Web Semantics: Science, Services and Agents on the World Wide Web 22 (2013) 1–18

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Review article

User interfaces for semantic authoring of textual content: A systematic literature review

Ali Khalili *, Sören Auer

AKSW Research Group, Institute of Computer Science, University of Leipzig, Augustusplatz 10, 04109 Leipzig, Germany

ARTICLE INFO

Article history:
Received 17 December 2011
Received in revised form
17 June 2013
Accepted 22 August 2013
Available online 11 September 2013

Keywords:
Semantic Web
User interface
Content authoring
Applications

ABSTRACT

Practical approaches for managing and supporting the life-cycle of semantic content on the Web of Data have recently made quite some progress. In particular in the area of the user-friendly manual and semi-automatic creation of rich semantic content we have observed recently a large number of approaches and systems being described in the literature. With this survey we aim to provide an overview on the rapidly emerging field of Semantic Content Authoring (SCA). We conducted a systematic literature review comprising a thorough analysis of 31 primary studies out of 175 initially retrieved papers addressing the semantic authoring of textual content. We obtained a comprehensive set of quality attributes for SCA systems together with corresponding user interface features suggested for their realization. The quality attributes include aspects such as usability, automation, generalizability, collaboration, customizability and evolvability. The primary studies were surveyed in the light of these quality attributes and we performed a thorough analysis of four SCA systems. The proposed quality attributes and UI features facilitate the evaluation of existing approaches and the development of novel more effective and intuitive semantic authoring interfaces.

Contents

* Corresponding author. Tel.: +49 17639002240; fax: +49 341 97 32310.
E-mail addresses: khalili@informatik.uni-leipzig.de (A. Khalili), auer@informatik.uni-leipzig.de (S. Auer).

1570-8268/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.websem.2013.08.004

1. Introduction

Practical approaches for managing and supporting the life-cycle of semantic content on the Web of Data have recently made quite some progress. The life-cycle of semantic content in particular comprises phases such as semantic content creation, publishing, quality assessment, annotation, enrichment, revision and archiving [1]. On the backend side, a variety of triple stores were developed and their performance and maturity improves steadily (cf. recent triple store benchmarking efforts such as the *DBpedia benchmark* [2]). Similarly tools and algorithms for linking data and schemata are progressing and approaches are deployed for the use on the emerging Web of Data [3]. The quantity of semantic content being made available on the Data Web is rapidly increasing, mainly due to the use of automated knowledge extraction techniques or due to the semantic enrichment and transformation of existing structured data (cf. *LODStats* [4]). Despite many interesting showcases (e.g. *Sindice*,[1] *Parallax*[2] or *PowerAqua*[3]), we still lack more user friendly and scalable approaches for the exploration, browsing and search of semantic content. However, the currently *least* developed aspect of the semantic content life-cycle is from our point of view the user-friendly *manual and semi-automatic* creation of rich semantic content. Results of surveys reported by Heitmann et al. 2009 [5], Paulheim et al. 2010 [6], and Hachey 2011 [7] support this fact as well.

We define *semantic content authoring* as the tool-supported manual composition process aiming at the creation of documents which are:

- fully semantic in the sense that their original data model uses a semantic knowledge representation formalism (such as RDF,[4] RDF-Schema[5] or OWL[6]) or
- based on a non-semantic representation form (e.g. text or hypertext), which is enriched with semantic representations during the authoring process.

A semantic authoring UI is a human accessible interface with capabilities for writing and modifying semantic documents. Semantic documents facilitate a number of important aspects of information management:

- For *search and retrieval* enriching documents with semantic representations helps to create more efficient and effective search interfaces, such as faceted search [8] or question answering [9].
- In *information presentation* semantically enriched documents can be used to create more sophisticated ways of flexibly visualizing information, such as by means of semantic overlays as described in [10].
- For *information integration* semantically enriched documents can be used to provide unified views on heterogeneous data stored in different applications by creating composite applications such as semantic mashups [11].
- To realize *personalization*, semantic documents provide customized and context-specific information which better fits user needs and will result in delivering customized applications such as personalized semantic portals [12].
- For *reusability* and *interoperability* enriching documents with semantic representations (e.g. using the SKOS[7] and Dublin Core vocabularies[8]) facilitates exchanging content between disparate systems and enables building applications such as executable papers [13].

There are already many approaches and tools available for semantic content authoring which address different aspects of this task by proposing appropriate user interfaces. Due to the wealth of different approaches emerging it is crucial to obtain an overview on the advancement in this dynamic field. Furthermore, having a holistic view on approaches and tools provides us with an exhaustive set of quality attributes, which are important for conceiving guidelines for developing more effective and intuitive semantic authoring interfaces.

