Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2011 (WowKiVS 2011)

The Semantic Shadow: Structuring the Web for Adaptations

Pascal Bihler and Armin B. Cremers

12 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

The Semantic Shadow: Structuring the Web for Adaptations

Pascal Bihler[1] **and Armin B. Cremers**[2]

[1] bihler@cs.uni-bonn.de
[2] abc@cs.uni-bonn.de
Institut für Informatik III
Rheinische Friedrich-Wilhelms Universität Bonn, Germany

Abstract: This paper introduces the concept of the *Semantic Shadow (SemS)*, a model for managing contentual and structural annotations on web page elements and their values. While the model is based on RDF, it supports a contextual weighting of the annotated information, allowing the annotator to specify the annotation values in relation to the evaluation context. This supports dynamic web page adaptation based on structural semantics on the page, whereas a context-awareness of the adaptation can be directly modeled alongside with the structural annotations.

Keywords: Semantic Shadow, RDF, Web Page Adaptation, Request Context

1 Introduction

Today's World Wide Web is a massive collection of information, and at the same time the primary platform for communication and commerce. While in the beginning, the web's user group was compact, nowadays it's extremely divers. People using a certain web based service not only differ by their social backgrounds, by their favors and dislikes. With the advent of small but high-performance mobile devices like netbooks and smartphones, mobile web use is rapidly increasing. Therefore, the situations, in which a web based service is employed are numerous and this usage context defines another dimension in the service design space.

Ideally, a web based service user interface would be optimized for every usage scenario: A teenager ordering at Luigi's Restaurant at midnight on his latest multimedia mobile phone would see another menu than his grandma, using the same restaurant's service at noon from her home TV. The classic way to personalize web information is the development of a dedicated, service-specific adaptation process. Taking into account the mentioned increase in user and device variety, this manual adaptation process results in an exponentially growing workload.

If the web information would be annotated with semantic information, revealing the structural and contextual content of a web page to the processing algorithms, these adaptation processes could be, to a certain extend, automated. Unfortunately, the majority of existing web sites are not annotated in a way supporting automated context adaptation, neither are most of the information and services published today. This paper introduces the concept of the *Semantic Shadow* to annotate web site elements with structural or contentual information in a context-aware way.

In the next section, the associated annotation model is introduced and an RDF based representation is introduced. In Section 3, applications of the *Semantic Shadow* concept are discussed, followed by the presentation of related research work in Section 4. In Section 5, the approach is summarized and further concept extensions and applications are discussed.

2 Annotations for Contextual Semantics

The main language to describe information on the web used on current websites is HTML [RHJ99]. Being a markup language, the goal of the HTML-tags is to describe in a very simple way the semantics of certain parts of a text document. It is up to a displaying application to interpret these semantic markups and to display the content appropriately.

Whereas at the development time of HTML the correct visualization of scientific research texts and, later on, the display of multi-media information on a desktop PC was primarily in focus, the requirements today are by far broader and the web is used in very different ways and on various devices. Especially barrier-free access for impaired users [AT00] and the visualization of web pages on mobile devices with reduced processing power and very limited display capabilities have been in research focus during the last decade [CMZ03]. Furthermore, the increasing mobile use of web resources makes the users aware of the possibilities of context dependent service adaptations, which not only makes sense for location information, but for various other contexts.

To enable automatic adaptation, further semantic information about the web site's content and its structure than given by traditional HTML is required. This information, which can be interpreted as a further annotation to the web site elements, can either be integrated directly into the HTML page code [W3C07] or stored in a parallel structure [HKO+00].

2.1 Annotation Semantics

Regardless of the way web element annotations are stored, two different kinds of annotations can be distinguished:

1. Annotations describing *the content* of the annotated element (i. e. some role, meaning etc.).
2. Annotations describing *the structural semantics* of the annotated element (i. e. grouping, priority in comparison to other elements etc.).

