

# Graphingwiki - a Semantic Wiki extension for visualising and inferring protocol dependency

Juhani Eronen and Juha Röning

Oulu University Secure Programming Group  
Computer Engineering Laboratory, Linnanmaa BOX 4500  
FIN-90014 University of Oulu, Finland  
ouspg@ee.oulu.fi

**Abstract.** This paper introduces the Graphingwiki extension to MoinMoin Wiki. Graphingwiki enables the deepened analysis of the Wiki data by augmenting it with semantic data in a simple, practical and easy-to-use manner. Visualisation tools are used to clarify the resulting body of knowledge so that only the data essential for an usage scenario is displayed. Logic inference rules can be applied to the data to perform automated reasoning based on the data. Perceiving dependencies among network protocols presents an example use case of the framework. The use case was applied in practice in mapping effects of software vulnerabilities on critical infrastructures.

Keywords: semantic wiki, protocol dependency, visualisation, inference

## 1 Introduction

In recent years, Wikis and the semantic web have become the state of the art methods for the management of information. Wikis have proven to be an effective means for the collective gathering and editing of bodies of data ranging from encyclopaedia to bug tracking and journals. Semantic web is envisioned as a universal medium for data exchange and as a tool to manage the interconnection of information, enabling automated analysis of data. [5] [18] [24] [1]

Both of the technologies have strong selling points: Wikis enable collaborative, open, evolutionary, and easy modification of data, and the semantic web employs Resource Description Framework (RDF), a powerful yet relatively simple language for representing information about World Wide Web (WWW) resources. RDF consists of subject-predicate-object triples that are used to make statements about resources [12]. An RDF resource can basically be anything that has a Uniform Resource Identifier (URI), so it can be used to refer to any web resource. The triples describe either relationships between two resources, the subject and the object, or an aspect of the subject, the value of which is specified by the object. The predicate is a resource that the relationship or aspect describes. Integrating wikis with RDF could bestow it with the editing abilities necessary for efficient knowledge management.

Combining the approaches and techniques of Wikis and semantic web has met little success. The little support traditional Wikis offer for semantic data usually culminates

in page categories and different kinds of comment tags. Semantic web tools are often single-user oriented and their operation frequently requires expert skills, which makes knowledge engineering challenging for domain experts. [7] [25] [18]

Wikis have the strength that they focus on the structure of the data instead of its presentation. Wiki users are accustomed to creating, linking and tagging content, which represent the bare minimum requirements for taking advantage of semantics. Adding semantic features to Wikis offers a smooth transition for exploiting different layers of knowledge. [23]

This paper introduces Graphingwiki, a Wiki extension that aims to enable knowledge engineering in Wikis by sidestepping the complexity of semantic technologies. The bare minimum functionality for semantic capabilities in a Wiki includes the implementation of a small but functional subset of RDF. This also follows the Wiki way of doing the simplest thing that could possibly work [6]. Users introduce semantic data into the Wiki by simply tagging pages and page links with words or phrases that sound suitable to them. RDF resources are represented on a Wiki page as tagged links and tagged page data. Together the page tags and the link tags create the RDF statements of the forms <page> <tag> <linked page>, <page> <tag> <URI resource> and <page> <tag> <tag value>.

The tags represent a flat namespace and do not have a hierarchy of any kind. In a way, this method of adding semantic data resembles folksonomies such as del.icio.us<sup>1</sup>. Tagging is simple and unrestrained as it aims for easy diffusion in the user base. Existing mechanisms, such as different kinds of linking, category pages and macros, are utilised as much as possible. Users may freely select the tags they use, which thus sacrifices consistency for practicality. This approach can prove more useful than forcing any predefined tagging schema [17] [22].

A Wiki functions as its own ontology, formed by all the tags in the Wiki's pages [2]. Each descriptive tag is assigned a page of its own so that terms can be defined and refined in the Wiki itself. The resulting ontologies are expressive to humans but lack the complexity and formality required for elaborate machine-processable constraints on the page data. This does not present a hindrance for knowledge management — in fact, the most successful knowledge models tend to be very simple and specific [18].

Interactive visualisation is proposed as a method for understanding the relations of information on the Wiki pages. Visualisations can be used to navigate the Wiki, and they include facilities for filtering out non-relevant data. This enables the quick derivation of a general view on any desired topic or entity.

Furthermore, Graphingwiki includes some logic reasoning capabilities for refining specific knowledge from the Wiki tags. Wiki pages can include rules that lead to new conclusions about specific tags, and the resulting data can be queried for sets of pages and tags that fulfil the premises of the query. This presents a fine-grained method for discovering relations amongst the wealth of data.

