# The BAsAS Architecture For Semantic Web Annotations

## [Poster Abstract] *

### Valentin Zacharias
ontoprise GmbH
Amalienabdstr. 36 (Raumfabrik 29)
76227 Karlsruhe, GERMANY
zach@ontoprise.de

## ABSTRACT

We describe a generic architecture for the (semi-automatic) creation, storage and querying for annotations of web resources. Our BAsAS architecture uses recent advances from the Semantic Web and Web 2.0 communities to make Semantic Web annotations a reality. The BAsAS architecture makes it easy for users to start to annotate and easy for developer to use the annotations that get created.

Besides describing the general architecture we will also detail an implementation of this architecture build for a Semantic Web community portal.

## 1. INTRODUCTION

The Semantic Web is the vision of having data on the web defined and linked in such a way, that it can be used by machines not just for display purposes, but for integration, automation and reuse of data across various applications.
An Annotation is a piece of information or knowledge entity that is associated with a document or part therof. In this paper we focus on *Semantic Web Annotation*, understood as annotations of web resources with the goal of lifting them into the semantic web. Unstructured content outside of the semantic web is annotated and these annotations become part of the Semantic Web. In this way (a part of) the content of the document becomes part of the Semantic Web, the document gets woven into the Semantic Web. Research that understands Semantic Web Annotations in this way must discuss the question of how the data created in the annotation process becomes part of the Semantic Web, how it is retrievable and usable for computer agents.

## 1.1 Contribution

What we are trying to do is similar to the Annotea[2][3] infrastructure: define an architecture that can form the backbone for Semantic Web annotations. We, however, improve

---

*A full version of this paper is available at www.valentin-zacharias.de/papers/BAsAS.pdf

on the well known Annotea idea in three crucial points: we show how the SPARQL language solves the problem of queries for annotations. We demonstrate how AJAX can be employed to build an annotation tool that is more lightweight and less browser dependent than anything that exists for Annotea. Finally we show a simple architecture that allows semi-automatic annotation while still retaining the extrem lighweight property of the annotation tools. To a large part this paper can be understood as an attempt to update the ideas pioneered by annotea using technology that was not available at the time it was initially conceived.

## 2. ARCHITECTURE

The BAsAS architecture is so named for its main characteristics: **B**rowser, **A**nnotation **s**erver, **A**JAX and **S**PARQL. These parts and their interaction are shown in Figure 1. At the core of the architecture is a Semantic Web data store that holds the annotations and that offers a SPARQL interface to access them. It should also offer an HTML interface for browser based access.
Annotations are done by the user with the help of an AJAX interface to an annotation server. This annotation server helps to keep the client side of the annotation as lighweight as possible, in particular any complex algorithms for automatic annotation are run on this server. The annotation server is also responsible for receiving the information from the annotation interface and translating it into update requests for the data store. Only with the help of such a server component is it possible to build an extrem lightweight annotation component.
The annotation component needs some way to establish the context it is called in for this gives the information on what the user wants to annotate. This context can be established by a lighweight browser "plugin". As we will see this needs not be an actual browser plugin. A short snippet of JavaScript that browsers treat like a bookmark is sufficient in many cases. We will also show that sometimes not even that is necessary.

## 3. UNITRACC COMMUNITY PORTAL

We are currently implementing the architecture described in the preceeding chapter for the use in the system unitracc.

Unitracc("Underground Infrastructure Training and Competence Center") is an internet based e-learning system for the area of canalization. Unitracc also contains a collection of web based tools that help public authorities manage and monitor underground infrastructure. The system already
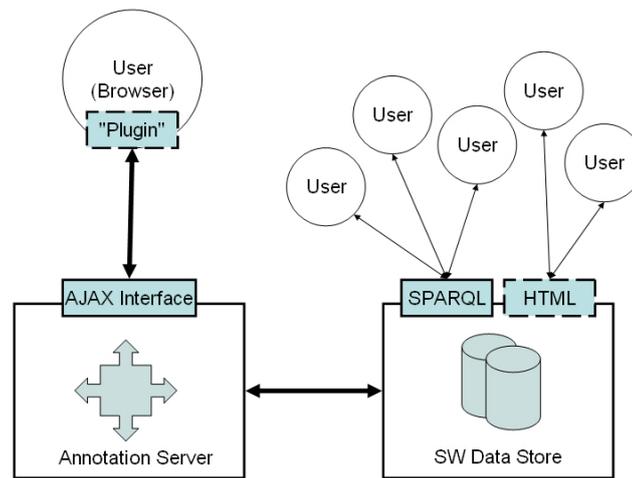
**Figure 1: High level view of the BAsAS architecture**

contains a large number of information units, especially enhanced digital versions of two standard textbooks about canalization. Unitracc is developed by the company Prof. Dr. Ing. Stein & Partner GmbH, a leading engineering firm whose founder is also the author of many standard works of technical literature. The development has been funded in part by the German Federal Ministry of Education and Research. Access to unitracc is available on a subscription basis, the target audience ranges from beginning trainees and their teachers to architects.

We are currently in the process of extending this platform in the direction of a community portal. In addition to the core content supplied by the creators of unitracc there should be an outer layer of user created content, such as comments, technical manuals uploaded by tool vendors or annotations of web sites. We expect this layer of content to be less reliable but also to be more current and diverse. We chose to build the this part of the platform as open as possible: we believe that the openness increases the motivation of people to contribute. We hope that people will be less reluctant to put effort into a commercial site when they know that this content can be used by everyone.

## 3.1 Annotation

Users of the site will be encouraged to annotate the web resources relavant to unitracc using a simple annotation tool. For the time beeing we have settled on a very simple annotation format: the annotation is always for the whole web page and it only consists of "has topic" relations to the topics from the unitracc topic hierarchy. The integration of the annotation tool with the browser is done with a bookmarklet. The annotation interface is realized with AJAX, it supports the user mainly by offering auto complete for the topics. It also receives the proposals from the automatic categorization performed by the annotation servlet and displays them to the user. A screenshot of the annotation interface is shown in Figure 2. The topic hierarchy that forms that basis for the annotations can also be edited using an AJAX interface. This interface also allows many people to change the topic hierarchy simultanously. Changes to the
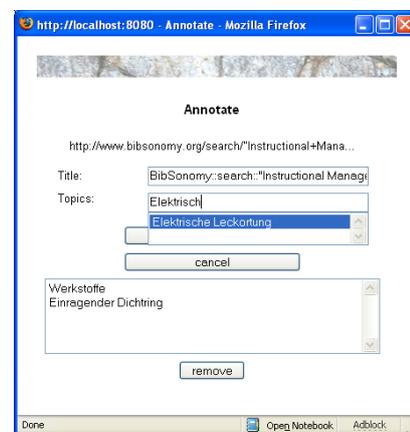


**Figure 2: The unitracc annotation interface**

topic hierarchy are currently meant to be done only by the administrators of unitracc.

The annotation interface, like the topic hierarchy editor, is created using the Google Web Toolkit[1].

## 4. ACKNOWLEDGMENTS

## 5. REFERENCES

[1] Google web toolkit,
    http://code.google.com/webtoolkit/.

[2] J. Kahan and M.-R. Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *WWW*, pages 623–632, 2001.

[3] M.-R. Koivunena, R. Swick, and E. Prud'hommeaux. Annotea shared bookmarks. In *Proceedings of the KCAP03 workshop*, 2003.