2.1.1 Contentual Annotations

The contentual annotation describes the contents of the marked information. These annotations can be interpreted as the "role" the content plays, like the approach of [W3C07] suggests. In the context of the Semantic Web [HHRS08] initiative, the term "annotation" traditionally refers to these contentual annotations [AIF04]. Since there has been research on this topic through the last decade [Rei06], very different roles of web page elements have been proposed (see e. g. [DCM08, W3C07, Att08]). It is easy to imagine that a specific web site context might trigger individual role descriptions. To associate a web element with a role, *SemS* defines a general annotation type:

- $hasRole(x,R)$: x defines an element (or a group of elements) of the web page, and R is a string with an application-dependent semantics.

For complex semantic relations or the interaction of annotations coming from different application domains, web ontologies [HHRS08] can be used to structure the role semantics and to support automatic reasoning on contentual annotations.

2.1.2 Structural Annotations

Structural Annotations do not make direct statements about the content of the marked subject, but about its function in the page structure or its interaction habits. As this, the semantics are less application specific but more generally directed to the page visualization. A base set is:

- $isMemberOf(x,G)$: The element or group x is member of a semantic group[1] named G.
- $hasPriority(x,P[,G])$: The element x has a relative priority, compared to other elements of the page or a specified group G of P (P being a rational number).
- $receivesKeypresses(x)$: The element x can receive key presses .
- $supportsCharset(x,C)$: The element x receives characters from the given charset C.
- $hasValueLength(x,N)$: The value of element x has a length of N characters.
- $hasAttentionTime(x,T)$: The users attention is focused for T seconds on the element x.
- $followsFocus(x,y)$: The focus of the element x follows the focus of y.
- $hasFocusFollower(x,y)$: The focus of the element x is followed by the focus of y.
- $isSummary(x,Y)$: x is the summary of Y, whereas Y can be a page element or a group.
- $inducedBy(v_x,v_y)$: The element x has the value v_x, if y has the value v_y.
- $induces(v_x,v_y)$: If the element x has the value v_x, the element y has the value v_y.
- $dependsOn(x,v_y)$: The element x only gets a non-default value, if the element y has the non-default value v_y.
- $hasDependent(v_x,y)$: If the element x has the non-default value v_x, the element y also gets a non-default value.

2.2 Annotation Model

As sketched in Figure 1 and described in the last section, every element of a web site can be enhanced with some semantic annotation *(Shadow Annotation) a*. Therefore, every annotation contains at least the element it relates to, named the *Subject s* of the annotation. Also the *Type t* of an annotation is required. Every concrete Shadow Annotation can carry one or several *additional properties* $p_1 \dots p_k$, as required by the semantics connected with the type of the annotation. These additional properties can be simple scalar values (Long, Double, String, Boolean), but they can also referenc other web site elements or their values. To determine the validity range of the annotation, a *Contextual Confidence* Γ can be given, referencing a context object[2] c and a real confidence value $\gamma \in [0..1]$ denoting the probability that the given annotation's statement is valid in the given context. Therefore, a Shadow Annotation is defined as the tuple

$$a = (s,t,p_1 \dots p_k,\Gamma) \quad (1)$$

with the contextual confidence level Γ being $\varnothing \equiv (1.0,\varnothing)$ or

$$\Gamma = (\gamma,c) \quad (2)$$

The database managing all known annotations is called the *Semantic Shadow* and supports querying (and storing) Shadow annotations using an interface depicted in Figure 2a.

[1] A *semantic group* is a named group of HTML elements on the same page.
[2] As the requirements for a "context" definition in the *Semantic Shadow* concept is reduced to the inclusion operation and the referencing to a compatible representation, any model supporting those requirements can be used.

