

A WIKI-BASED APPROACH TO ENTERPRISE ARCHITECTURE DOCUMENTATION AND ANALYSIS

Sabine Buckl, Florian Matthes, Christian Neubert, Christian M. Schweda
Technische Universität München, Chair for Informatics 19 (sebis) Boltzmannstr. 3, 85748
Garching, Germany, {buckls,matthes,neubert,schweda}@in.tum.de

Abstract

Enterprise architecture (EA) management is a challenging task, modern enterprises have to face. This task is often addressed via organization-specific methodologies, which are implemented or derived from a respective EA management tool, or are at least partially aligned and supported by such tools. Nevertheless, especially when starting an EA management endeavor, the documentation of the EA is often not likely to satisfy the level of formalization, which is needed to employ an EA management tool. This paper tries to address the issue of starting EA management, more precise EA documentation and analysis, by utilizing a wiki-based approach. From there, we discuss which functions commonly implemented in wiki-based systems could be used in this context, which augmentations and extensions would be needed, and which potential impediments exist.

Keywords: Enterprise Architecture, Wikis, Information Modeling.

1 MOTIVATION

In recent years, companies had to face various challenging environmental influences, amongst others forcing them to ensure and further develop the alignment of their business and the corresponding IT support (see e.g. Laudon et al. 2006). Therefore, the management of the enterprise architecture (EA), which is commonly regarded as means to support this task, currently gains increasing attention in many medium-sized to large enterprises, reflected by a rising awareness for this field in academia (Langenberg and Wegmann 2004). Although a multitude of approaches to EA management with different backgrounds exists, ranging from practical origin (cf. Dern 2006, Niemann 2008, and Keller 2007) via standardization bodies, as e.g. the Open Group (TOGAF, cf. Open Group 2009) or the Object Management Group (CWM, cf. OMG 2001) and tool vendors (cf. Matthes et al. 2008), to academia (cf. Fischer et al. 2007, Frank 2002, Lankhorst 2005, or Ross 2006) no commonly accepted best-practice yet exists.

In spite of the number of different origins, all approaches center around the focal term *architecture*. According to ANSI/IEEE Std 1471-2000 architecture is defined as ‘the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.’ (IEEE 2000, S.3) Applying the aforementioned definition to the context of *enterprises*, the EA refers to the fundamental organization of an enterprise, embodied in its components (e.g. organizational units, stakeholders, locations, business processes), their relationships to each other (e.g. supports, hosts, is responsible for), and the principles (e.g. profit, continuity, innovation) governing its design and evolution. Additionally, the term *management* according to Mary Parker Follet refers to ‘the art of getting things done through people’ (van Aken 2005). Thereby, a typical management process is organized as a cycle containing the following phases: *Plan – Do – Check – Act* (Deming 1986, Sheward 1986). Put in the context of EA management, the *Plan* phase includes the development of future planning scenarios of the EA as well as the decision-making. The selected scenario is realized in the *Do* phase, followed by the analysis and evaluation of the implemented EA during the *Check* phase. The following iteration cycle is prepared in the *Act* phase via the identification of potential process improvements.

The goals of the ANSI/IEEE Std 1471-2000 are inter alia the *documentation*, *communication*, and *analysis* of architectures. Based on the definitions given above, the main tasks of EA management are:

- **Documentation:** In order to support the plan phase of the management cycle, the current (as-is) situation of the EA has to be documented. Thereby, elements on different layers ranging from *business & organizational* to *infrastructure* aspects (see Figure 1) have to be considered to provide a holistic view on the enterprise. Prior to gathering the data about the current situation, an information model¹ of the architecture has to be developed, which defines the elements and relationships inbetween constituting the EA. Besides the current situation, information about future states according to the plans has to be documented. Complementing the current and planned states of the EA an ideal target (to-be) state should be envisioned, which can be derived from the long-term vision of the enterprise.
- **Communication:** Gathering information from the different layers in the plan, do, and check phase, as introduced above, requires the involvement of a multitude of stakeholders, e.g. business process owners, project managers, business architects, etc. Although working for the same company, the terminologies used by these stakeholders differ widely. This communication issue is often referred to as *the communication gap between business and IT* (Lankes 2008, Schekkerman 2004). This gap is likely to hamper effective communication and

¹ The *information model* is sometimes also referred to as *meta model*. We prefer the term *information model*, which we regard to be more unambiguously (cf. Buckl et al. 2007b).

collaboration in the EA management process. Visualizations are a commonly accepted means to bridge this gap. Another challenge is connected to the aspect of historization and traceability of management decision (Buckl et al. 2008).