The paper presents the methods in the context of a practical use case, fathoming interdependencies in communication protocols. It is also argued that a similar knowledge management approach would also be effective for other domain-specific tasks where an

---

<sup>1</sup> <http://del.icio.us/>

universal topical scope and some of the other stumbling blocks of semantic technologies are not an issue [18] [21].

The paper is structured as follows. In section 2, additions to traditional Wiki features are presented, which beget methods for gathering, visualising and reasoning on the data for the example case. Section 3 presents some results of an initial analysis of the use of Graphingwiki. Directions for future work are laid out in section 4. Finally, the work is summarised in section 5.

## 2 Methodology

The main methods used in Graphingwiki include additions to the MoinMoin<sup>2</sup> Wiki markup and plug-in tools that save the semantic data for later processing, visualise the semantic data and make logical reasoning based on it.

### 2.1 Implementation issues

MoinMoin was selected as the starting point for semantic Wiki development based on the criteria that it is open source, implemented in the Python<sup>3</sup> programming language, is mature and extendable, and uses a file database. Graphingwiki is implemented as a set of plugin *actions* to manipulate the page data, *macros*, and *formatters* to render the semantic data to the desired viewable or processable forms. The design strives to maximise backwards-compatibility and the use of existing MoinMoin features.

The semantic data in each Wiki page is stored into a file of its own, in a symmetrical manner with the page data storage in the MoinMoin Wiki. A general-purpose graph library was created for this purpose. Semantic data is interpreted with the help of existing and augmented Wiki markup, and serialised in the defined graph format. As the markup allows for incoming links links that are not shown on the wiki page itself, a global file database of page linkage was also implemented.

Graphingwiki uses the Python bindings of the Graphviz<sup>4</sup> suite of layout tools to visualise the semantic relations of a Wiki page as graphs. The inference module is a simple unifier-based design in the style of many Prolog implementations.

### 2.2 Wiki markup additions

The chosen markup additions resemble closely those utilised by the semantic Wikipedia-project [24] and semantic Mediawiki [14]. Similar semantic additions developed for MoinMoin<sup>5</sup> were investigated but deemed to include only a portion of the desired features.

The goal of the markup additions is not to implement the whole of the RDF notation, but to present the user a simple and intuitive way to make statements about a Wiki page.

---

<sup>2</sup> <http://www.moinmoin.wikiwikiweb.de>

<sup>3</sup> <http://www.python.org>

<sup>4</sup> <http://www.graphviz.org>

<sup>5</sup> <http://theendmusic.org/programming/MetaDataPlugin>

Statements can only describe the containing Wiki page in relation to page tag values, Wiki pages and URI resources. Semantic data is marked up within page content and rendered in a meaningful manner when the page is viewed.

There are two kinds of statements users can make about a Wiki page: MetaData statements and augmented link statements. MetaData statements are used to realise semantic page tags. They are implemented with a macro and therefore follow the MoinMoin macro syntax of the form `[[MacroName(arguments)]]`. The arguments of the MetaData macro consist of tag-value pairs with an optional third argument that omits the macro from page rendering. For example, the statement

```
[[MetaData(SpecialPower, x-ray vision)]]
```

on a superhero Wiki page denotes that he or she has the extraordinary ability to conduct airport security checks without external hardware, among other things.

Respectively, augmented link statements are used to implement semantic link tags. They extend the MoinMoin named link syntax forms

```
[ :OtherPage:Wiki page] and  
[http://example.com URI resource]
```

that create links with descriptive labels (see Figure 1). Augmented link syntax adds a link tag to this markup, resulting in links of the forms

Wiki markup	Rendering
<code>[ :OtherPage:wiki page]</code>	<a href="#">wiki page</a>
<code>[http://example.com URI resource]</code>	<a href="#">URI resource</a>

**Fig. 1.** Rendering of normal MoinMoin links.

```
[ :OtherPage:linktag: page] and  
[http://example.com linktag: URI resource].
```

The special keyword “From” in the end of the type string denotes that the link is an incoming link, i.e. the referenced page links to the current page instead of the current page linking to it. For example, the statements

```
[ :OtherPage:linktagFrom: page] and  
[http://example.com linktagFrom: URI resource]
```

indicate that the current page is referenced by the Wiki page or the WWW page, respectively.

The statement `[ :DrX:Nemesis: DrX]` on the superhero Wiki page tells that the nemesis of our hero is Dr. X, described in the same Wiki. Respectively,

```
[http://example.com FanClub: http://example.com]
```

states that the hero’s fan club has its web page at the URI `http://example.com`. Repeating the link in the descriptive string is not required, the examples do so for reasons of clarity only. Figure 2 illustrates the rendering of these statements.

