lazenby||is main actor of]] as James Bond." Our choice of syntax here is not essential and one may prefer the

[4] Note that we adhere to the convention of naming such types starting with lowercase "is" or "has" so that their relational meaning is immediately clear and they are not confused with categories.

usage of characters other than || to separate link targets from types. All that one has to take care of is that the choice does not meddle with the article name (e.g. ":" is not available) and that it is still possible to unambiguously add the displayed name of an article (like in "[[George Lazenby|Lazenby||is main actor of]]").

As with categories and articles, giving types for links of course is not mandatory. In fact, there are many cases where there is no suitable type that characterizes the relationship specified by a link between two articles. In the above example, "Sean Connery" is mentioned since he has been the main actor in all other James Bond movies from this time, but he is not directly related to "On Her Majesty's Secret Service."

As mentioned above, link types are to be administrated like categories. This includes the possibility to specify a hierarchy of link types that defines whether some relationships are more general than others. For example "is actor in" is more general than "is main actor of": every main actor is also an actor. The possibility to define such relationships is crucial for the usage of link types. Without it, users looking for all actors either would not find any main actors or one would have to specify huge numbers of link types for every link.

Based on these simple mechanisms one can already make most of Wikipedia's hyperlinks accessible to computers. The resulting network of typed links can be used for further implementations, but it is also interesting in its own right. Even simply displaying part of it (e.g. all movies that are linked to "George Lazenby" via "is main actor of") would be interesting to users.

3.1 Usability aspects

The incredible success of Wikipedia is based on its simplicity: everybody can contribute without having to learn a complicated markup-language. The technical barriers for reading and even authoring are low. Any extension which has the potential to make Wikipedia complicated or even unusable to some people is unacceptable, whatever its possible merit is. For this reason, the proposed technology is designed to be simple and completely unintrusive: an author can completely ignore the availability of link types without any noticeable impact on her editing.

On the other hand, the definition and maintenance of the available link types is very similar to the current task of creating and editing Wikipedia categories. Experience shows that these tasks can easily be solved by the community, even if many people will not get involved in categorization or editing of category definitions at all. We believe that our current approach will indeed behave very similar to the category system in practical usage. Moreover, experience in ontology modelling shows that the number of required relationships is often rather small when compared to the number of useful categories – the reader may want to compare the obvious categories (genres) of movies with the number of useful links between movies and, say, persons (actors, directors, etc.). Indeed, the expressivity of typed links is in their ability to connect articles, possibly from different categories. Describing that a city is in Germany or that a lake is in Europe

can be done with the same type of link without introducing any inconvenience in usage.

Still it is possible to integrate typed links in an even more transparent fashion, by connecting current *templates* with link types. The template technology allows one to insert predefined parts of Wikipedia source code into articles, and to fill certain positions within these templates with customized code. An example of this practice is the "Infobox Movie" template[5], where one can specify standard information about movies (composed of properties like "Starring" and "Directed by"). It would be easy to associate fixed types to all (untyped) links provided for one particular property of a template. Doing so, one obtains a way to add link type information without even changing the source code of existing articles.

3.2 Implementation, performance and scalability

For a worldwide distributed system of Wikipedia's size, performance is almost as important as usability. The question is how the proposed features will impact performance and how they will scale in the context of a system that is subject to perpetual change. We will now discuss this question and propose a concrete schema for implementing our ideas.

As expounded above, the additional typing information will be composed of two parts: the general definition of the available types with their hierarchical organization and the concrete instances of these typed links between articles. Furthermore, all of this data changes continuously. Considering the fact that processing of a single query will usually require all the information about types and links, and possibly even about the whole category schema[6], one may wonder whether our proposal is realistic at all.

We argue that it is, because the linking information – although it deals with connections between articles – is completely local to the article it is specified in. In other words, it is possible to process the given link types each time that an article is edited and to attach this information to the article. Changes in other articles will not affect the validity of the extracted statements.