- **Analysis:** Concluding the management cycle, the current, planned, and target architecture of the enterprise have to be analyzed and evaluated in the check phase to support decision making and identify potential improvements (Johnson et al. 2007). The analysis results are finally used to improve the procedure of EA management itself in the act phase.

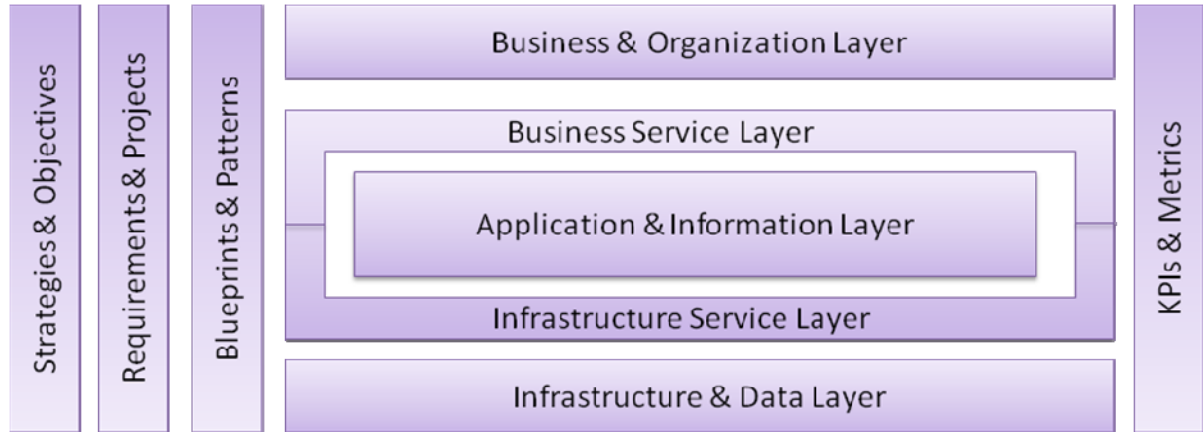


Figure 1: Layers and cross-functions, which form a holistic view on the enterprise

The *traditional* way to address the challenges of documentation, communication, and analysis in the context of EA management, as described above, is to select an EA management tool available (cf. Matthes 2008), and to use the inbuilt information model and methodology. Due to the high investments of such an approach, we propose an alternative lightweight wiki-based approach to EA management in this article, which we deem to be suitable especially for emerging EA management endeavors. Starting with a discussion about the application of wikis in related contexts, as e.g. software engineering, in Section 2, Section 3 explicates on the state-of-the-art in starting EA management endeavors. Thereby, we emphasize on the main tasks of EA management as introduced above. Following Section 4 introduces the wiki-based approach and discusses how it can be used to facilitate the challenges arising in the context of documentation, communication, and analysis of EA related aspects. The proposed approach is thereby complemented by methods for extracting structured information from wikis, by which we hope to promote a smooth transition from beginning small-scale EA management initiatives to mature enterprise-wide endeavors through leveraging information reuse. Finally, Section 5 concludes with a short outlook on upcoming or interesting research concerning the topic and indicates points of interest, which could be subject to further research initiatives.

2 SELECTED WIKI USAGES AND EXTENSIONS IN LITERATURE

A number of attempts have been undertaken to use wikis in enterprises as a tool for documentation and communication. Quite a lot of these attempts have been reported in papers on knowledge management (KM). Although EA management, as introduced above, can be regarded as an issue of knowledge management, we do not detail considerations on wikis from a pure KM perspective as found in literature. Instead, we discuss usage approaches and tool extensions directly focused on the three main tasks according to the IEEE, namely *documentation*, *communication*, and *analysis*.

The first approach is taken from the field of software engineering, which is strongly concerned with architectural topics and focuses on the tasks of documentation and communication. In a research project executed at Carnegie Mellon University a wiki-based system was used to collaboratively create

and maintain the documentation of the architecture of a single software system under development (Bachmann and Merson 2005). A key finding of research was that the structure of the documentation emerging in a wiki was different from the structure typically employed in a document-centric setting. The documentation in the wiki ended up at a different level of abstraction concerning architecture elements, i.e. complex elements were split up into single and separately maintained pages of which each documented a small portion of the element. This effect, which normally does not occur in document-centric settings, was considered helpful to reduce contention between the parties employed in documenting. This tendency of the information contributors to change the level of abstraction, if they regard an entity to be too complex to describe it in a single text, further promotes documentation maintenance. This leads to a more current documentation, although issues of assigning responsibilities for keeping pages up-to-date were experienced. The wiki-based system was also found useful in promoting reuse of documentation fragments, e.g. by linking related pages or embedding visualizations such like class diagrams (Bachmann and Merson 2005). Nevertheless, the latter aspect was not free of problems, as the system did not provide means for collaboratively editing visualizations. Thus, they were created offline and embedded as images – a fact greatly hampering their maintenance.