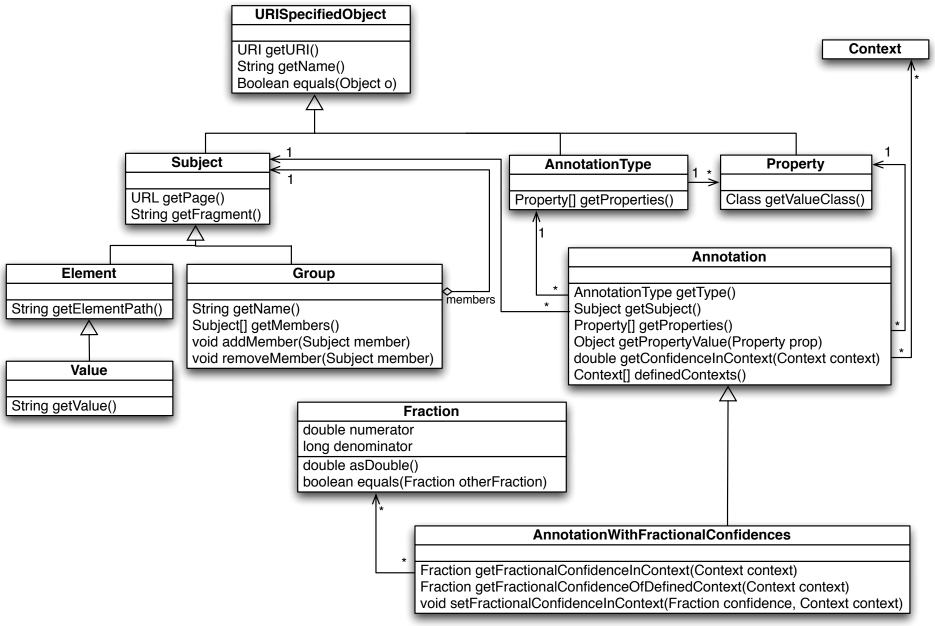


Figure 1: The general model of a Shadow Annotation and associated classes.

2.3 Model Representation

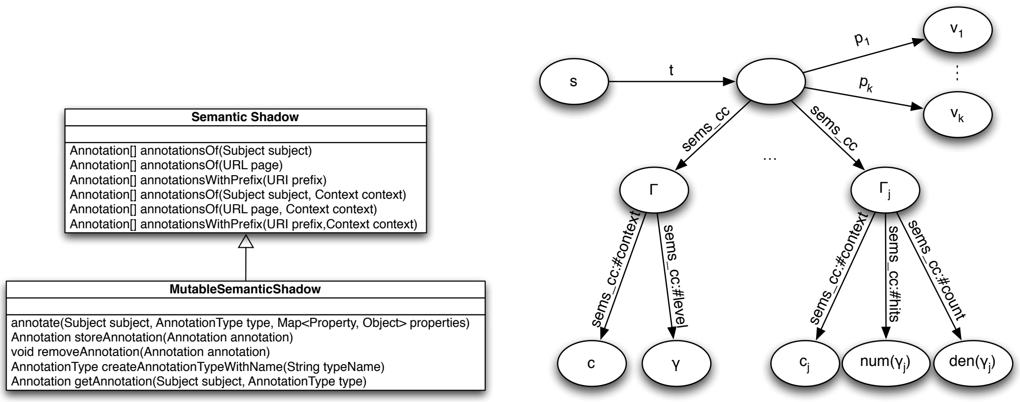
To interact with the annotations, some representation of the model is required. To match with existing Semantic Web technologies, an RDF [KC04] based representation has been developed.

In RDF, every data element consists of the triple *(Subject, Property, Object)*. Whereas the Subject and Property are required to be Resources identified by a Unique Resource Identificator (URI), the Object can either be a single value/resource or another RDF triple. The mapping of an annotation $a = (s, t, p_1 \ldots p_k, \Gamma)$ is realized as follows (see Figure 2b):

- s is mapped to the subject of the RDF-triple (see Section 2.3.1)
- t is mapped to the property of the RDF-triple (see Section 2.3.2)
- As object, an empty node ("Annotation Property Node") is introduced.
- The type specific annotation properties $p_1 \ldots p_k$ are mapped to properties of the Annotation Property Node (see Section 2.3.3)
- The contextual confidence Γ is also modeled as a property of the Annotation Property Node with the URI `sems:contextual_confidence`[3] (abbrev. `sems_cc`, see Section 2.3.4)

If several annotations only differ by their Γ property, they can share the same RDF representation, whereas several `sems_cc` properties are attached to its Annotation Property Node.