It is clear that the required processing of articles is a very simple task (compared to, say, link extraction in general), and that the present implementation could easily be adapted to handle this. The same is true for the processing of the type hierarchies, which are indeed very similar to the existing category hierarchies. Representing the extracted link data internally then is straightforward based on an additional customized database table.

However, to fully exploit the new features, we suggest to go further than this. For example, writing a system for query answering from scratch would be

[5] See http://en.wikipedia.org/wiki/Template:Infobox_Movie.

[6] For example, many geographical entities can be related via links of the type "is located in" (cities are located in countries, islands are located in oceans, etc.). The user may want to narrow down the search to obtain only the rivers and lakes that are located in Germany, which requires the evaluation both of typed links and of categories.

quite a difficult task. Luckily, systems that can work on the emerging structure of typed links (relations), categories (classes), and articles (instances) exist and have been subject to intense research in recent years. It is thus desirable to provide Wikipedia's structural information in an output format that is understood by available applications. OWL/RDF is such a format which (1) has been tailored to express the kind of semantic information described in this paper, and (2) is supported by many relevant applications due to its status as an official W3C standard.

Therefore Wikipedia could utilize numerous powerful applications (many of them freely available online) by just providing OWL/RDF output. Considering the possible visibility of such an effort, some research projects might even provide dedicated Wikipedia search engines on their own servers. OWL/RDF also enjoys the property of being specified in a completely modular XML syntax, where the order of statements is not relevant. Thus, one can simply combine the OWL/RDF fragment for each article in an arbitrary order to obtain a valid knowledge base that can be fed into existing tools. Similarly, it is straightforward to automatically generate OWL/RDF specifications that express the current categories with their hierarchy and the classified articles.

3.3 Additional features

The only new feature introduced so far are typed links between articles, which can be organized in a hierarchy. While this already offers plenty of new possibilities, it is only a part of the structural information that can be specified in OWL (and that can thus be supported by applications). We will now give a short account of other useful properties that also could be supported.

First of all, the constraint of having links only between articles may be too strict at some point. For example, when looking for all actors with "Sean" as a first name, one would not be able to rely on article-article links, since there are no articles for first names linked to from the actors articles. However, OWL supports datatypes and can express relations between instances (articles) and strings or integers. Likewise, it would be possible to find a convenient syntax for expressing such relations in Wikipedia's source code, e.g. by allowing users to select pieces of text to be interpreted in this way (similar to the hyperlink syntax, but without any effect on the compiled article). Another advantage is that datatypes come with additional background information. For example, if the size of a person is specified as an integer, it is possible to search for all persons that are smaller than this size. If the size is just an article name, the machine will usually not be able to interpret the "smaller than" relation appropriately. However, it still has to be considered whether such relationships are useful in Wikipedia.

Next, let us consider the following example: the geographical location of a city can be defined via a link of type "is located in" like e.g. Karlsruhe is located in Germany. On the other hand, Germany is located in Europe and, considering this, one might also want to infer that Karlsruhe is located in Europe. Using

OWL, it is possible to do this during query answering, without having to give additional "is located in" links in the article on Karlsruhe. One just has to specify that the property of being located in something is *transitive*, i.e. that if A is located in B and B is located in C, then A is located in C. This advanced feature clearly is of some importance for Wikipedia, since similar transitive relations like "is a relative of" or "is ingredient in" are common to many application domains. Another advantage of this is that most users will not have to decide whether a relation is transitive or not; they can just use the types as usual and possibly wonder how Wikipedia knows that Karlsruhe is in Europe even if this is written nowhere in the article. Transitivity of a particular link type can be switched on or off at any time and can of course be ignored by any application that does not support this feature of OWL (leading to some fewer results).

Another possible situation is that two types of relations are *inverse* to each other. For example, there might be link types "is capital of" (from countries to cities) and "has as capital" (from cities to countries). Clearly, these two types are just inverse, so specifying one is sufficient to know about the other. Again this can be expressed in OWL and it is easy to imagine an interface on the type page to edit the property.