An application showing, how wiki entries created and maintained by an appropriately large number of users, can be used as a basis for collaboratively and iteratively building a documentation schema for a field of interest is given in Hepp et al. (2007). Thereby, the authors show how basic wiki concepts as the URIs, referring to described entities, the links between the entities, and the versioning history can be used to derive a basic and simple schema of the documented concepts. While Hepp et al. outlines that the approach taken might be greatly augmented e.g. using semantic wiki technologies, one of the core findings also holds for simple wiki-based systems: increasing stability of entity descriptions indicates the evolution of, what the authors call, *wiki consensus* on this entity's meaning.

In Auer and Lehmann (2007) an approach to lift *hidden semantic treasures* out of simple wiki-based systems is presented. In the approach especially wiki templates, as commonly used throughout the wikipedia, play an important role. These templates can be seen very similar to predefined forms, which were initially intended to provide consistently formatted graphic elements for certain content in specific articles. The articles using the same template can mostly be regarded to describe entities of the same type, e.g. all articles, which use the template *Settlement*, describe settlements. As a consequence, Auer and Lehmann the authors of the article developed an algorithm capable of searching a wiki (in this special case the wikipedia) for the occurrences of specific templates, which were then subsequently parsed and translated to a more formal representation. Having had on the one hand astonishing success with the extraction, some caveats with the approach could be identified, especially concerning the design of the used templates. Some templates are made up in a way, that (semantic) information and presentation information are not strictly separated. Additionally, for the same concept several templates might exist. As a consequence Auer and Lehmann (2007) provide guidelines for designing semantically rich templates, which we will refer to in the presentation of our approach.

3 STARTING AN EA MANAGEMENT INITIATIVE – STATE OF THE ART

In our research project *Software Cartography*, we gained insights regarding the first steps taken to establish an EA management endeavor in different global acting companies, e.g. Allianz, BMW, MunichRe, Siemens, T-Com. Accordingly, a typical emerging EA management initiative typically starts with the documentation of the as-is situation of the EA. Thereby, seldomly a strict and predefined information model is utilized at the beginning of an EA management initiative. Thus, the documentation of the EA is often left to text or spreadsheet documents without a formal structure. This method is often useful to gain first insights on the EA and, more important, to satisfy information needs without having to first explicate and formally describe them. Additionally, the method is commonly regarded as more appropriate for starting with EA management, as the creation or selection

of an EA information model can be considered a labor intensive task, which often employs various review-cycles.

Nevertheless, when the EA management endeavors become more mature and multi-partied, commonly an object-oriented information model for the EA concepts is used. The same is true with the rising demand for aggregated information representations such as e.g. computed metric values (see Lankes 2008), which are mostly grounded on object-oriented information models. Numerous of these models currently exist, heavily researched in the academic community (see e.g. Braun and Winter 2005 or Lankhorst 2005), by tool vendors (cf. Matthes et al. 2008) as well as standardization bodies (e.g. the Object Management Group or The Open Group), and developed by the companies practicing EA management (see e.g. Halbhuber 2004, Lauschke 2005, Brendebach 2005). This fact is further promoted, as no common and fully accepted standard information model has yet been published – with some researchers, tool vendors, and EA management practitioners claiming, that such a model suitable for any company and any way of performing EA management does not exist (Buckl et al. 2007a or Kurpjuweit and Winter 2007). Whereas Kurpjuweit and Winter propose an stakeholder-centered approach to information model creation, Buckl et al 2007a and Buckl et al. 2008 presents a concern-centered approach.

With the growing complexity of the body of management and the rising number of people to be involved as information suppliers, spreadsheets and text documents do not scale appropriately, negatively influencing the overall acceptance of the initiative, e.g. due to arising problems of maintaining the consistency (Fischer et al. 2007). A way to bring EA documentation and communication efforts to the *next level* is the introduction of an EA management tool, into which the previously gathered data shall be imported. This step can be considered as a turning point in two ways: Firstly, up to that point EA management and especially the documentation efforts do not comprise a major investment and are thus likely to emerge as bottom-up initiatives. In contrast switching to a tool not already present in the enterprise commonly cannot be done without having management support as this poses a great investment for the company. Secondly, the tabular and textual documentations rarely adhere to an explicated and formal structure. Bringing this information into an EA management tool can be regarded a quite complicated task, as the majority of tools uses an object-oriented information model for the governed information (cf. Matthes 2008 et al.). Therefore, the structure behind the information, previously gathered in spreadsheets and text documents, has to be explicated and mapped to the structure provided by the future EA management tool.