[3] The prefix `sems:` is an abbreviation for the URI `http://www.cs.bonn.edu/sems/2010/08/14/`.



(a) Semantic Shadow API

(b) RDF representation of an annotation

Figure 2: An annotation, accessible via the *SemS* API can be directly mapped to RDF-triples.

2.3.1 The Annotation Subject

The annotation subject can either be a web site element, the value of a web site element or a semantic group in a web site. To map to RDF, these resources have to be identified by a URI. In every case this URI is constructed of the related web site URL (including the location part, if relevant), a `#` as separator and a fragment part.

In the case of a web site element, the fragment part identifies the web site element using a simplified and URI-compatible XPath 1.0 [CD99] description:

- If the element contains an id, simply the id can be stated.
- Otherwise, the fragment starts with a `/` followed by the simplest possible path only containing the `child` axes and the functions `position()` and `id()`.
- To shorten the URI length, the following abbreviations are used:

```
/element(n) ≡ /child::element[position()=n]
/element ≡ /child::element[position()=1]
bla ≡ /id('bla')
```

In case of an element value as annotation subject, the value is added in apostrophes (`'`) to the canonical element fragment. A group is always represented by the fragment `#$groupname`.

2.3.2 The Annotation Type

The URI of the annotation type, constructing the property of the annotation RDF-triple, is generally constructed by using the prefix `sems:aytpe/` and the common name of the annotation, e. g. `hasValueLength`. For annotations introduced by independent modules, individual prefixes might be used.

2.3.3 The Annotation Properties

To represent the annotation properties, the blank "Annotation Property Node" is used as RDF subject for triplets representing each a type specific property. Their value is represented by the triplet object, and the triplet property URI is defined as `annotationTypeUri#propertyName`, e. g. `sems:aytpe/hasValueLength#length`.

2.3.4 The Contextual Confidence

The contextual confidence, if applying, is modeled by a blank node connected with the property `sems:contextual_confidence` (`sems_cc`) to the Annotation Property Node. It contains two RDF triples named `sems_cc:#context` pointing to a context description represented as RDF resource and `sems_cc:#level` stating the confidence level in between $[0\dots1]$. As an alternative, the confidence level can be expressed using 2-tuple `sems_cc:#hits` and `sems_cc:#count`. Then, the confidence level can be calculated as the fraction $hits/count$, while at the same time, the fraction's numerator and denominator are maintained for incremental level updates.

2.3.5 Annotation stability

As websites are updated over time, not only the website content, but also the presentation structure and within it the elements in the HTML DOM change. If content or new elements have just been added to the webpage, the XPath pointers are still valid, and therefore annotations can still be correctly associated. In most cases, even if the structure of a web page changes, the element ids are preserved. As the annotation subject identification in the Semantic Shadow concept are based on these ids as soon as they are available, the "stability" of these annotations is high. In some cases, if element ids are changed or not available at all, a re-identification of the original annotation subject in the restructured web page is not possible.

To maintain *Semantic Shadow* annotations despite fundamental page structure changes, a time stamp based transformation document is required. This document needs to be maintained by the site publisher, since correspondences in between different page revisions are hard to detect automatically.

2.4 Interoperability with Semantic Web Tools

In the last section, a way to store *Semantic Shadow* annotations in an RDF model has been presented. While the way of introducing a blank node to store multiple values of an RDF property (here: The Annotation Property Node) is generally supported by Semantic Web Tools, the notion of *contextual confidence* and its representation are specific to the *Semantic Shadow* concept. To process the data, the RDF model needs to be reduced depending on the operations which ought to be executed on the model.