In this article, we summarize the findings of a systematic literature review on semantic content authoring. We extract different types and properties of user interfaces proposed for semantic authoring of textual content. The results reveal a set of quality attributes which can be used for classification of semantic authoring tools. Furthermore, we report on the suggested user interface types and features proposed in the literature to realize these quality attributes.

The rest of the paper is organized as follows. In Section 2 we describe the research method and the review protocol used for conducting the systematic review. In Section 3 we define the terminology of the paper then we elaborate on the results of the review by surveying the extracted quality attributes in Section 5. In Section 9 we discuss four existing semantic authoring tools and describe them in the light of the quality attributes. In Section 10 we report on the gaps and open research issues revealed from the results of our systematic literature review. Finally in Section 11 we conclude and present some ideas for future work.

1 http://sindice.com/.
2 http://www.freebase.com/labs/parallax/.
3 http://technologies.kmi.open.ac.uk/poweraqua/.
4 Resource Description Framework: http://www.w3.org/RDF/.
5 http://www.w3.org/TR/rdf-schema/.
6 Web Ontology Language: http://www.w3.org/OWL/.
7 Simple Knowledge Organization System: http://www.w3.org/2004/02/skos/.
8 http://dublincore.org/.

2. Research method

We followed a formal systematic literature review process for this study based on the guidelines proposed in [14,15]. A systematic literature review is an evidence-based approach to thoroughly search studies relevant to some predefined research questions and critically select, appraise, and synthesize findings for answering the research questions at hand. Systematic reviews maximize the chance to retrieve complete datasets and minimize the chance of bias. As part of the review process, we developed a protocol (described in the sequel) that provides a plan for the review in terms of the method to be followed, including the research questions and the data to be extracted.

2.1. Research questions

The goal of our survey is analyzing existing user interfaces for semantic content authoring and thereby providing a set of quality attributes, which can serve as guidelines for designing suitable and effective user interfaces for semantic content authoring. To achieve this goal we aim to answer the following general research question:

What are existing approaches for user-friendly semantic content authoring?

We can divide this general research question into the following more concrete sub-questions:

- RQ1. *How to classify existing approaches for semantic content authoring?*
- RQ2. *What type of user interfaces are used by each approach?*
- RQ3. *What are the features supported by the proposed user interfaces?*
- RQ4. *What type of users are targeted in each approach?*
- RQ5. *How is the user interface evaluated?*

After doing some pilot searches and consulting experts in the field, we obtained a list of pilot studies which served as a basis for the systematic review.

2.2. Search strategy

To cover all the relevant publications, we used the following electronic libraries:

- ACM Digital Library
- IEEE Xplore Digital Library
- ScienceDirect
- SpringerLink
- ISI Web of Sciences

Based on the research questions and pilot studies, we found the following basic terms to be most appropriate for the systematic review:

1. *semantic* OR *linked data* OR *web of data* OR *data web*
2. *content* OR *web page* OR *document*
3. *authoring* OR *annotating* OR *annotation* OR *annotate* OR *enrich* OR *edit*.

To construct the search string, all these search terms were combined using Boolean "AND" as follows:

1 AND 2 AND 3.

The next decision was to find the suitable field (i.e. title, abstract and full-text) to apply the search string on. In our experience, searching in the 'title' alone does not always provide us with all relevant publications. Thus, 'abstract' or 'full-text' of publications should potentially be included. On the other hand, since the search on the full-text of studies results in many irrelevant publications, we chose to apply the search query additionally on the 'abstract' of the studies. This means a study is selected as a candidate study if its title or abstract contains the keywords defined in the search string. In addition, we limited our search to the publications that are written in English and are published after 2002 (when the first ISWC conference was held).

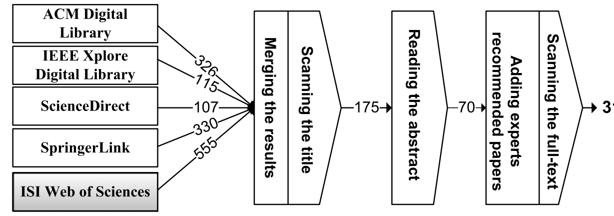

Fig. 1. Steps followed to scope the search results.

2.3. Study selection

Some of the studies might contain the keywords used in the search string but might still be irrelevant for our research questions. Therefore, a study selection has to be performed to include only studies that contain useful information for answering the research question.

Peer-reviewed articles that satisfy all the following inclusion criteria are selected as primary studies:

- I1. A study that focuses on semantic content authoring.
- I2. A study that either proposes a user interface or a set of user interface features for the purpose of semantic content authoring.