The notation defaults to the namespace designated by the Wiki. To avoid collisions with regular Wiki pages, the pages describing the page tags and the link tags are pre-

Wiki Markup	Rendering
<code>[[Metadata(SpecialPower, x-ray vision)]]</code>	<b>SpecialPower</b>
<code>[:DrX:Nemesis: DrX]</code>	x-ray vision
<code>[http://example.com FanClub: http://example.com]</code>	<a href="#">Nemesis: DrX</a>
	<a href="#">FanClub: http://example.com</a>

**Fig. 2.** Rendering of semantic statements.

fixed with 'Property'. Thus, in the examples of the previous paragraphs, 'PropertySpecialPower' and 'PropertyFanClub' are pages in the same Wiki.

By editing the descriptions and semantic data on the Wiki pages describing the page tags and the link tags, the community creates a contract on the formal meaning of a domain - effectively an ontology. This lets the users freely edit the ontology in a very Wiki-like fashion, which reduces the entry barrier and encourages vocabulary growth and expressiveness. For example, users of the superhero Wiki can elaborate on the concept of special powers (i.e. the content of the 'PropertySpecialPower' page), adding further information, declaring exceptions, and so forth. The availability of discussions on the subject, along with relevant links and multimedia, will help in understanding the concept. [11]

Graphingwiki is not planned to support any deeper semantic meaning to ontology entries. RDF schema or datatypes are not supported, nor are pages checked for consistency with any formalism. However, template pages can be used to create implicit meta-ontologies similarly as in Wikitology [8]. For example, a 'SuperheroTemplate' could include statements common for all superheroes, so that when a page for a superhero is created using that template, the author is reminded about the kinds of semantic data that should probably be included.

The semantic markup supports namespaced statements. The list of valid namespaces is gathered from the Wiki's *InterWiki* list. For example, the statement

```
[[Metadata(Wardrobe:JumpSuit, Spandex)]]
```

tells us that the hero in question wears a flashy spandex jump suit, and that the specifics on the style of dress can be found in the Wardrobe Wiki. Respectively, the statement

```
[wiki:WikiTwo/PageTwo OtherWiki:SeeFrom: wiki:WikiTwo/PageTwo]
```

represents the situation where the page 'PageTwo' of the Wiki 'WikiTwo' has a relation with the referencing page defined by the page 'PropertySee' in the Wiki 'OtherWiki'. Naturally, by adding the line

```
dc http://purl.org/dc/elements/1.1/
```

to the *InterWiki* list of the Wiki in question enables the user to employ Dublin Core<sup>6</sup> definitions in the Wiki pages. Although *InterWiki* lists are currently not user-editable in MoinMoin, the *InterWiki* list provides a relatively clean and straightforward way to add new scope to Wiki editing. Graphingwiki uses the namespaces merely as URI prefixes to the resource names, the RDF data corresponding to the resource is not fetched. Still, the namespaced URIs offer some advantages, as users can use standardised seman-

<sup>6</sup> <http://dublincore.org/>

tic tags with well-defined meanings, some primitive inference rules involving different namespaces can be used, and external RDF tools can utilise the full scope of the external semantic data. The semantic data in the Wiki data can also be dumped from the Wiki in N3 [3] notation for further analysis with external RDF tools.

### 2.3 Visualisation

Visualisations are composed of the node of the current Wiki page, the links leading to the page and from the page, and the nodes depicting the linked pages. Alternatively, all pages belonging to a category of the current page can be used as the root nodes of the graph, instead of merely the current page node. Visualising a category shows a whole field at one glance, including the direct and indirect relations of all the members, along with their immediate surroundings.

Page tags can be used to colour the nodes of the graph, and pages can be filtered based on their tags. Respectively, augmented links are coloured with respect to their link tags, by which they can also be filtered. Filtering can greatly reduce the clutter in the visualisation, and helps in concentrating to desired aspects of the data. Graphs can also be ordered with respect to one of the page tags. The tag values are lexically sorted, determining the rank of the nodes corresponding to the pages. Colouring and ordering the nodes offers two dimensions by which to organise the semantic data.

As an example, Figure 3 depicts a visualisation made by Graphingwiki with data automatically extracted from the WiFiPedia<sup>7</sup> wireless standard resource.

### 2.4 Inference

While visualisation makes semantics comprehensible, inference makes it operational. Generally speaking, inference is used to extend the set of known facts with the help of rules that concern them, and to find the facts, if any, that prove a stated goal. Inference engines that take the first approach are called forward chaining, as they start from valid data, while backwards chaining starts from the goal to be proved, and apply known facts and rules to produce a proof. [16]

A backwards-chaining inference engine is used to answer queries on semantic data. The engine uses Horn clause logic, i.e. clauses that do not have more than one positive literal, also used by many logic programming approaches such as Prolog. Horn clauses have desirable properties in that their satisfiability is solvable in polynomial time with algorithms linear to formula size. As the semantic data can be expressed in the terms of RDF triples, which are basically simple relations, it is straightforward to map them as clauses.