In addition, OWL also supports the specification of axioms for describing the domain and range of each relation. For example, one may want to require that every article that has an outgoing link of type "is capital of" belongs to the category "Country." Such constraints should never disallow an editing step, but they may serve to detect problematic specifications at some later stage. In fact, due to the aforementioned locality principle that ensures good performance, domain and range axioms will not even be available when editing an article.

We remark that the classical interpretation of domain and range in OWL is to change the classification rather than to reject a specification. For example, if we declare Berlin as the capital of Germany and require that every capital must belong to the category "City", then one could infer that Berlin is a city. So someone who is looking for all cities might get Berlin even if this is not explicitly specified in the according article. However, whether this is inferred or whether domain and range is just ignored for querying is the choice of the concrete implementation.

Finally, OWL does also support statements that constraint the number of things that are allowed to be related via links of a given type. For instance, one may want to require that every country has exactly one capital specified. Again, the usage of such additional information is left to the software working on the exported OWL specification.

4 Implementation plan

In this section, we sketch the steps that are needed to put our approach into practice. The intention is to provide a concrete roadmap for introducing the desired new features gradually into the working system.

1. As a first step, one needs to represent a simple management system for link types. For a start, it is sufficient to adapt the current functionality of Wikipedia's category system. Note that, unlike categories, link types do usually not show up explicitly on an article page, so there has to be some way of accessing their pages. One feasible possibility for this is to create a simple search or listing function with the type management system.
2. Next, the definite syntax for typing links within articles needs to be fixed and implemented. Extracting the according information from source code should be fairly simple, but one still has to decide whether there should be an additional internal representation for link types, or whether the OWL export suffices. We remark that the link information is rather independent of the article. Especially, types are not associated with any particular occurrence of a link to another article, but describe the general relationship between the two articles. Thus duplicate connections of the same type can be eliminated. Furthermore, one again should include a way to access the pages where the link types that occur in an article have been defined. This can be done in a way that is similar to the solution for templates, where used templates are listed at the bottom of the edit page.
3. Up to this point, the public may not have taken much notice of the enhancements. Now in order to promote the new editing options in Wikipedia, it will be helpful to start an initiative for typing the links in some particular subdomain of Wikipedia. This task can best be solved in cooperation with a dedicated Wikiproject that has a good overview of some limited topic. The domain experts in this project can then develop a first hierarchy of link types and incorporate these types into part of the articles within this project. The generated OWL-output can then be used within offline tools to demonstrate the utility of the effort. Domains that already offer a rigidly used template, like countries, may be early adopters due to the efficiency gained by combining typed links and templates as described in Section 3.1.
4. It is to be expected that freely accessible online tools for querying the Wikipedia specification are not available at this stage. A first step to change this is to make the OWL output of Wikipedia public so that external projects can work on this data. Note that it is not necessary to update the specification continuously. As in the case of categories, link types are assumed to be rather stable and the envisaged querying functions do not require ultimate precision anyway.

   At the same time, one can consider cooperations with research facilities to provide online interfaces to their applications in exchange to being hyperlinked on Wikipedia's search pages.
5. In a similar fashion, one can start to provide (simple) internal services based on the new data. Such services can be based on existing tools if these are available. Alternatively, simple functions can also be implemented in an *ad hoc* fashion. This part is very important to demonstrate the value of typed links to the community, since one completely relies on the effort of the authors worldwide to include type information in the millions of articles Wikipedia currently consists of.

The issue with the previous two items is that they require a way for the user to pose queries. Doing this in an OWL-based query language or in SPARQL[7] is possible, but most users would not be able to specify their queries in this language. Another tempting possibility is to allow links within articles to point to query results instead of articles. For example, one can easily provide a link to the current list of all movies an actor ever participated in, without having to write (or update) this list. Many highly specialized lists and categories could be simplified in this way. The queries in this case are only given within the link, such that most users do not have to edit them.
Further ways to pose queries in a more user-friendly way include providing a simplified query language (e.g. a kind of *controlled English*) or to include a language processor for transforming natural language queries to formal ones. It is also possible to provide an interface for query construction that allows one to create queries by combining predefined expressions.