Studies that met any of the following criteria were excluded from the review:

- E1. A study that does not focus on semantic content authoring but only mentions the term e.g. as an example or use case.
- E2. A study that does not propose any user interface or user interface feature for semantic content authoring but only a generic, non-user interface supported method, approach or algorithm for semantic annotation.
- E3. A study that is not about Web-based content authoring (e.g. studies about semantic authoring in word processors like LaTeX).
- E4. A study that is only about the ontology creation or ontology annotation (e.g. using natural language).
- E5. A study that does not discuss textual Web content authoring but other modalities such as image, audio or video annotation.

We conducted our review in early July 2011. As a consequence, our review included studies that were published and/or indexed before that date. As shown in Fig. 1, we first applied the search query on each data source separately. Subsequently, to remove duplicate studies, we merged the results obtained from the different data sources. To remove irrelevant studies, we scanned the articles by title and thereby reduced the number of studies to 175. Then, we read the abstract of each publication carefully and further decreased the number of studies to 70. Finally, we added a list of additional papers recommended by experts and then scanned the full-text of the publications. We checked the full-text of studies to see if they fit with our predefined selection criteria. The result comprised 31 publications that represented our final set of *primary studies*.

2.4. Data extraction and analysis

The bibliographic metadata about each primary study were recorded using the bibliography management platform *JabRef*.[9] In addition, we extracted the following information from each paper:

- The used approach for semantic content authoring.
- The type of user interface.

[9] http://jabref.sourceforge.net/.

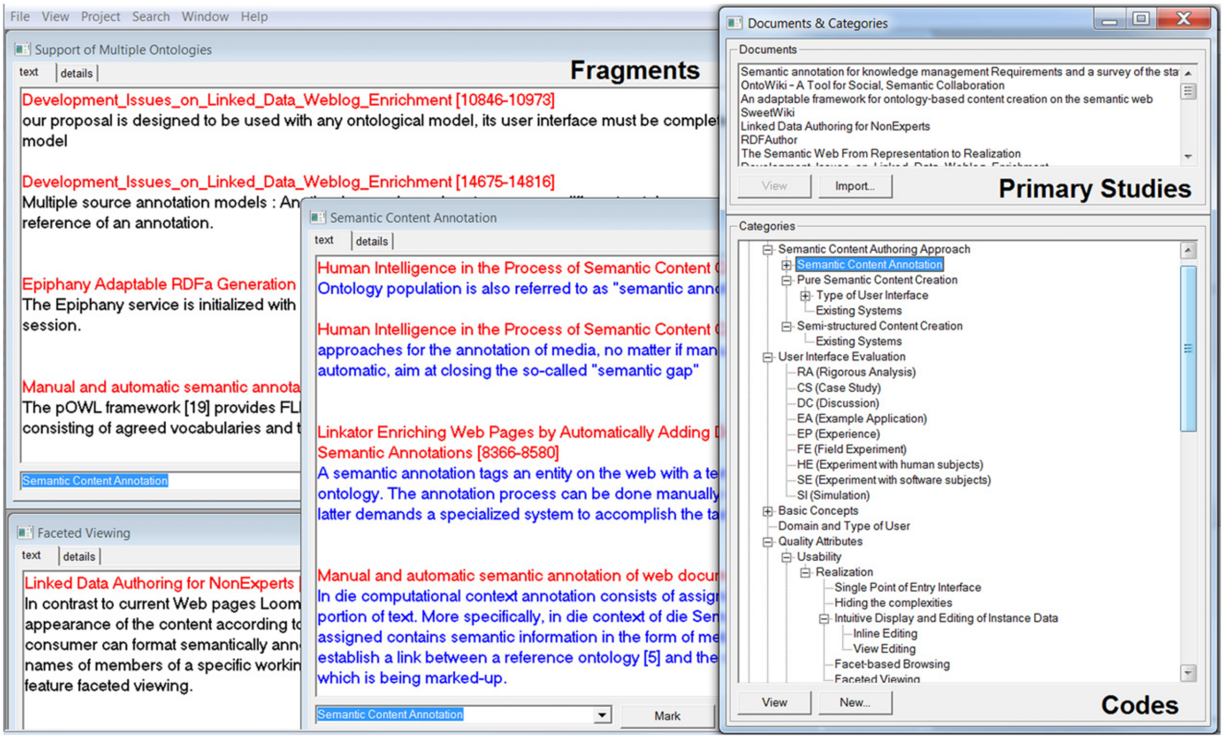

Fig. 2. The screenshot of the coding software showing the generated list of codes from the primary studies.