The inference rules and queries are stored as Wiki pages for easy editing and reference. The rules are expressed in the N3 notation, as Graphingwiki markup extensions do not include any way to express them. The result of the query is a set of RDF triples, also in N3 format, that maintain the conditions presented by the query. For example, according to the old adage “the enemy of my enemy is my friend” an evil mastermind, Dr. X, might want to query the superhero Wiki for enemies of his enemies to find new

---

<sup>7</sup> <http://www.wifipedia.org/>



allies to battle his nemesis, Goody Two Shoes. The rule, representing Dr. X's notion on enemies and allies, and the query would be as follows:

```
{?x Enemy ?y. ?z Enemy ?x} => {?z Ally ?y} .  
{?who Ally DrX} => [] .
```

The query could result in the following reply:

```
CookieMonster Ally DrX .  
DrEvil Ally DrX .  
PowderedToastMan Ally DrX .  
GoodyTwoShoes Ally DrX .
```

Having disproved the old adage, Dr. X curses his wretched query in frustration.

### 3 Practical usage scenario

Graphingwiki has been used for the purpose of discerning and visualising interdependencies of protocols. Data is gathered from technical specifications and from experts of different protocol environments. The accumulated data is then visualised, bringing up different aspects from the data related to protocol dependency and security. The resulting views can additionally be used as a communication method between researchers and other actors. Inference is used as a method of gaining deeper insight to dependency chains and networks.

#### 3.1 Protocol dependency

Protocols can be thought of as languages shared by the information systems for communication. Most current information systems implement a large number of protocols, most of which it requires for normal functionality. In effect, the system can be communicated with by a number of means, and it parses diverse network data. This makes the system, as the other systems on the network, dependent on the implemented protocols in a multitude of ways. Assessing the dependency of protocols and the predominance of the protocols in the network is critical in the view of its robustness.

The issue is further complicated by the fact that protocols within a single protocol family or even between protocol families often have connections. Thus, the impact area of vulnerabilities in a shared component is greatly expanded due to protocol dependency. This may lead to faults that can have a significant effect on an infrastructure. [9]

#### 3.2 Extraction and augmentation of data

Initially, protocol data is gathered from standardisation organisations and from indices collecting data on standards. Examples of semantic data in standards include status, types of relations with other standards, the protocols involved and so forth. The data is gathered with scripted methods and inserted into corresponding Wiki pages with

similar means. Most of the structured data in the standard texts is imported, following the approach of aggressive population of semantic and ontological data from existing databases [4]. This results in the quick generation of a relatively rich body of data as a starting point for a comprehensive protocol Wiki. Also other semistructured data on standards can be inserted.

While the process of adding given semistructured data cannot be effectively automated for all cases, the extraction approach is a pragmatic one, making the best use of the data available. Although the different data sources may adhere to any number of conflicting explicit or implicit ontologies, a lightweight approach to ontology gives the leverage to process the resulting primordial soup. This represents a bootstrapping process for semantic Wikis, as the benefits of semantic data are illustrated only by the availability of such data. These benefits far outweigh the costs of generating the semantic data along with the data. Similar approaches to data extraction have been applied successfully [20] [19].

After the initial data gathering phase, the data is inserted into Graphingwiki. The details of this process are somewhat subject-dependent, but follow the same basic principles. Whenever new concepts are introduced in the data, new Wiki pages are created to describe them, and data concerning a protocol or other concept already in the Wiki is simply updated to that page.

As much of this data as possible is inserted to the pages in the forms of the attributes of the concept and its relations to other concepts, as these forms of data are machine-processable. Page templates can be used to help formalise the extended markup [24]. On the other hand, custom semantic tags for specific situations or scenario can be used. Explanations, quotes, and WWW resources can be written on the page as is.

In the collaboration phase, the experts are invited to join in to view and augment the results gathered in the Wiki from their interviews and additional sources. Experience has indicated that it may help in this phase if the data gathering phase has not been exceedingly careful in filtering contradictory or controversial arguments about the protocols. This is due to the fact that experts are often more keen to remove such flaws from existing data than to add complementary data to an empty page.

During these phases the data body is developed from a fairly generic and dry viewpoint towards exceedingly rich and specific use cases. Users immediately benefit from the practical domain experience included in the Wiki.

