

Lightweight Approach For Enterprise Architecture Modeling and Documentation

Sabine Buckl, Florian Matthes, Christian Neubert, and Christian M. Schweda

Technische Universität München, Institute for Informatics,
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl,matthes,neubert,schweda}@in.tum.de
<http://www.systemcartography.info>

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. 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 address the issue of starting EA management, more precise EA documentation and analysis, by utilizing a wiki-based approach.

Today’s enterprise wiki-systems provide numerous of services such as collaborative authoring, tagging, bookmarking, awareness, commenting, rating, linking, search, social networking, versioning, access control [1]. Furthermore, recent efforts aim to allow annotating semantic data within the wiki pages’ content in enterprise wiki-Systems. How these functions can be applied for collaborative enterprise architecture (EA) modeling and documentation is explained by means of the following scenario. We illustrate how to model and describe the concepts shown in Figure 1 and corresponding instances in a wiki-based manner.

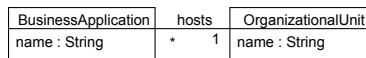


Fig. 1. UML diagram of the concepts business application and organizational unit

In order to model concepts and attributes (top-down) we use an open templating mechanism [2]. Thereby two wiki-pages describing the concepts business application (BA) (cf. Figure 1) and organizational unit (OU) with their name attributes are defined. To mark the wiki-page as *template* reserved tags are utilized, e.g. `typedef`. Since the templates are wiki pages, the concept BA and its attributes can be described textual in the wiki pages’ content. By using semantic annotations [3] a ”name” attribute can be defined by marking the literal ”name” within the text. All attribute-definitions could be shown in a tabular view for purpose of exposing semi-structured parts of the wiki-page. In some cases it can be beneficial to define mandatory attributes for a template, e.g. all BAs must provide a name attribute. This could be achieved by specifying rules on template level. In contrast to templates, concepts can also evolve dynamically on ”instantiation” level (bottom-up). Thereby, a wiki-page is created to describe a

concrete instance of a BA (cf. Figure 1). To mark the page to be a BA the page can either be tagged "business application" on page level or the literal "business application" can be marked with semantic annotation within the text, e.g. "is-a". In the latter case a new template (providing all currently in the text defined attributes) could be created if no BA template exists so far. If a template is already defined, the assigned attributes (having no values) could be shown in a tabular view. Furthermore, additional attributes, which are frequently used in other BA wiki-pages could be recommended by showing them in a tabular view.

The figure shows two side-by-side screenshots of a wiki page. The left screenshot shows a template for 'BusinessApplication' with various annotations. The right screenshot shows an instance of this template, 'SAP Business Suite', with a tabular view of its attributes and a list of organizational units it is referenced by.

Left Screenshot (Template):

- Tags: `prototyp`, `template`, `typed` (Annotation: "instance" of BusinessApplication template)
- Template definition for the concept BusinessApplication: "A software system, which is part of an information system within an organization. An business application references its hosting `OrganizationalUnit` and provides a `name`." (Annotation: "marked up text, representing the value of the name attribute")
- Attributes: (Annotation: "tabular view of all attribute-value pairs assigned to the page")
- Attribute Value table:

Attribute	Value
Name	SAP Business Suite

 (Annotation: "customized embedded query showing the additional attribute 'Abbreviation' for organizational units")
- Referenced by: `OrganizationalUnit` (Annotation: "Referenced template")
- Referencing template: `OrganizationalUnit` (Annotation: "Attribute declaration in the text (semantic markup)")

Right Screenshot (Instance):

- Tags: `businessapplication` (Annotation: "instance" of BusinessApplication template)
- SAP Business Suite (Annotation: "marked up text, representing the value of the name attribute")
- Attributes: (Annotation: "tabular view of all attribute-value pairs assigned to the page")
- Attribute Value table:

Attribute	Value
Name	SAP Business Suite

 (Annotation: "customized embedded query showing the additional attribute 'Abbreviation' for organizational units")
- OrganizationalUnits:

Resource	Name	Abbreviation
http://www.matthes.in.tum.de/wikis/sebs/marketing	Marketing	MKG
http://www.matthes.in.tum.de/wikis/sebs/human-resources	Human Resources	HR

 (Annotation: "generic view of all referencing instances")
- Referenced by: `Human Resources`, `Marketing` (Annotation: "generic view of all referencing instances")

Table 1. Template for the concept BusinessApplication (left) and BusinessApplication instance "SAP Business" (right)

Association between concepts can be expressed via hyperlinks to other wiki-page instances representing an OU, i.e. which are tagged "organizational unit". The latter wiki-page could list all pages (as links) it is referenced by, by means of backlinks, i.e. links to all BAs the OU is using. This could either be achieved in a generic manner or by embedding custom queries in the page content. In both cases it might be useful to customize which attributes of the referencing concept to be shown. In contrast to define links on concrete (BA) instances (bottom-up), hyperlinks could also be used on template level to express directional relations between concepts (top-down). E.g., the literal "OrganizationalUnit" shown in Figure 1(left) could be a hyperlink to the OU-template. To express that exactly one OU is referenced by a link on instance level rules could be applied (cf. mandatory attributes).

Summary: Both ways of lightweight EA modeling can be applied, top down by using predefined wiki-templates as well as bottom up by evolving the model indirectly on the instance level. Wiki-pages can be utilized for documentation of EA concepts and their instances collaboratively. Besides that, other services of today's wiki-systems can be utilized, e.g lists and change feeds to get notified when the model or the documentation changes.

References

1. T. Büchner, F. Matthes, and C. Neubert. A concept and service based analysis of commercial and open source enterprise 2.0 tools. In *KMIS*, pages 37–45, 2009.
2. S. Buckl, F. Matthes, C. Neubert, and C. M. Schweda. A wiki-based approach to enterprise architecture documentation and analysis. In *The 17th European Conference on Information Systems (ECIS) – Information Systems in a Globalizing World: Challenges, Ethics and Practices*, 8. – 10. June 2009, Verona, Italy, pages 2192–2204, Verona, Italy, 2009.
3. The Dublin Core Metadata Initiative. The dublin core metadata initiative. <http://dublincore.org/> (cited 2010-03-18), 2010.