6. If the previous steps have been successful, there should be a growing stock of helpful applications available inside and outside Wikipedia. The primary development task then is to consider the implementation of additional features for defining link types.

From this implementation plan it is obvious that the basic functionality could be provided rather quickly. The crucial point for the overall success is to ensure the availability of useful applications by coordinating the work with research institutes that are active in the field. Given the fact that researchers are always looking for real-world scenarios to test and demonstrate their developments, and considering the amount of visibility that can be achieved by providing software for Wikipedia, it should not be too difficult to find partners for this endeavor. To this end, it also helps that the related topics of ontology research and semantic technologies currently receive a high interest in computer science.

5 Applications

The availability of machine readable descriptions of Wikipedia's content allows for a multitude of new applications. The development of such applications is greatly aided by the fact that the formal specification of Wikipedia's internal structure is provided separately and in a standardized format. Given that the specification is relatively stable (in contrast to the actual content of articles), it might suffice to update the ontology in regular intervals, e.g. on a weekly or monthly basis, without sacrificing functionality. This situation is very convenient for Wikipedia as well: for instance, it allows external applications or web services to answer rather difficult questions (like "Which actors ever had a leading role in a James Bond movie?") without having to contact Wikipedia or to download large parts of its content. Wikipedia is still available if the user is interested in article contents, while much of Wikipedia's bandwidth is saved.

[7] http://www.w3.org/TR/rdf-sparql-query

The major application to motivate our earlier investigations was searching and querying. However, as discussed in the previous section, posing queries is also a challenge for human-computer interaction. Much research is invested in these areas as well, and for ultimate usability for the end user, certain simplifications will have to be made. For instance, systems based on ORAKEL [5] can be used to offer a natural language query interface to the semantic structures inside the Wikipedia. Users can ask questions like "What are the ten biggest cities in Nigeria that don't lie at a river?" and the ORAKEL system translates this into queries to the underlying inference engine working on the Wikipedia data. The availability of machine readable information will greatly enhance the capabilities of such an approach, since the task is reduced to correct query generation (in contrast to systems like Internet search engines where query answering itself is also a challenge).

Alternatively, one can implement systems that assist the users in creating queries based on predefined schemas. To this end, it is helpful that the underlying semantics of OWL allows to combine queries with logical operators. Especially, one can refine queries by conjunctively adding additional conditions, such that even a simple query generation interface could be quite powerful as a searching and answering tool. The possibility of providing a simpler interface to create queries directly might also involve a simplified query language that can be translated to OWL-queries automatically.

As a third option for querying, we propose to "hard-wire" queries that have been prepared by knowledgeable users. OWL is not particularly difficult (compared to query languages like SQL) and many people will be able to familiarize themselves with it easily. Handwritten queries could then be placed in Wikipedia articles as links to the query result. These links can then be used to replace the continuously growing amount of specialized lists and categories that are currently applied to mimic such querying behavior. A single role like "is actor in" already can replace multiple manually edited article lists, e.g. the filmography of Sean Connery or the list of all James Bond actors. Query results could be cached or recomputed on every request, depending on how much computation time is available to the server. This use of prepared queries alone can solve many of Wikipedia's current structuring problems by helping to overcome redundant categorization. Categories like "People with asteroids named after them" or "Rivers in Northamptonshire" could readily be emulated by queries – categories could go back to just saying what *kind* of thing an article describes, and not how it *relates* to other articles.

Important as querying might be, there are many other applications that can be build on top of semantic information. For example, there are many ways to improve the editing process through computer assistance. As mentioned above, the specification of range and domain categories for typed links can be used to suggest categorization to contributors. For example, the editing system could warn that articles about someone who was an actor in a movie should also belong to the category of actors. Such warnings must never prevent the user from editing as he wishes (which would be incompatible with the Wiki idea),

but they can help editors to spot potential misconceptions when using link types and categories.