2.4.1 Reduction to fuzzy datasets

A number of approaches exist to introduce fuzzyness into the normally binary world of RDF (see Section 4). Most of these have in common that there is a way to state the probability of a RDF tuple being true. Depending on the output representation, a specific mapping can be defined, which, given a specific evaluation context c transforms the contextual confidences γ_c into the required probability value on the main annotation tuple. This corresponds to an evaluation of the *Semantic Shadow* in a concrete context c.

2.4.2 Reduction to plain RDF

If uncertainty has to be eliminated completely from the RDF model, the reduction has not only to take into account a concrete interpretation context c, but a decision baseline γ_{min} as well. Each annotation persisting in the original model with $\gamma(c) < \gamma_{min}$ will be removed from the model, as well as all `sems_cc`-properties. This corresponds to an evaluation of the *Semantic Shadow* in the context c, while for every annotation a the mapping $\gamma_a(c) \leftarrow \lceil \gamma_a(c) - \gamma_{min} \rceil$ is performed.

3 Applications of the Semantic Shadow Concept

The concept of the *Semantic Shadow* extends the currently established web data specification by another, orthogonal information layer. The information available in form of contextual semantic annotations can be used for different scenarios, of which at least three groups can be distinguished:

1. Live adaptation of HTML data
2. Off-line adaptation of HTML data and generation of target group variants
3. Static analysis of usage information

3.1 Live adaptation

"Live web page adaptation" is characterized by three vertex points:

1. The web page content stored on the web server is generally stored in a way that no specific, access-context dependent adaptations are required to be performed by the serving host. This includes static web pages as well as dynamically generated ones. Site adaptations, which are required by the business logic context (e. g. the visualization of a virtual purchase cart) are evidently handled by the web server software.
2. The client is not required to perform adaptations manually, e. g. by defining user side style sheets or by navigating to a specific URL triggering a predefined adaptation process.
3. The adaptation work is done on demand, i. e. following a request by the user's browser, the request context is determined and using Semantic Shadow information, the requested web data is adapted.

The adaptation process can take place itself on *client side* or using a *proxy software*.

Performing *client side* adaptation takes place on the user's machine, e. g. using a browser plug-in or some kind of network middleware. This approach is favorable if privacy concerns prohibit the processing of context data outside the client's machine, or if the *Semantic Shadow* database is user specific and maintained on the client machine. Nevertheless, web page adaptation might consume quite a bit of computing power and memory, depending on the complexity of the adaptation process. Limited to the client's resources, a dynamic scaling neither in the means of CPU, nor in the means of available memory is possible. In addition, the responsiveness of the complete system and with it of other applications running concurrently might be affected. As a further drawback, the user has to install a specific piece of software on his system, which makes the approach more obtrusive. If the *Semantic Shadow* information has to be requested from a distant server, the querying time adds up to the perceived page loading time and the amount of transmitted data increases. Summarized, client side processing might be an option when used in environment with specialized requirements and on desktop or notebook computers, but does not seem to be an option for today's generation of smart phones.

A *proxy software* hooks into the data flow and works in between the client's browser and the web server software. The technique is also called "third-party adaptation" or "dynamic re-authoring" and is, according to Kurz et al. [KPG04], one of the most widely accepted approaches for dynamic adaptation processes. Kurz names minimal maintenance effort and single-source authoring as reasons, but this goes together with other advantages like the option for transparent setup, minimization of data transferred to the client's device over the "last mile" and externalization of resource usage. In their paper from 2004, Kurz et al. mention nine approaches of adaptation proxies in research studies dating back to 1987, so the approach can be seen as being established. There are several deployment options which rank from a personal proxy installed on the user's system (similar to the approach sketched in the last paragraph) over a transparent proxy system running on a provider's gateway computer, an external service which is provided as a general solution and has to be configured in the user OS/browser's network connection settings up to acting as the web server software, which transparently forwards the request to the actual, hidden web server.