The process of mapping a grown structure, which evolved in a process potentially without centralized control, to the structure as incorporated in the tool, can be regarded to be challenging (see Matthes et al. 2008). This is on the one hand true, if the tool comes shipped with a predefined and fixed model, to which all the concepts have to be mapped. More often than not the predefined information model demands some concepts to be maintained, about which up to that point no information was gathered. On the other hand, if the tool provides adaptability of the underlying information model, the issue to tackle does not become smaller. A variety of different stakeholders throughout the enterprise wants or needs to be involved in creating and developing a company specific information model, which supports the analyses needed to perform enterprise-specific EA management tasks. Beside the different concerns raised by the different stakeholder, also definitional issue are likely to arise, because terms as *application* are widely used and are thus strongly overloaded concerning their meaning.

4 A WIKI-BASED APPROACH ...

Based on the findings of the work of Bachmann and Merson 2005, we discuss how a wiki-based approach facilitates EA management endeavors. Thereby, we structure our discussion along the three main tasks *documentation*, *communication* and *analysis*, and show possible impediments for effectively using wikis in these areas.

4.1 ... to EA documentation

Applying tags to objects is prominent method for categorizing concepts especially in the web 2.0 environment (cf. <http://del.icio.us>). Also many wiki-systems utilize this mechanism, to allow their users to contribute classification information, which constitute bottom-up taxonomies, so called *folksonomies*. Further, widely used wiki-systems (see e.g. <http://www.wikipedia.org>) employ the mechanism of templating to provide pre-formatted and structured building blocks for wiki-pages. We subsequently discuss how these two techniques can be used to facilitate the creation of an EA information model consisting of classes and properties².

4.1.1 Collaborative Tagging

Golder and Huberman (2005) introduce a classification for different kinds of tags, which are used throughout a collaborative tagging environment. Based on this classification, we discuss which types of tags can be used to facilitate the collaborative creation of an EA information model. These are on the one hand the *what-it-is-about* tags and *what-it-is*, which can be regarded to be quite identical in the context of EA documentation. Such tags commonly form the overwhelming majority of tags used in collaborative tagging environments. On the other hand, the *who-owns-it* tags, which are used especially for tagging blogs, can also be used in EA documentation, although ownership has a different notion there. Owning a concept means be responsible for this specific concept. Such information should better not be expressed via tags but using properties as described in the next subsection. Therefore, we do not expect ownership tags to be very frequent in documentation wikis. A last type of tags, the *refining-categories* tags are not used on their own, but together with tags of other types for refining their semantics and creating subcategories. These tags can provide information valuable to EA information modelling, although we do not expect them to be more frequent as in general tagging environments.

The tag type discussed above can be considered very useful for finding and refining classes in a similar way as e.g. in object oriented design. There, the creation of classes is simply spoken related to grouping objects, which share common properties, and finding an abstract label for describing the group. Nevertheless, especially in complex and evolving environments, as e.g. an enterprise such a classification might not easily be found. Therefore, collaborative tagging can provide a useful contribution, when EA concepts documented in the wiki are applied tags, e.g. "business application" or "server". Other tags as "important", which widely used in tagging environments, are nevertheless not directly related to classes in the information model, but merely have a user-related semantics. They can notwithstanding be used for our approach, cf. Section 4.2.1.

4.1.2 Open templating mechanism for property modeling

Wiki templates, as discussed above, provide a mechanism for defining values for a set of fixed properties of a certain type of object, e.g. a settlement. Nevertheless, as neither a set of fixed types, which could be used for defining templates, nor a fixed set of properties exist, the templating mechanism cannot directly be applied here. Therefore, we propose an adapted form of the mechanism, which we call *open templating mechanism*. In the context of the mechanism the template itself is reduced to a simple table, in which the user can add rows and thereby properties as well as corresponding values. As a consequence, the user can freely introduce properties and apply values to them. Because of the nature of wiki systems these values are only textual, no data types are assigned to

² Herein, we use the terminology of the Essential Metaobject facility (EMOF), see OMG (2006). Therein, no distinct concept for modeling relationships exists; instead they are represented as connected properties.

the properties, which might cause several problems discussed below. Nevertheless, semi-complex data structures (e.g. arrays, sets) can be utilized to bundle composite values.