- The features supported by the user interface.
- The domain and type of user.
- The evaluation method used in the paper.

To analyze the information appropriately, we required a suitable qualitative data analysis method applicable to our dataset. A common method that is used for this purpose is the *grounded theory method* because the theories (the SCA approaches and UI features) are "grounded" in the data [16].

The *Constant Comparison* method, one of the grounded theory techniques, has been often used in analyzing data and generating categories of data. Although the Constant Comparison method can be used on any set of data, it is particularly suitable for data that is context sensitive [17] (i.e. data can be interpreted differently in different contexts). To interpret SCA approaches and UI features correctly, one often needs to understand in which context the approach and feature is proposed and how it is addressed. For instance, consider one study that mentions "evolvability" as a feature for UI. Without understanding the context of this feature, we cannot conclude whether this feature is about designing evolvable UIs or about supporting annotation/ontology evolvability in the UI (which is our aim here).

Miles and Huberman [18] described *coding* as a procedure for the constant comparison method. Codes are tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study. Codes are efficient data-labeling and data-retrieval devices. One method of creating codes which is employed in our review is creating a provisional "start-list" of codes prior to fieldwork. We created this list from our research questions and the pilot studies. To carry out the analysis systematically, we used the following coding procedures proposed by Lincoln–Guba [18]:

- *Filling-in*: we read each study carefully and added the codes for related fragments and items. As new insights or new ways of looking at the data emerged, we reconstructed our coherent coding schema.
- *Extension*: if needed, we returned to materials coded earlier and interrogated them in a new way, with a new theme, construct, or relationship.
- *Bridging*: if new or previously not understood relationships within units of a given category were found, we recorded that relationship.
- *Surfacing*: we identified new categories which contained the previously created codes.

As shown in Fig. 2, we used the *Weft QDA* software[10] to record the codes. The final list of codes are available online.[11]

2.5. Overview of included studies

For quantitative analysis purposes, we performed some queries on the collected database of primary studies. The distribution of studies per year as shown in Fig. 3 indicates an increasing intensity of research in the area of semantic content authoring. The remarkable rise after 2008 can be explained with the emergence and adoption of weak semantic techniques (the so-called 'lowercase' Semantic Web), such as the use of Microformats,[12] RDFa[13] and Microdata.[14] These techniques facilitate semantic content authoring by embedding semantic annotations into the HTML Web pages.

The primary studies included 14 conference papers, 11 journal articles, 4 workshop papers, one thesis and one technical report. Among them, the following four studies are survey papers. Uren et al. [19] have reported a comprehensive review of the studies and applications for semantic annotations which were published before 2006. In [5], Heitmann et al. conducted an empirical

10 http://www.pressure.to/qda/.
11 http://rdface.aksw.org/SLR/codes.qdp.
12 http://microformats.org/.
13 http://www.w3.org/TR/rdfa-syntax/.
14 http://www.w3.org/TR/microdata/.

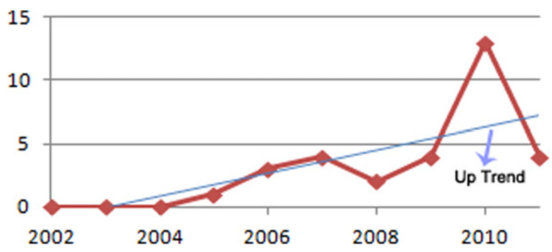

Fig. 3. Publications per year.

survey of Semantic Web applications and have reviewed the challenges of them. Paulheim and Probst [6] surveyed the ontology-enhanced user interfaces and have introduced a schema for characterizing the requirements of ontology-enhanced user interfaces. In [7], Hachey and Gaševic provide an overview of the current progress and gaps in the area of Semantic Web user interfaces in general. Compared to these surveys, the focus and the coverage range of our survey is different. The goal of our survey is to perform a *systematic analysis* of the existing collection of research material addressing the *user interface* aspects of *Web-based* semantic authoring systems. We focus on *semantic authoring of textual content* and cover the literature published between 2002 and 2011.