### **3.3 Visualisation and Reasoning**

The ability to make logic deductions on the expert-supplied data can unearth results not easily discovered by traditional means. As an example from the Wiki context, Decker et al. uses reasoning to enable reuse of software engineering knowledge [8]. The approach taken in the development of Graphingwiki with respect to reasoning techniques is straightforward and pragmatic, so that the inclusion of logic is based on approaches that are known to work and are required. The focus lies heavily on immediate benefits of reasoning, the inclusion of higher-order structures is deferred until they are explicitly needed [4].

As an example case of inference on the domain of protocol dependency, the true cause of a network error related to two hosts containing a plenitude of services can be

inferred from a data body on protocols and related implementations. Similarly, the gross effect of a single vulnerability for a network can be assessed, optionally involving even chains of vulnerabilities and exploits. Similar approaches have emerged in the context of security research, particularly in network vulnerability assessment (e.g. [15]), but also in inspecting the configurations of single workstations (e.g. [10]).

### **3.4 Limitations**

The population of a Wiki with data from semistructured sources is a useful facility, but it may not be applicable to a portion of available material due to technical or licensing issues. In some cases, the data abstraction features may suffer from some constraints. Visualisation techniques are naturally limited to a certain volume of data that they can relay in an efficient manner.

Reasoning also has its limitations that have hindered its use in many cases. Main problem is the state space explosion resulting from massive knowledge bases. This can be countered by using monotonic logic and highly domain-specific data sets, although limits on query tree depth and traversal time can also be of help. All the statements made with Graphingwiki are essentially monotonic, as they only bring more data to the knowledge base without contradicting earlier statements. This is due to the inherent lack of meaning of the statements in the Wiki, as the different aspects and relations are only given meaning by humans interpreting them, or by the inference rules and queries.

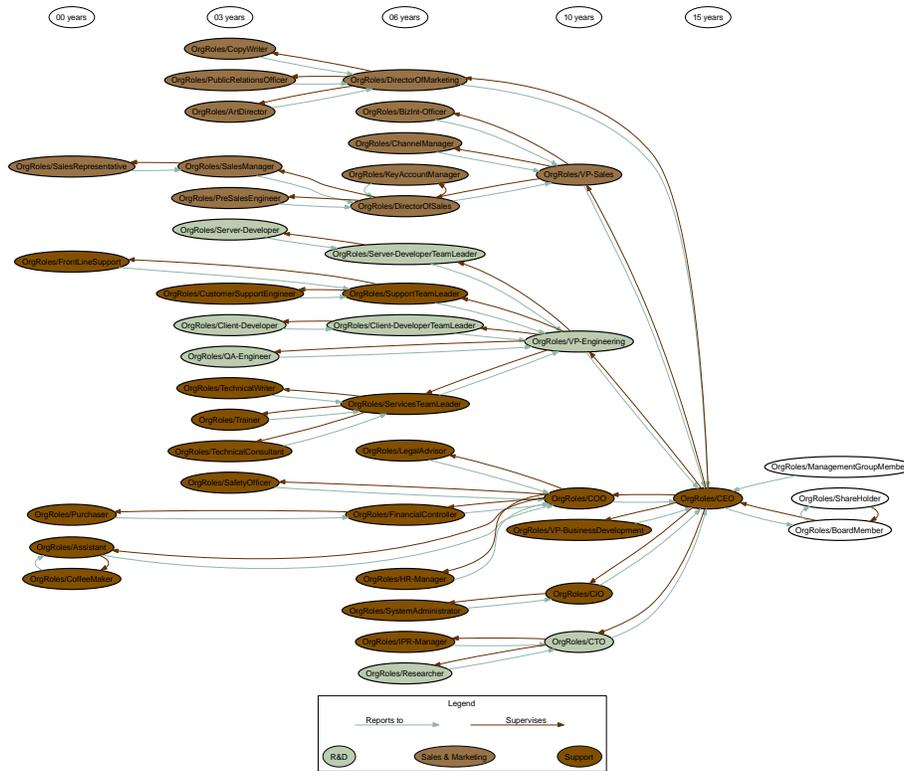
While the statements are limited in their effect, there are no similar restrictions to the inference rules queries. Thus, great care must be taken when generating them, as they might bring contradiction or belief revision into the system. The heterogeneity of the data gathered from various sources can present limitations to reasoning. As there are no guarantees on given semantic data being present on all concerned pages, the inference rules may not match all relevant data [19].

## **4 Discussion**

Semantic Wikis are a natural placeholder for various kinds of domain-specific data that are produced in normal course of work, enabling collaboration and groupwork. The gathering and visualisation of information was found straightforward with the methods explained in this paper. Visualising the relations of protocols has proved to be an effective method for realising the scope of a protocol in application and network contexts. The visualisations have been used in various stages of protocol-related vulnerability work.