In order to prevent anarchic evolution of templates, again a recommender mechanism is added, providing properties, which are potentially relevant to the object currently described. The relevance therein is not computed from the overall frequency of that property but from a correlation analysis between the property and the tags, which are applied to the described object. Thereby, it is determined how frequent given property and tag are used in conjunction compared to the number of usages in general. We regard this method to be appropriate for quickly and easily finding the important properties. In order to display the recommendation of the system, again a visualization resembling a tag-cloud could be utilized, in which the properties most correlated are displayed as more relevant.

4.1.3 Potential Impediments

Collaborative tagging and open templating might be well-suited for the above context; nevertheless some impediments might hamper the development of a semi-structured EA documentation. Subsequently, we outline two of them, which can negatively influence user acceptance. This acceptance is generally considered a critical success factor for tool-supported, bottom-up knowledge management endeavors.

A phenomenon repeatedly reported in the context of classification attempts, is what was called the *base level* in Tanaka and Taylor (1991). When asked to classify an object, people tend to use terms, which are neither strongly generalized nor specific according to their distinct knowledge backgrounds. The chosen intermediate level between generalization and specificity is called the base level of the particular person. As a consequence of the base level being specific for every person, the expertise in the domain, from which objects have to be classified, greatly influences the specificity or generality of the used classification terms. This effect might especially apply in the area of EA management, where stakeholders from many different areas are contributing to a holistic documentation of the EA. Therein, they are likely to classify according to their domain expertise, leading to issues, which potentially impede the development of a common and consistent information model.

Wiki-entries and therefore also the values contained in templates are mainly textual. While this can be seen as an important success factor for wiki-based systems in general, the fact has some negative influence in information modeling. The properties' values in the templates are pure text values, even if they represent information about date or numbers. As a consequence, no type and consistency checking can be performed - endangering the overall quality of the data collected. In addition, when a full migration of the semi-structured wiki-based EA documentation to a fully object-oriented model should be performed, the data types of the properties had to be determined. While possibly some of this determination may be performed automatically, no one expects that to be possible in all cases. This is especially true, if properties representing relationships between objects are considered.

4.2 ... to EA information communication

The different stakeholders of EA management endeavors have distinct information demands and perform specific activities during the management processes. According to the information demands, these stakeholders can be assigned to groups – a functionality, which many wiki-systems provide out-of-the-box. The specific activities can be connected to roles in the wiki, such that access rights can also be granted based on the role-description taken by a person. The formation of virtual communities of practice of different stakeholders as wiki-groups can be leveraged to facilitate communication among these stakeholders by means of collaborative bookmarking or watch lists.

4.2.1 Collaborative bookmarking

In contrast to collaborative tagging, the collaborative bookmarking does not assign classifying tags to the objects, but assigns contextualizing tags, which have no general meaning, but provide sense for the user (or user group), from which the tag originates. Golder and Huberman (2005) provide a classification for these types of tags into *self-referential* and *task-organization* tags. Self reference tags are used to express a relationship of the tagged object to the user. These relationships do mostly not refer to formal and organizationally established responsibilities, but refer to intrinsic areas of interest of the specific user. Task organization tags, as e.g. "TOBEREAD" or "TODO", are commonly used in tagging environments to keep track of open activities or inform fellow users of open tasks. Both types of contextual tags do not contribute to the creation of an EA information model, but are important means for information distribution among a certain group, e.g. used as hints during the collaborative creation of an EA documentation artefact.

In order to clearly separate the different types of tags – the classifying ones needed for EA information modelling, and the contextualizing ones needed for EA information communication, we propose an additional concept, which might be helpful in this context. The user should be given the possibility to chose, whether a given tag is visible in *public*, to a *selected user group only*, or *private*, according the aforementioned virtual communities of practice, which the user participates in. This also gives rise to the idea to extend the tag recommender system, such that a user is recommended tags in respect to the groups, he belongs to. Such a recommender system can be helpful to prevent the evolution of numerous synonymic tags in the wiki. The tags recommended are therein sorted by relevance and displayed in an appropriate way, e.g. in a tag-cloud.

4.2.2 Watch lists and change feeds

As the documentation of the EA can be regarded an effort of strongly collaborative character, group related notification in case of specific changes and communication of ongoing documentation initiatives is often considered to be important by practitioners for a number of reasons:

- The person responsible for certain parts of the EA can get a notification, if the documentation of one of these elements is adapted, e.g. to perform quality assurance.
- The head of a group of people employed with documenting parts of the EA can keep track on the progress made in the documentation endeavour.