Another type of inconsistency detection becomes possible by comparing Wikipedias in different languages. Since articles, categories, and link types can be connected to their counterparts in other languages, software can automatically or semi-automatically check whether multiple Wikipedias agree in content. This can immediately be applied to suggest new articles or interwiki links that are still missing in one of the encyclopaedias. Thus one obtains the first potent possibility to directly compare the content of Wikipedias of multiple languages (though this is certainly not a fully automatic process, since there are many causes for not finding full correspondences here).

A further advantage is that ontological information, in contrast to full article texts, can also be collected and generated very easily in a fully automatic fashion. For example, robots can include semantic information in Wikipedia articles based on data that is freely provided by other websites, such as online libraries, shops, movie databases, or scientific repositories. Furthermore, given that the interlanguage links within Wikipedia are present, such information can readily be included in encyclopaedias of any language.

Due to the usage of a standard ontology format one can also make use of existing Semantic Web applications, e.g. for visualization of content: tools like Aduna[8] or the KAON OI Modeler [8] are able to visualize the relationships between articles, and even to offer a user interface based on graph visualization. Taking this even further, one can create new methods for graphical browsing of Wikipedia. Even complex graphical features that do not lend themselves to online implementation might still have the ability to enhance offline (CD/DVD) versions of Wikipedia.

In spite of the envisaged opportunities for improving Wikipedia, one has to be aware that not all of the existing implementations are instantly applicable to Wikipedia. OWL/RDF export basically would make Wikipedia the biggest ontology repository in the world, and it is obvious that this seriously challenges scalability and efficiency of the available software. Huge ontologies with tenth of thousands of classes are already in practical use in science and industry (e.g. for medical classification) and can be handled by existing software. But ontologies of this size are still rare and are often created for commercial purposes, such that software developers usually have no free access to realistic data for benchmarking and improvement.

While this situation is certainly a hindrance for the targeted exploitation of semantic technologies, Wikipedia itself is part of the solution: By creating a huge real-world ontology for the public domain, Wikipedia would position itself as a forerunner in Semantic Web technologies, allowing scientists to reap upon real world ontologies. Developers can create new tools or enhance existing ones with machine readable data fostered at Wikipedia. Web developers can query Wikipedia for certain pieces of knowledge and aggregate this in existing web

[8] `http://aduna.biz`

pages or web services, thereby creating dynamic pages that are enhanced with the knowledge offered by Wikipedia. In addition, a knowledge base of Wikipedia's size and scope can be used to enhance interoperability between semantically enabled applications, since it specifies a huge corpus of standard concept names, provides relationships among these, and describes connections to concept names in other languages. So, aside from the advantages Wikipedia would benefit from itself by implementing typed links, it could possibly become the single most important stepstone to the future of the Semantic Web.

6 Related approaches

The idea of combining the usability and openness of a Wiki with the machine-readable, extensible and standardized languages from the Semantic Web community is not new and has been discussed for years.[9] An approach close to our suggestions probably is the "Kendra Base" wiki[10].

The recently developed WikiSAR [1] integrates many of our ideas and shows the feasibility of combining semantic data entry with rich query facilities. Two mature semantic Wiki implementations that are well-known in the Semantic Web community are Platypus [11] and [13]. However, in contrast to our proposal, both separate Wiki page data from more complex semantic page metadata.

The idea of integrating machine readable data into Wikipedia has also been discussed earlier[11]. This problem also came into focus in the context of the introduction of categories, which in practice have a rather sloppy semantics, reaching from the classic instance-of-relation that we employ as a basis for our approach to more general types of relationships between topics. It is to be expected that the annotation with categories will have to be made somewhat more precise in order to allow for a formal evaluation of Wikipedia. The extraction of semantic (linking) information from the current corpus of Wikipedia is also relevant to our current approach, since it can generate candidates for link types automatically. First steps in this direction are taken in [7].

Another related effort is the semantic annotation of the Wiktionary projects, with the goal of allowing wide-reaching interoperability between languages and inclusion of external knowledge-bases like thesauri [10]. It will be important to ensure compatibility of the semantic languages chosen for this purpose with possible semantic extensions of Wikipedia, e.g. by employing the same ontology language as an underlying formalism.

As requested in "The Wiki Way" [6], we stay true to the free-text entry mode and thus avoid to create an in-browser ontology-editor.