3.2 Off-line variant generation

While the live adaptation outlined in the last section is a very flexible approach, it is at the same time very ineffective regarding CPU and memory resources. At the instance the user requests a web page, the adaptation process has to be triggered and performed without saddling the user with a long waiting time until the requested web page is displayed in his browser. This limits the runtime of the adaptation algorithm and therewith its complexity. In addition, if several users are requesting an adaptation process concurrently, enough resources have to be provided to perform the adaptation calculation in parallel.

One approach to reduce the adaptation resource demands is the pre-calculation of adapted web pages and the persistent storage of those variants. This approach works especially well if the content to be adapted is rather static and the request context can be simplified into a few standard context classes. The definition of these context classes and the mapping of a specific request class to a specific context class as a step of simplification is highly scenario dependent and might require further domain knowledge.

Practically, the off-line variant generation can be combined with the live adaptation process presented in the last section: Upon an incoming client request, the request context is mapped to a variant context class and the variant cache is checked for an appropriate recent page variant. If there is no variant available or the pre-cached variant is outdated, a live adaptation using a representative context description of the required context class as request context is performed.

The result is returned to the client as well as cached on the adaptation server for further use. This combined approach still suffers from complexity limitations implied by the response time limits, but on the other hand minimizes response time for further requests in context mapped to the same context class. In addition, the variant management is simplified and optimized, since no pre-generation of variants practically never requested is done. Using a simple request statistics (or an analysis process as sketched in the next section), proactive pre-generation of the most probably requested variants can be performed while at the same time requests with contexts mapped to uncommon classes can still be served.

3.3 Static analysis

The third category of applications for the *Semantic Shadow* concept presented here does not have a direct impact on the user's page visualization: The static analysis of annotations. An application from this category uses the annotations of the *Semantic Shadow* as input data to infer further knowledge about the web page and its structural semantics. The result of such an algorithm might be the definition of representative context classes to partition the set of possible request contexts (see the previous section). Another result might be a human readable report indicating the typical use of the web site by the clients in varying contexts, so that content- or presentation-optimizations can be defined. A third option is to generate new *Semantic Shadow* annotations, based on further interpretation knowledge embedded into the analysis algorithm (e. g. detecting structural patterns).

4 Related Work

Siegfried Handschuh and Steffen Staab describe in [HS02] a system to annotate existing and newly created web pages with metadata. The described CREAM system focuses on using annotations to describe semantically, what is being represented by the HTML elements (used referred to in this paper by the term "contentual annotations"), i. e. they do not focus on structural annotations. In addition, they do not take into account, that the context of evaluation might change the semantics, thus they do not model any context dimension for their annotations. The same is true for the SHOE language [HHL03], Ontobroker [FDES98] and SemTag [DEG+03].

In their work [HKO+00], Masahiro Hori and his colleagues from IBM sketch a system designed to use external, RDF-based annotation of web page elements to dynamically optimize these web pages for mobile web use. They introduced the referencing of web page elements by using XPath and XPointer expressions in RDF subjects. The annotations are created using a dedicated authoring tool, and using web proxy technology they are interpreted upon mobile request to create a optimized web page. Annotations they envisioned to perform this transformation process are *alternatives*, *splitting hints* and *selection criteria*:

Alternatives are hints for content adaptation and encoding, whereas the appropriate alternative visualization of the content should be chosen based on the client's capabilities.

Splitting Hints are general information about the *grouping of elements*. In their framework, they are used to split up web pages exceeding the client's visualization capabilities.

Selection Criteria are semantic descriptions used in Hori's framework to decide upon the element to display on a mobile device. Besides stating required client capabilities, annotations for describing the elements *role* and its *importance* relatively to other elements of the same page are provided.