Watch lists and change feeds, as implemented in many different wiki- systems, may be appropriate to address the notification demands as sketched above. Nevertheless, further demands for notification may exist, e.g. a user might want to get a message on pages, which have not been changed for a certain period. These messages might give indications on parts of the EA documentation becoming outdated or parts being well established. While such a feature is not commonly supported in simple wiki-systems, the information on the last update of a page is stored and could be easily used to satisfy the notification demand as outlined before.

4.2.3 Potential impediments

Each of the communication facilities outlined above has some subtle complexities assigned, which may lead to sub-optimal usage in an EA communication environment. The collaborative bookmarking might fall for his apparent similarity to the collaborative tagging facility. While the latter is intended to evolve a bottom-up classification schema for concepts, the former can be used as mnemonic for a user or to guide the attention of a user group. Nevertheless, one cannot expect the users of the wiki-system to keep this strict separation alive. Hence, the classification schema might get "contaminated" with contextualizing tags, which have no meaning to the public. The opposite effect might also occur – users might classify objects in the wiki private, such that the information model cannot benefit from.

The watch lists might fall for legal problems, as they could be misused as means to observe and track employee behaviour in the wiki-system. Beside this problem, the watch lists facilitate a one-way usage paradigm, where users only aggregate and consume information, but do not directly contribute to the information and documentation base. Finally, watch lists might, similar to task organization tags, be helpful for directing user attention. Nevertheless, an excessive supply of such list is most likely to distract the attention of the users.

4.3 ... to EA Analysis

The documentation of the EA as incorporated in the wiki-system has to be analyzed in various ways to support the EA management process. Different wiki-systems provide functionalities, which are useful for performing EA analyses. These functionalities range from *versioning*, which is a common feature for most wiki-systems, to *textual querying*, which is supported by some semantic wiki-systems (e.g. the semantic media wiki – see <http://www.semantic-mediawiki.org>). By textual querying, semi-structured information from wiki-pages is queried against, and the results of the query are presented in a tabular report.

4.3.1 Versioning

Most wiki- systems provide versioning of pages, i.e. the history of past versions of the page is stored in the underlying database of the wiki. Thereby, two important aspects of EA analysis are facilitated. At first, the description of the EA or parts thereof is historized – a feature, which is often requested by EA practitioners to ensure traceability of management decisions and project changes. In contrast, many of the current tools for EA management (see Matthes et al. 2008) do only rudimentarily support versioning. Secondly, if mechanisms as collaborative tagging and open templating are used to develop and evolve an information model for EA management, versioning of wiki-pages also contributes to a historization of information models.

The possibility to distinguish between *minor* and *normal changes* of wiki-pages additionally allows to clearly separate maintenance activities in the documentation from situation, where substantial change in the underlying architecture is documented in the wiki. If the wiki-system further supports to give comments on the changes, especially for the normal ones, this functionality can be used to relate a documented architectural change to the causing EA project. This traceability of the projects, which drive the managed evolution of the EA, is often regarded to be important for EA planning and analysis (Buckl et al. 2009). Finally, the option to mark a change as minor allows to reduce the amount of information, which shows up in watch lists, such that the user attention is not distracted by these maintenance activities but is guided towards “real” architectural changes.

4.3.2 Textual querying

Textual querying allows to access both the textual and the semi-structured part of wiki-pages in a more convenient way. In particular, tabular views on the structured information can be created via queries, which can be used to get an overview on specific concepts, e.g. all objects tagged with “business application”. In this overview, also different kinds of sorting can be applied and comparisons of objects of the same type are performed easily.

A tabular listing of “business application” concepts might additionally be helpful in guiding the evolution of the information model, as maintained in the wiki-system. While at the beginning of a collaborative documentation endeavor, frequency counts and correlation analyses might be appropriate methods for determining the importance of a tag and the corresponding properties used in the open templates, the evolving information model may at some point become overloaded and partially anarchic. By comparing the property supply for a certain type of object, a domain expert for EA modelling can decide to make some of the properties mandatory, leading to increased consistency of

the information model. Having taken such a decision, the tag-cloud recommender functionality as introduced above, could be adapted to incorporate the effects of the decision, e.g. by colour coding certain elements or enlarging their font-size. This could be used to leverage the documentation of specific concepts, which are especially necessary for a distinct EA management endeavour.