It has been claimed that semantic tools also have applications in learning by evaluating, manipulating, and presenting data in new ways [20]. Visualising this data according to the requirements of a given domain presents an effective method for making its contents easier to grasp by humans. Consequently it is no surprise that in addition to fulfilling its intended purpose for creating protocol visualisations, Graphingwiki has proved to be useful for a variety of other tasks. New application areas emerged at a constant rate during its development, indicating that there is a great need for lightweight information visualisation facilities. Some of these areas are illustrated by the examples in the following paragraphs.

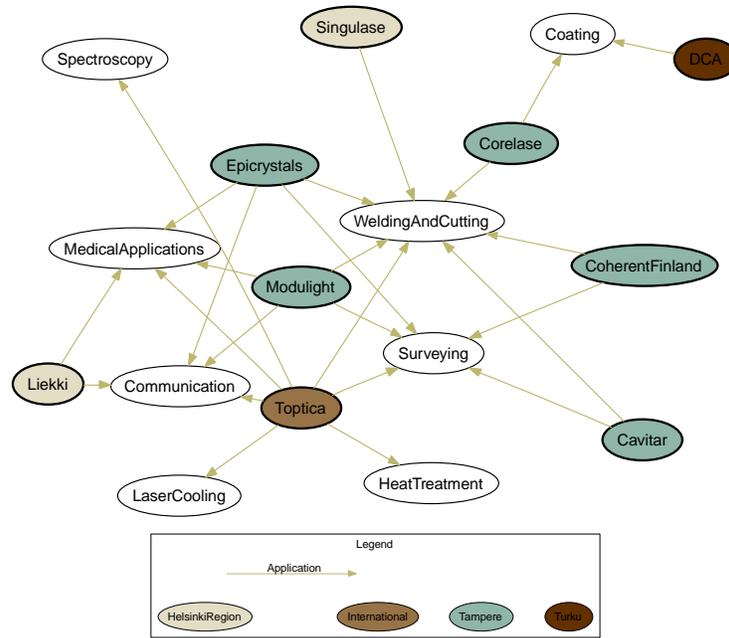
Figure 4 is an organisational chart of a company that has been created with Graphingwiki. The nodes of the graph represent the roles of different employees while edges report the reporting and management chains between the roles. The roles are ordered by their required experience and colored according to the departments they belong to. Similarly, Wiki pages containing data on employee responsibilities and fields of know-how could enable efficient resource management and aid in problem resolution. Social network mapping techniques could be used on this data even further, for example to identify communities and communication bottlenecks.



**Fig. 4.** An organisational chart created with Graphingwiki.

Figure 5 represents a survey on the research on laser technologies and on the manufacturers of laser products. Data on different actors of the field was inserted to a Wiki, along with their relations. This view on the Wiki data depicts the Finnish laser product vendors by location, with links to the application areas of their products.

Graphingwiki could be enhanced in a variety of ways to increase its efficiency and expressiveness, and to make it more approachable for users. A full support for different levels of ontology formalisation would be an obvious benefit, along with mechanisms



**Fig. 5.** Different laser applications, their producers, and locations.

that check the page's adherence to a specified ontology [18][8]. RDF schema to manipulate typed data could be added, as well as some OWL features. Many of the implicit Wiki relations, such as being part of a certain category or being made with a specific template, could be formed explicitly with these facilities. Importing RDF data related to instances of other namespaces would also increase the application scope of Graphingwiki.

The creation of ontologies might be easier and more scalable if users could first use the augmented link syntax to denote all statements, shifting to use the MetaData-macro only when it has been ascertained that the values of the link tags do not have further structure and can be considered to be mere tag value data.

Some of the semantic data in a Wiki could also be automatically generated from the knowledge of who created and modified the page, creation date, data on referring page given by the browser, and so on [8]. Similarly, page categories could be automatically suggested to the user by comparing the page with representatives from existing categories using Bayesian classification.

The inference engine in the extension will yet require some work to be fully operational in a practical manner. A major part of this work includes creating the basic queries representing the common use cases of the inference engine. It also includes queries with additional functionality such as "find all of the links from the Wiki that point to non-existing pages". Visualising the results of the queries would increase their understandability in the case of complex queries.

Users could be greatly aided by the creation of semantic data macros specific to their domains of knowledge. Further, the user interface could include tag word suggestions to help converge the tagging scheme, similarly as in the del.icio.us service and the Makna semantic Wiki. Another aid for the tagging scheme would be the use of synonym-declaring relations. However, experiences from Wikipedia suggest that problems regarding the selection of tags are not critical, and that the situation is further ameliorated by the Wiki pages describing the tags [24].