3. Terminology

The terminology basis of this article is depicted in Fig. 4. In the sequel we describe the individual concepts in more detail:

i. *Semantic gap* is a term coined to describe the discrepancy between low-level technical features of multimedia, which can be automatically processed to a great extent, and the high-level, meaning-bearing features a user is typically interested in [20]. As discussed in [21], semantic gaps in the process of constructing and managing digital content can be divided into three types namely *human-to-machine*, *machine-to-machine*, and *machine-to-human*. In this article we mainly focus on the machine-to-machine semantic gaps which are important when searching or reusing content by machines. In this context, semantics consists of concepts and their logical relationships in an explicit form. When the semantics is processed by a machine, the lack of a common vocabulary may lead to alterations in the original semantics thus resulting in semantic gaps.

ii. *Semantic computing* is a research field that addresses the extraction and processing of the semantics of digital content and naturally expressed user intentions to help retrieve, manage, manipulate, or even create the content. Semantic computing aims to bridge the semantic gap by employing appropriate semantic analysis techniques such as *natural language processing*, *processing of multi-modal content*, *speech recognition*, *Web, data and process mining*, *semantic link discovery* as well as *semantic enrichment and repair*. Semantic Web knowledge representation techniques (e.g. *OWL*, *RDF*, *RDFa*, *SPARQL*, *SKOS*) help to bridge the semantic gap through a common ground of shared vocabularies and ontologies [22,23].

iii. *Semantic document* is an intelligent document (with explicit semantic structure) which "knows about" its own content so that it can be automatically processed in unforeseen ways. These benefits, however, come at the cost of increased authoring effort [23,19].

iv. *Semantic Content Authoring (SCA)* is a tool-supported manual composition process aiming at the creation of semantic documents. With an ontology and a user interface appropriate for the type of content, semantic authoring can be easier than traditional composition of content and the resulting content can be of higher quality [23].

v. Semantic Authoring User Interface (SAUI) is a human accessible interface with capabilities for modifying and writing semantic documents.

vi. Human Computer Interaction (HCI) is a research field that aims to improve the interactions between users and computers by making computers more usable and receptive to the user's needs.

vii. Social Semantic Web is a very general research field triggered by the advent of Web 2.0. It aims at bringing a social novelty, rather than a technical one by providing user-friendly tools to facilitate broad user participation in the process of creating semantic content [20]. The Social Semantic Web vision comprises many of the aforementioned domains and techniques.

4. Semantic authoring approaches

There are already different approaches proposed for semi-structured but non-semantic content authoring (e.g. [24]). These approaches aim at immediate user gratification in the form of useful visualizations and interesting data aggregation but do not focus on using shared vocabularies and formal ontologies which ultimately facilitate portability and reuse. With regard to explicit semantic content authoring recent approaches can be roughly classified into the categories *top-down* and *bottom-up*. As demonstrated in Fig. 5, the classification is based on the starting point of

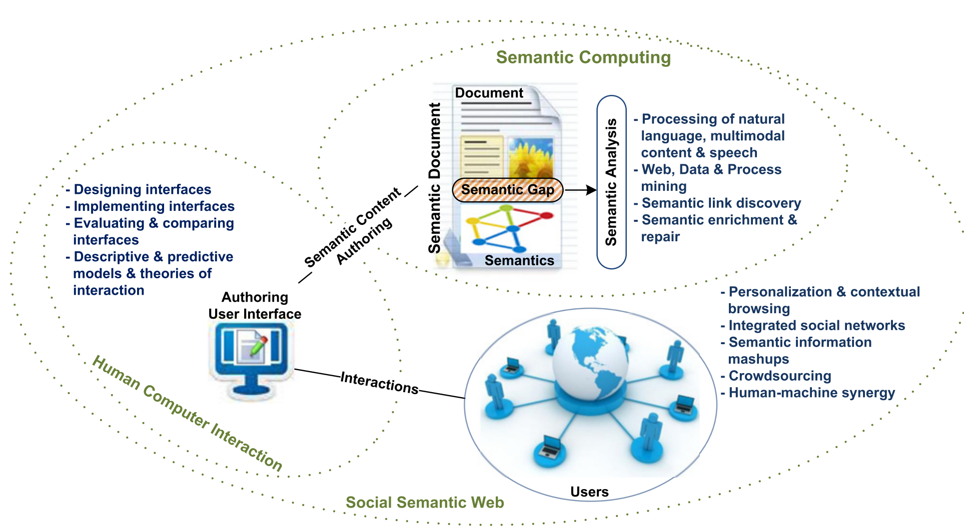

Fig. 4. Semantic content authoring ecosystem.