4.3.3 *Potential impediments*

The basic facilities for comparing selected versions of a wiki-page are often somewhat limited in respect to the semi-structured information supplied by open-templating mechanisms. Therefore, especially functions for executing comparisons within the templates can be considered beneficial, so that the user can see a tabular report on the values supplied for properties in the selected versions. For such a comparison, it would also make sense to compare more than two versions of the object – a feature also not commonly supported in wiki-systems.

The synopsis of different objects of the same type or versions of the same object over time can, as alluded to above, be used to facilitate decisions on the EA information model. Nevertheless, if a decision on certain properties is taken, e.g. making them mandatory, the bottom up development of the EA information model is disturbed, which might cause a decrease of user acceptance. Hence, any top-down adaptation to the community developed model should be carefully assessed to prevent a downturn in user motivation.

Textual queries are only the first step on the way to conveniently present semi-structured information to a broader user community. Going beyond simple tabular reports, graphical visualizations of the EA or parts thereof are regarded as an important instrument (Lankes et al. 2005) to both support and promote the EA management effort as well as to increase the awareness for that topic among the company's employees. Such visualizations, called Software Maps in the context of documentation of the EA, can according to the approach presented in Buckl et al. 2007b be created from information about the EA, if all concepts displayed are sufficiently documented. This should be the case, if open templating mechanisms are employed in the wiki to support semi-structured information provision. Here, we see a potential synergy effect, if the wiki- system can provide access to automatically generated visualizations displaying concepts within the EA, which have been described sufficiently detailed. Such visualizations, if comprising which information has to be maintained for a specific concept in order to be displayed therein, could then leverage the maintenance of that data. Thereby, users of the documentation wiki are encouraged to supply additional information about the EA constituents, which they are responsible for, to make sure, that these objects are displayed in the visualizations of the EA.

5 OUTLOOK

In this article, we presented an approach to use wiki-based systems for documenting, communicating, and analyzing the EA. Furthermore, we elaborated on different aspects of wiki-based systems, which might be helpful in this context. Nevertheless, the approach presented has yet not been validated in practice. We assume, that especially medium-sized companies may benefit from the presented techniques for starting EA management, while larger enterprises are likely to stay to the more formal and structured tools.

Further aspects of interest regarding the approach are e.g. scenarios to use rating functions on wiki-pages to allow users to provide feedback in a structured and uniform way. Concerning these functions a multitude of implications, e.g. on user motivation would have to be considered. Additionally, these ratings could provide valuable input to the analysis of documentation templates, although a clear correlation between a low rating and the respective template used, is not likely to be easily determinable.

6 REFERENCES

- Auer, S. and Lehmann, J. (2007). What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In: Franconi et al. editors, Proceedings of European Semantic Web Conference (ESWC'07), LNCS 4519, Springer, 503-517.
- Buckl, S. and Ernst, A. and Matthes, F. and Schweda, C. M. (2009). An Information Model for Managed Application Landscape Evolution. In: Journal of Enterprise Architecture (JEA), 12-26.
- Buckl, S. and Ernst, A. and Lankes, J. and Matthes, F. (2008). Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical Report TB 0801, Chair for Informatics 19, Technische Universität München.
- Buckl, S. and Ernst, A. and Lankes, J. and Schneider, K. and Schweda, C. Wittenburg, A. (2007a). A Pattern based Approach for constructing Enterprise Architecture Management Information Models. In A. Oberweis, C. Weinhardt, H. Gimpel, A. Koschmider, V. Pankratius, and Schnizler, editors, Wirtschaftsinformatik 2007, pages 145–162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.
- Buckl, S. and Ernst, A. and Lankes, J. and Matthes, F. and Schweda, C. and Wittenburg, A. (2007b). Generating Visualizations of Enterprise Architectures using Model Transformation (extended version). In: Enterprise Modeling and Information Systems Architectures - An International Journal, 2 (2), 3-13
- Buckl, S. and Ernst, A. and Matthes, F. and Schweda, C.M. (2008). An Information Model for Landscape Management – Discussion Temporality Aspects. In: The 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008), Sydney, Australia, 63-77.
- Bachmann, F. and Merson, P. (2005). Experience Using the Web-Based Tool Wiki for Architecture Documentation. Technical report, Software Engineering Institute, Carnegie Mellon University.
- Brendebach, K. (2005). Integrierte Modelle und Sichten für das IT-Management. Diploma thesis, Fakultät für Informatik, Technische Universität München.
- Braun, C. and Winter, R. (2005). MA Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform. In Enterprise Modelling and Information System Architectures (EMISA), pages 64–79, Klagenfurt, Österreich.
- Dern, G. (2006). Management von IT-Architekturen. 2nd edition, Vieweg, Wiesbaden.
- Deming, W.E. (1986). Out of the Crisis. Massachusetts Institute of Technology, Center for Advanced Engineering Study.
- Frank, U. (2002). Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. Proceedings of the Hawaii International Conference on System Sciences (HICSS-35).
- Fischer R. and Aier, S. and Winter, R. (2007). A Federated Approach to Enterprise Architecture Model Maintenance. In: Reichert, M and Strecker, S. and Turowski, K. (ed.) Enterprise Modelling and Information Systems Architectures – Concepts and Applications. Proceedings of the 2nd Int'l Workshop EMISA 2007. Gesellschaft für Informatik, Köln, Bonn, 9-22.
- Golder, S. and Huberman, B. A. (2005). The Structure of Collaborative Tagging Systems. Information Dynamics Lab: HP Labs, Palo Alto, USA, <http://arxiv.org/abs/cs.DL/0508082>.
- Gamma, E. and Helm, R. and Johnson, R. and Vlissides, J. (1994). Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley Longham.
- Halbhuber, T. (2004). Entwicklung eines Informationsmodells für das IT-Management. Diploma thesis, Fakultät für Maschinenwesen, Technische Universität München.
- Hepp, M. and Siorpaes, K. and Bachlechner, D. (2007). Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management. IEEE Internet Computing, 11(5):54–65.
- IEEE (2000). IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std 1471-2000). IEEE Computer Society, New York, NY.
- Johnson, P. and Lagerström, R. and Närman, P. and Simonsson, M. (2007). Enterprise architecture analysis with extended influence diagrams. Journal on Informations Systems Frontiers, Issue 2-3, Springer.