The visualisation style and the GUI would benefit from user interaction studies and research on other visualisation styles. Different dimensional views such as Zzstructures and Polyarchies could be used to produce more data-compact views [13]. Wiki pages could include navigation section of related links created with the help of faceted classification [23] [1], providing another alternative to the traditional wiki category scheme.

Many common use cases of Wikis, such as systems documentation and contracts, can encompass a smorgasbord of pages while placing great demands for the trustworthiness of the included data. As Wiki pages are by nature under constant revision and refinement, these use cases require facilities for specifying the page versions that constitute the de facto state of the entity. Visualisations that are bound to specific page revisions could be used to facilitate the version control of such entities while making their structure easier to apprehend.

Encapsulating the revision state in visualisations is a similar concept as the transition of software version control from the per file Revision Control System (RCS) into the set oriented Concurrent Versions System (CVS). Whereas in software development the module hierarchy facilitates easy revision tagging, in non-hierarchical Wikis the bound visualisations can provide for one click capture of a snapshot of a larger concept. Following the evolution of these visualisations could give insight into the development of the entity, and the processes involved.

## 5 Conclusions

This paper has shown how the MoinMoin Wiki can be extended to include some semantic capabilities. Graphingwiki uses the MoinMoin plugin mechanism along with its existing capabilities of linking and category pages to create a simple and lightweight semantic tagging scheme. The tagging scheme was further used to provide for the visualisation of semantic data and making reasoning upon it.

Graphingwiki has been used for knowledge engineering in the domain of network protocols. The visualisations have proven to be an effective aid in discovering dependencies between protocols, while the reasoning capabilities showed promise for uncovering complex relationships in the semantic data. The visualisations have been used in various stages of protocol-related vulnerability work.

Future research on Graphingwiki include analyses on the visualisation style and user interaction methods in the tool. This research could result in more compact and easily manageable views. Another future direction is the inclusion of more sophisticated semantic features, the lack of which currently limits the use of Graphingwiki with other semantic tools and data sources.

A great demand was noted for the management and visualisation of data from diverse domains. Usage of the tool was then attempted in a number of application areas. Initial experiences on the applicability of Graphinwiki for purposes outside its intended domain of application were very encouraging.

Therefore, a similar approach to handling, visualising, and inferring on data would probably be of much use in many other domains, including enterprise resource management and social network mapping. Organisational human resources related skill and social network mapping and documenting information systems from deployment level to strategy view with dimensions on security policy and system interdependencies are examples of envisioned use cases.

## References

1. Aumueller, D. SHAWN: Structure Helps a Wiki Navigate. In Proceedings of the BTW-Workshop "WebDB Meets IR", Karlsruhe, Germany, March 1, 2005. URL: <http://dbs.uni-leipzig.de/~david/2005/aumueller05shawn.pdf>
2. Aumüller, D., Auer, S. Towards a Semantic Wiki Experience - Desktop Integration and Interactivity in WikSAR. In proceedings of the 1st Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure, Galway, Ireland, November 6, 2005. URL: [http://www.semanticdesktop.org/SemanticDesktopWS2005/final/22\\_aumueller\\_semanticwikiexperience\\_final.pdf](http://www.semanticdesktop.org/SemanticDesktopWS2005/final/22_aumueller_semanticwikiexperience_final.pdf)
3. Berners-Lee, T. An RDF language for the Semantic Web - Notation 3 (1998). URL: <http://www.w3.org/DesignIssues/Notation3.html>
4. Berners-Lee, T. WWW2004 Keynote. Keynote speech in the 13th World Wide Web Conference, New York City, US, May 17-22, 2004. Slides available at: <http://www.w3.org/2004/Talks/0519-tbl-keynote/>
5. Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. Scientific American 284, May 2001. URL: <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
6. Cunningham, W. et. al. Do The Simplest Thing That Could Possibly Work. URL: <http://c2.com/cgi/wiki?DoTheSimplestThingThatCouldPossiblyWork>
7. Cunningham, W. et. al. Wiki Design Principles. URL: <http://c2.com/cgi/wiki?WikiDesignPrinciples>
8. Decker, B. et. al. Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. Workshop on Semantic Web Enabled Software Engineering (SWESE), at the 4th International Semantic Web Conference (ISWC 2005), Galway, Ireland, November 6, 2005. URL: [http://www.mel.nist.gov/msid/conferences/SWESE/repository/11self-org\\_reuse\\_of\\_se.pdf](http://www.mel.nist.gov/msid/conferences/SWESE/repository/11self-org_reuse_of_se.pdf)
9. Eronen, J., Laakso, M. A Case for Protocol Dependency. In proceedings of the First IEEE International Workshop on Critical Infrastructure Protection, Darmstadt, Germany, November 3-4, 2005. URL: <http://www.ee.oulu.fi/research/ouspg/protos/sota/matine/IWCIP2005-dependency/index.html>
10. Govindavajhala, S., Appel, A.W. Windows access control demystified (2006). URL: <http://www.cs.princeton.edu/~sudhakar/papers/winval.pdf>
11. Hepp, M., Bachlechner, D., Siorpaes, K. OntoWiki: Community-driven Ontology Engineering and Ontology Usage based on Wikis. In proceedings of the 2005 International Symposium on Wikis, San Diego, US, Oct 16-18, 2005. URL: <http://www.heppnetz.de/files/ontowikiDemo-short-camera-ready.pdf>