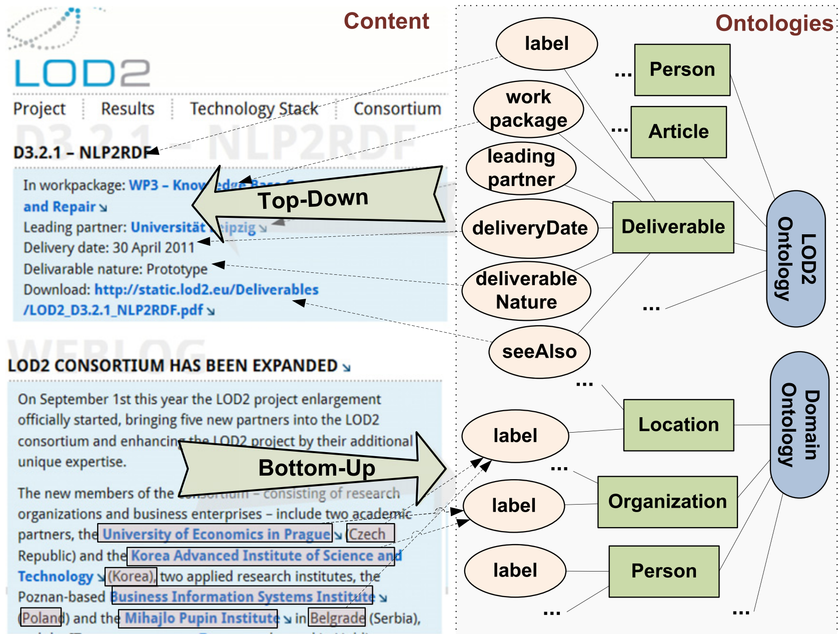

Fig. 5. Top-down and bottom-up approaches for semantic content authoring.

the authoring process which can be ontologies (with upper level of expressiveness) or unstructured content (with lower level of expressiveness). A third category of approaches (called *Middle-Out* or *Hybrid*) that balance between the top-down and the bottom-up approach can also be considered, but is beyond the scope of this paper.

4.1. Bottom-up approaches

These approaches which are usually called semantic annotation techniques (a.k.a. semantic markup [25]) aim to annotate existing documents using a set of predefined ontologies. The basic ingredients of a semantic annotation system are ontologies, the documents and the annotations that link ontologies to documents [19]. Here, we need two kinds of ontologies [26]: *Annotation ontologies* (i.e. metadata schemata) which define what kind of properties and value types should be used for describing a resource. For example, the Dublin Core schema uses elements such as `dc:title`, `dc:creator`, `dc:subject`, etc. *Domain ontologies* which are used to define vocabularies providing possible values for metadata properties. Examples are *eClassOWL*[15] defining products and services, *MeSH*[16] defining medical subjects or the *DBpedia*[17] knowledge base, which is a cross-domain ontology extracted from *Wikipedia*.

The result of the annotation process is a document that is marked-up semantically. For that concern, some markup strategies are already proposed:

- *Microformats*[18] is an approach to integrate semantic markup into XHTML and HTML documents. Microformats re-purpose existing markup definitions (particularly the HTML `class` attribute) in a non-standard way to convey (meta-) data. This approach is limited to a set of few published Microformat templates and thus not easily extensible for domain-specific applications. Moreover, it is not possible to validate Microformat annotations since no proper grammar is used for their definition.
- *eRDF*[19] (embedded RDF) is similar to Microformats but annotates HTML using RDF. However, it faces the same criticism as Microformats, since it uses the same non-standard compatible annotating strategy [25].
- *RDFa*[20] (Resource Description Framework in attributes) is a W3C Recommendation that adds a set of attribute level extensions to XHTML for embedding RDF metadata within web documents. RDFa fulfills the principles of interoperable metadata such as publisher independence, data reuse, self containment, schema modularity and evolvability to a good extent.
- *Microdata*[21] is an HTML5 specification used to nest semantics within existing content on Web pages. It is already in use by popular search engines for interpreting the information contained in a Web page. Microdata is complemented by *Schema.org*[22]—a repository of annotation schemata.

There are normally two types of metadata applied to a document in the process of semantic annotation:

- *Content metadata* describe specific things the author of the document wishes to write about (e.g. people, cities, etc.). These content-related metadata cover a broad domain of information [27]. Natural Language Processing (NLP) annotation APIs (e.g. *DBpedia Spotlight*[23]) are one approach to automatically add content metadata into a document.
- *Context metadata* refers to the general topic, structure or temporal aspects of a document (e.g. title, theme or creation date of a document). These context-related metadata cover a very specific domain of information. Semantic Tagging (e.g. *Faviki*[24]) and structured templates [28] are two approaches to automatically embed context-related metadata in a document.

15 http://www.heppnetz.de/projects/eclassowl/.
16 http://www.ncbi.nlm.nih.gov/mesh.
17 http://dbpedia.org/.
18 http://microformats.org.
19 https://github.com/iand/erdf.
20 http://www.w3.org/TR/rdfa-syntax/.
21 http://www.w3.org/TR/microdata/.
22 http://schema.org.
23 http://spotlight.dbpedia.org.
24 http://www.faviki.com.