- Keller, W. (2007). IT-Unternehmensarchitekturen. dpunkt, Heidelberg, Germany.
- Kurpjuweit, S. and Winter, R. (2007). Viewpoint-based Meta Model Engineering. In M. Reichert, S. Strecker, and K. Turowski, editors, Enterprise Modelling and Information Systems Architectures, Gesellschaft für Informatik, Bonn, Germany, 143–161.
- Lankhorst, M. (2005). Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, December.
- Lauschke, S. (2005). Softwarekartographie: Analyse und Darstellung der IT-Landschaft eines mittelständischen Unternehmens. Bachelor thesis, Fakultät für Informatik, Technische Universität München.
- Laudon, K.C. and Laudon, J.P. and Schoder, D. (2006). Wirtschaftsinformatik – Eine Einführung. Pearson Studium, München.
- Lankes, J. and Matthes, F. and Wittenburg, A. (2005). Softwarekartographie: Systematische Darstellung von Anwendungslandschaften. In Wirtschaftsinformatik 2005, Bamberg, Germany.
- Lankes, J. (2008). Metrics for Application Landscapes – Status Quo, Development, and a Case Study. PhD Thesis, Technische Universität München, Fakultät für Informatik.
- Langenberg, K. and Wegmann, A. (2004). Enterprise Architecture: What Aspects is Current Research Targeting? Technical report, Ecole Polytechnique Fédérale de Lausanne, Laboratory of Systemic Modeling.
- Matthes, F. and Buckl, S. and Leitel, J. and Schweda, C.M. (2008). Enterprise Architecture Tool Survey 2008. TU München, Chair for Informatics 19 (sebis), Germany.
- Niemann, K. D. (2008). From Enterprise Architecture to IT Governance Vieweg, Wiesbaden.
- OMG (2001). Common Warehouse Metamodel specification (ad/01-02-01).
- OMG (2006). Meta Object Facility (MOF) Core Specification, version 2.0 (formal/06-01-01).
- Ross, J. (2006). Enterprise Architecture: Driving Business Benefits from IT. 359, Massachusetts Institute of Technology, Center for Information Systems Research (CISR), Cambridge MA.
- Schekkerman, J. (2004). How to survive in the jungle of Enterprise Architecture Frameworks. Trafford, Victoria B.C., Canada.
- Shewart, W. A. (1986) Statistical Method from the Viewpoint of Quality Control. Dover Publication, New York.
- Tanaka, J. W. and Taylor, M. (1991). Object categories and expertise: Is the basic level in the eye of the beholder? Cognitive Psychology, 23:457–482.
- The Open Group (2009). TOGAF "Enterprise Edition" Version 9.0.
- van Aken, J.A. (2005). Management Research as a Design Science: Articulating Research Products of Mode 2 Knowledge Production in Management. British Journal of Management, Vol. 16, 19-36. 2.