12. Manola, F., Miller, E. Resource Description Framework (RDF) primer. W3C Recommendation (2004). URL: <http://www.w3.org/TR/rdf-primer/>
13. McGuffin, M.J., Schraefel, M.C. A Comparison of Hyperstructures: Zstructures, mSpaces, and Polyarchies. In Proceedings of ACM Conference on Hypertext and Hypermedia, Santa Cruz, USA, August 9-13, 2004. URL: <http://eprints.ecs.soton.ac.uk/9230/>
14. Muljadi, H. et. al. Semantic MediaWiki: a user-oriented system for integrated content and metadata management system. In Proceedings of the IADIS International Conference WWW/Internet 2005, Lisbon, Spain, Oct 19-22, 2005. URL: <http://www-kasm.nii.ac.jp/papers/takeda/05/hendry05icwi.pdf>
15. Noel, S. et. al. Efficient Minimum-Cost Network Hardening Via Exploit Dependency Graphs. In proceedings of the 19th Annual Computer Security Applications Conference (ACSAC '03), Las Vegas, US, December 8-12, 2003. URL: <http://www.acsa-admin.org/2003/papers/98.pdf>
16. Norvig, P. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. Morgan Kaufmann Publishers (1992)
17. Rocha, L.M., Bollen, J. Biologically motivated distributed designs for adaptive knowledge management. In: Segel, L., Cohen, I. (eds) Design Principles for the Immune System and other Distributed Autonomous Systems. Oxford University Press, Oxford, UK (2001) 305-334
18. Schaffert, S., Gruber, A., Westenthaler, R. A Semantic WIKI for Collaborative Knowledge Formation. In proceedings of SEMANTICS 2005 Conference, Vienna, Austria, November 23-25, 2005. URL: [http://www.salzburgresearch.at/research/gfx/SemWikiForCollKnowForm\\_20060120.pdf](http://www.salzburgresearch.at/research/gfx/SemWikiForCollKnowForm_20060120.pdf)
19. Schraefel, M.C. et. al. The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex. In Proceedings of the 16th ACM Conference on Hypertext and Hypermedia, Salzburg, Austria, September 6-9, 2005. URL: <http://eprints.ecs.soton.ac.uk/10710/>
20. Shadbolt, N. et. al. CS AKTive Space or how we learned to stop worrying and love the Semantic Web (2004). URL: <http://eprints.ecs.soton.ac.uk/8817/>
21. Shirky, C. The Semantic Web, Syllogism, and Worldview. Clay Shirky's Writings About the Internet (2003). URL: [http://www.shirky.com/writings/semantic\\_syllogism.html](http://www.shirky.com/writings/semantic_syllogism.html)
22. Shirky, C. Ontology is Overrated: Categories, Links, and Tags. Clay Shirky's Writings About the Internet (2005). URL: [http://www.shirky.com/writings/ontology\\_overrated.html](http://www.shirky.com/writings/ontology_overrated.html)
23. Völkel, M. Personal Knowledge Management with Semantic Wikis (2005). URL: [http://www.xam.de/2005/12\\_voelkel\\_oren\\_SPKM\\_submission\\_eswc2006.pdf](http://www.xam.de/2005/12_voelkel_oren_SPKM_submission_eswc2006.pdf)
24. Völkel, M., Krötzsch, M., Vrandečić, D., Haller, H. Semantic Wikipedia. In proceedings of the 15th International World Wide Web Conference, Edinburgh, UK, May 22-26, 2006. URL: <http://www.aifb.uni-karlsruhe.de/WBS/hha/papers/SemanticWikipedia.pdf>
25. Völkel, M. et. al. Semantic Wiki State of The Art Paper. Under development. URL: [http://wiki.ontoworld.org/index.php/Semantic\\_Wiki\\_State\\_of\\_The\\_Art](http://wiki.ontoworld.org/index.php/Semantic_Wiki_State_of_The_Art)