Table 1
List of quality attributes together with their corresponding UI features suggested for SCA systems.

Quality attribute	Realization
Usability	Single point of entry interface [32,29,19], faceted browsing [29,33], faceted viewing [34–36], inline editing and view editing [29,37]
Customizability	Living UIs [38], providing different semantic views [29,30,13,39]
Generalizability	Supporting multiple ontologies [19,40,35,41–43], supporting ontology modification [19,26,30], supporting heterogeneous document/content formats [19,35]
Collaboration	Access control [19,43,37], support of standard formats [19,26,40,35,34,43,37,44], UIs for social collaboration [29,39]
Portability	Cross-browser compatibility [26,41], UIs for mobile devices [45]
Accessibility	Accessible UIs [7]
Proactivity	Resource suggestion [34,40], real-time semantic tagging [35,36], concept reuse [35,36,29,40], real-time validation [26,30]
Automation	Automatic annotation [20,25,44,46,26,40,34]
Evolvability	Resource consistency [35], document and annotation consistency [19], versioning and change tracking [29]
Interoperability	Support of standard formats [19,26,40,35,34,43,37,44], semantic syndication [29]
Scalability	Support of caching [29,47], suitable storage strategies [29,19,41,34]

4.2. Top-down approaches

These approaches which are also called *Ontology Population* [20] techniques aim to create semantic content based on a set of initial ontologies which are extended during the population process. When compared to the bottom-up approaches, these approaches deal with semantic representations from the beginning instead of lifting unstructured content to a semantic level.

These approaches combine ontological rigor with flexible user interface constructs to create semantic content. Semantic templates as discussed in [29–31] are one technique to realize this goal. In this approach each class of the ontology has an associated template. Each instance of a class is represented by a page using that template. Data properties are displayed as simple text while object properties are displayed as links to other pages (representing other instances of the ontology). Users can also edit the underlying ontology which will result in changes of the corresponding templates.

5. Quality attributes

In order to evaluate the strengths and weaknesses of different SCA systems, we assess the systems according to predefined criteria which we call *Quality Attributes* in this paper. Quality attributes are non-functional requirements used to evaluate the performance of a system. They are widely used in architecture development and assessment as high level characteristics which systems enclose. In the context of this paper, quality attributes represent the areas of concern regarding the development of an SCA system from the viewpoint of its consumers.

Based on the qualitative analysis of our primary studies, we obtained 11 quality attributes. For each quality attribute we extracted one or more UI feature(s). Features describe a specific type or property of UI that can be used to realize an intended quality attribute. The features are directly (e.g. faceted browsing) or indirectly (e.g. UIs for mobile devices) addressing the required UI functionality for an SCA system. Table 1 surveys the quality attributes and various UI approaches for their implementation. In the sequel we describe each of the 11 quality attributes in more detail.

5.1. Usability

Usability is a measure of the quality of a user's experience in interacting with a system. In ISO 9241 usability is defined as the *effectiveness*, *efficiency* and *satisfaction* with which specified users achieve specified goals in particular environments. Lauesen [48] and Nielsen [49] add more factors such as *learnability* and *utility* to usability definition. In the context of this paper we consider the following factors for defining the usability:

(A) *Efficiency*. How efficient is the system for the frequent user to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use?
(B) *Effectiveness*. How effective is the system to achieve specified tasks with accuracy and completeness?
(C) *Satisfaction*. How satisfied is the user with the system?
(D) *Learnability*. How easy is the system to learn for various groups of users?
(E) *Utility* (or *Usefulness*). Assesses whether the system enables users to solve real problems in an acceptable way.

Simplicity is the main prerequisite of usability. An SCA system should, as a rule, hide technical concepts related to markup languages and ontologies from the non-expert end-users [26,37]. It is crucial to provide end-users with easy to use interfaces that simplify the annotation process and place it in the context of their everyday work. More attention needs to be paid to decrease or blur the gap between the normal authoring process and the semantic authoring process. SCA systems should focus on the user's main task [35]. Usually, a user wants to perform the task of writing some text and not to annotate content. Integrating semantic authoring process into the commonly used packages is one approach to encourage users to view semantic authoring as part of the authoring process not as an afterthought process [19].