[9] `http://www.c2.com/cgi/wiki?SemanticWikiWikiWeb`

[10] `http://www.kendra.org.uk/wiki/wiki.pl?KendraBase`, reviewed for WikiData at `http://meta.wikimedia.org/wiki/Kendra_evaluation`

[11] `http://de.wikipedia.org/wiki/Benutzer:Duesentrieb/Semantic_Wiki_Web` discusses the semantic content of categories and the possibility of annotating Wikipedia with RDF-like relations.

7 Summary and conclusion

In this article, we discussed the problem of disseminating substantial parts of Wikipedia's knowledge in a way that enables programs to query, browse or visualize Wikipedia's content. Recognizing that additional machine-readable information will be needed for this purpose, we proposed to introduce a new system of typed links to Wikipedia. Like categories are now used for classifying articles, new types would then be used for classifying links. The rationale behind this idea is that each hyperlink specifies a certain relationship between the linked articles, which is comprehensible to humans through the explanations within the article. Adding types to links makes this information available to machines as well, thus creating a huge and highly structured network of typed links between articles.

Discussing the details of syntax and implementation, we pointed out that the proposed changes will neither affect the current editing experience nor have a significant impact on performance. However, the full power of our approach is only harnessed by providing Wikipedia's added structural content in a standard specification format such as OWL/RDF. This dedication to open standards is achieved very easily but has tremendous impact on the utility value of the extracted information: one can now use numerous available applications that readily support such formats. Instead of laboriously implementing each desired functionality, Wikipedia could thus profit from the many years of research on and development of semantic technologies within the computer science community.

Furthermore, we argued that our approach carves the path to solve many existing problems. The insight that relations (links), classes (categories) and instances (articles) are the basic elements of machine readable domain specifications is well-known in the related research areas, which further substantiates our claim that these basic concepts are a feasible choice for our setting as well.

Finally, we also discussed in detail how to put our approach into practice and gave examples of concrete applications that will then become available immediately. It was pointed out that the main difficulty is not the implementation of the required editing functions (which is indeed rather straightforward), but the timely availability of powerful features or applications that reward the authors' efforts in adding the proposed information. Thus the full exploitation of the proposed semantic technologies will be aided by cooperations with research facilities and external developers. Again, it is the support of open standards that allows for such cooperations in the first place.

In summary, we suggested an approach of combining semantic technologies with the Wiki paradigm in a way that can be highly beneficial to both areas. We believe that semantic technologies can be implemented in a way that allows everybody to contribute to a collaborative knowledge base without major technical barriers. Wikipedia is in the position to make this important step towards the future of knowledge organization.

References

1. D. Aumüller. Semantic authoring and retrieval in a wiki (WikSAR). In *Demo Session at the ESWC 2005*, May 2005.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The description logic handbook: theory, implementations and applications*. Cambridge University Press, 2003.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001. Available at `http://www.sciam.com/2001/0501issue/0501berners-lee.html`.
4. D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. available at `http://www.w3.org/TR/rdf-schema/`.
5. P. Cimiano. Orakel: A natural language interface to an F-Logic knowledge base. In *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems*, LNCS. Springer, 2004.
6. W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, 2001.
7. D. Kinzler. Wikisense – mining the wiki. In *Proceedings of the 1st International Wikipedia Conference, Wikimania 2005*, 2005.
8. A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies in the semantic web. *VLDB Journal*, 12(4):286–302, 2003.
9. F. Manola and E. Miller. Resource Description Framework (RDF) primer. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/rdf-primer/`.
10. G. Meijssen. The excitement of Wiktionary. In *Proceedings of the 1st International Wikipedia Conference, Wikimania 2005*, 2005.
11. S. E. Roberto Tazzoli, Paolo Castagna. Towards a semantic wiki wiki web. In *Demo Session at ISWC2004*.
12. M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Recommendation 10 February 2004, available at `http://www.w3.org/TR/owl-guide/`.
13. A. Souzis. Rhizome position paper. In *1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, September 2004.