The *context*, the framework of Hori et al. is taking into account, is primarily defined by the requesting device's capabilities, to which a requested web page is adapted based on the static annotations. These possible capabilities are reflected in the capability description of the *selection criteria* annotation. Further incorporation of other context information is not specified by Hori. The research found its way into the IBM WebSphere Transcoding Publisher product [Web09].

A more complex approach to model context with technologies from the Semantic Web is described by Strang et al. in [SLF03]: While the authors focus on the use of the presented Context Ontology Language for determining service interoperability, it seems reasonable to use the basic Aspect-Scale-Model from this work combined with adequate rules as context representation for *Semantic Shadow* annotations.

With the definition of a confidential value, the interpretation of the corresponding annotation can be seen as being fuzzy with respect to an evaluation context. Several approaches have been made to extend Semantic Web technologies in a way that allows to encode this fuzziness directly and therefore enable Semantic Web reasoner to handle those fuzzy facts: Fernando Bobillo et al. present in [BS09] a general extension for OWL 2 to represent uncertain information by adding OWL classes to containers to which also "uncertainty" information is added (in the form of an uncertainty type and value), together with a reasoner to extract new information from such ontologies. Giorgos Stoilos et al. include fuzzy knowledge into the Semantic Web by extending SHOIN DL [SSSK06], and the W3C Uncertainty Reasoning for the World Wide Web Incubator Group [W3C08] discuss broadly the question of how to model uncertainty in the context of WWW and the Semantic Web. Thomas Lukasiewicz and Umberto Straccia provide in [LS08] an overview over different approaches to integrate fuzzy Logic into OWL.

Stephan Weibelzahl provides in [Wei02] an extensive overview of the characteristics and evaluation of adaptive software systems. He emphasizes that an adaptive system extends manual configurability by inferring individual user characteristics. An implementation of an adaptive system based on the *Semantic Shadow* concept can perform this "nontrivial inference" by evaluating the contextual confidence values in the given request context.

5 Summary and Further Work

In this paper, the *Semantic Shadow* concept was introduced: A context-aware way to annotate web page elements and values using an XPath based reference scheme. Some examples for contentual and structural annotation types have been given, while a concrete application is able to extend this type set. In addition to earlier web element annotation frameworks, the *SemS* concepts allows to specify context-related confidence values for the annotation, thus bringing the context-awareness of web adaptation into the data model.

Despite presenting the model and its RDF representation, several applications have been discussed, which are based on the *SemS* annotation model: Live adaptation of web sites using a proxy software evaluating the request context, off-line variant generation to reduce the calculation overhead of real-time adaptation, and static analysis of web page structure, which goes beyond the current possibilities of log file and HTML DOM analysis.

While this paper has discussed in detail the fundamental annotation model of the *SemS* concept and its application domains, very little has been said about how these annotations are generated. One option is of course the manual generation of annotations by the page author or a third editor. In the course of the *SemS* research, a GUI-based editor has been developed which supports the author on this task. Another way to generate structural annotations can be the analysis of web usage data: "Looking over the user's shoulder", the way of interacting with a given web page can induce certain implicit structures, which can then be made explicit by generating the respective annotations. Further research on this topic is in progress.

Bibliography

[AIF04] AIFB. Annotation Portal. 2004.
http://annotation.semanticweb.org/

[AT00] C. Asakawa, H. Takagi. Annotation-Based Transcoding for Nonvisual Web Access. In *Assets '00: Proceedings of the fourth international ACM conference on Assistive technologies*. Pp. 172–179. ACM, New York, NY, USA, 2000.
doi:10.1145/354324.354588

[Att08] R. Atterer. *Usability Tool Support for Model-Based Web Development*. PhD thesis, Fac. of Mathematics, Computer Science and Stat., University of Munich, 2008.

[BS09] F. Bobillo, U. Straccia. An OWL Ontology for Fuzzy OWL 2. In *ISMIS '09: Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*. Pp. 151–160. Springer-Verlag, Berlin, Heidelberg, 2009.
doi:10.1007/978-3-642-04125-9_18

[CD99] J. Clark, S. DeRose. XML Path Language (XPath). November 1999.
http://www.w3.org/TR/1999/REC-xpath-19991116

[CMZ03] Y. Chen, W.-Y. Ma, H.-J. Zhang. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*. Pp. 225–233. ACM, New York, NY, USA, 2003.
doi:10.1145/775152.775184

[DCM08] DCMI. Dublin Core Metadata Element Set, Version 1.1. January 2008.
http://dublincore.org/documents/2008/01/14/dces/

[DEG+03] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, J. Y. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In *WWW '03: Proc. of the 12th int. conf. on World Wide Web*. Pp. 178–186. ACM, New York, NY, USA, 2003.
doi:10.1145/775152.775178

[FDES98] D. Fensel, S. Decker, M. Erdmann, R. Studer. Ontobroker: Or How to Enable Intelligent Access to the WWW. In *11th Workshop on Knowledge Acquisition, Modeling, and Management (KAW '98)*. AAAI Press, 1998.

[HHL03] J. Heflin, J. A. Hendler, S. Luke. SHOE: A Blueprint for the Semantic Web. In *Spinning the Semantic Web*. Pp. 29–63. 2003.

[HHRS08] P. Hitzler, M. Hrtzsch, S. Rudolph, Y. Sure. *Semantic Web*. Springer Akademischer Verlag, 2008.

[HKO+00] M. Hori, G. Kondoh, K. Ono, S. ichi Hirose, S. Singhal. Annotation-Based Web Content Transcoding. *Computer Networks* 33(1-6):197 – 211, 2000. doi:10.1016/S1389-1286(00)00068-2

[HS02] S. Handschuh, S. Staab. Authoring and Annotation of Web Pages in CREAM. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*. Pp. 462–473. ACM, New York, NY, USA, 2002. doi:10.1145/511446.511506

[KC04] G. Klyne, J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. February 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[KPG04] B. Kurz, I. Popescu, S. Gallacher. FACADE – A FrAmework for Context-aware content Adaptation and DElivery. In *Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04)*. 2004.

[LS08] T. Lukasiewicz, U. Straccia. Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(4):291 – 308, 2008. doi:10.1016/j.websem.2008.04.001

[Rei06] G. Reif. *Semantic Web - Wege zur vernetzten Wissensgesellschaft*. Chapter Semantische Annotation, pp. 405–418. Springer Berlin Heidelberg, 2006. doi:10.1007/3-540-29325-6

[RHJ99] D. Raggett, A. L. Hors, I. Jacobs. HTML 4.01 Specification. W3C recommendation, W3C, December 1999. http://www.w3.org/TR/1999/REC-html401-19991224/

[SLF03] T. Strang, C. Linnhoff-Popien, K. Frank. CoOL: A Context Ontology Language to enable Contextual Interoperability. In Stefania et al. (eds.), *DAIS 2003, LNCS 2893*. Pp. 236–247. 2003.

[SSSK06] G. Stoilos, N. Simou, G. Stamou, S. Kollias. Uncertainty and the Semantic Web. *IEEE Intelligent Systems* 21:84–87, 2006. doi:10.1109/MIS.2006.105

[W3C07] W3C - WorldWideWeb Commitee. XHTML Role Attribute Module, A Module to Support Role Classification of Elements. October 2007. http://www.w3.org/TR/2007/WD-xhtml-role-20071004/

[W3C08] W3C Incubator Group on Uncertainty Reasoning for the World Wide Web. Final Report. Technical report, W3C, 2008 2008. http://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/

[Web09] W. Web/Durham/IBM. WebSphere Transcoding Publisher Version 4.0. May 2009. http://www-01.ibm.com/software/pervasive/transcoding_publisher/

[Wei02] S. Weibelzahl. *Evaluation of Adaptive Systems*. PhD thesis, Pedagogical University Freiburg, October 2002.