The following features of UIs are proposed for improving the usability of SCA systems:

- *Single point of entry interface*.
  It means the environment in which users annotate documents should be integrated with the one in which they create, read, share and edit them. So, there is no added user effort involved in creating a semantic content versus a conventional approach, because the real work is done by the software through capturing semantics that is already being provided by the user [32,29,19]. This will minimize user actions as well as memory load thereby increasing the efficiency, user satisfaction, learnability and utility of the system.
- *Faceted browsing*.
  Faceted browsing is a technique for accessing a collection of information represented using a faceted classification, allowing users to explore by filtering the available information. In the UI which implements this technique, all property values (i.e. facets) of a set of selected instances are analyzed. If for a certain property the instances have only a limited set of values, those values are offered to further restrict the instance selection. Hence, this way of navigation through data will never lead to empty results [29,33]. This feature is useful when searching for available resources or vocabularies. Faceted browsing increases the efficiency and effectiveness of the system by improving the navigability.
- *Faceted viewing*.
  Faceted viewing [34,35] also known as augmented browsing [36] is very similar to faceted browsing but is used to distinguish the semantically annotated content from the normal

content based on the different facets selected by user. For example, highlighting the names of members of a specific working group with a yellow background in the text. Similar to the faceted browsing, faceted viewing will increase the efficiency and effectiveness of the system by improving the navigability.

- *Inline editing and view editing.*
  An SCA system should provide different editing modes for editing single and batch items. Inline editing allows editing items by clicking on them. View editing supports the editing of a combination of items in a specific view in one single step [29,37]. This feature helps users to edit items in a minimum number of steps (minimal action) thereby increasing the efficiency and user satisfaction.

Table 2 shows how the aforementioned UI type and properties affect our previously defined usability factors. It is based on the *QUIM* model defined by Seffah et al. [50]. Quality in Use Integrated Measurement (QUIM) model brings together usability factors, criteria, metrics, and data mentioned in various standards or models for software quality and defines them and their relations with one another in a consistent way.

5.2. Customizability

Customizability is the ability of a system to be configured according to users' needs and preferences. Instead of being a static form strictly dependent on a given schema, an SCA system should provide mechanism to tailor its functionalities based on the user needs [30]. In [36] the concept of "semantics in the eyes of the end-user" is introduced which means an SCA system should provide different views for different personas using the system.

The following features of UIs are proposed for improving the customizability of SCA systems:

- *Living UIs.*
  A Living UI is a user interface that configures itself to automatically display the information most relevant to the user, dynamically adjusts to changing data, and still allows single users to customize according to their preferences [38]. End-user development techniques like *Programming by Example* (PbE) allow inferring user intents in real interactions and according to that providing customized outputs [51].
- *Providing different semantic views.*
  Semantic views allow the generation of different views on the same metadata schema and aggregations of the knowledge base based on the roles, personal preferences, and local policies of the intended users [29,30,13]. Such views can be either generic or domain specific. Generic views provide visual representations of instance data according to certain property values (e.g. map view or calendar view). Domain specific views address the requirements of a particular domain user (e.g. chemists need specific views for visualizing the atomic structure of chemical compounds).

5.3. Generalizability

Generalizability is the ability of a system to adapt to different situations or use cases. An SCA system should support a wide range of metadata schemata in a flexible way. In fact, the more flexible and adaptable a system is, the more valuable it is for different contexts and users. A generic SCA system reduces the costs of supporting new schemata considerably, by following the evolution of existing standards and integrating heterogeneous resources [30]. *Adaptivity* is an important capability of a generic system. An SCA system should be adaptable to different annotation and authoring uses with different kinds of contents to be processed [26,42]. In most of the cases generalizability is in opposition to *Usability* of a system. For instance, adding more and more syntactic possibilities counteracts ease of use for SCA systems [29].

Table 2
Relation between usability factors and criteria ('+' indicates the positive effect of criteria on usability factors).

Criteria - UI feature	Usability Factors: Efficiency	Effectiveness	Satisfaction	Learnability	Utility
Minimal Action - Single point of entry UI - Inline editing & view editing	+		+		
Minimal Memory Load - Single point of entry UI - Integration into the commonly used packages	+		+	+	+
Navigability - Faceted browsing - Faceted viewing	+	+			
Simplicity & Familiarity - Simple UIs focusing on user's main task - Integration into the commonly used packages				+	

The following features of UIs are proposed for improving the generalizability of SCA systems:

- *Supporting multiple ontologies.*
  A domain is usually described by several ontologies. For example, in a medical context there may be one ontology for general metadata about a patient and other technical ontologies that deal with diagnosis and treatment. SCA systems need to be able to support multiple ontologies [19,40,35,41–43,33,34,29,52]. In a generic SCA system, the user interface must be completely decoupled from the ontological models. Models should be able to be added at runtime and become immediately accessible to the users [43,40].
- *Supporting ontology modification.*
  A generic SCA system should provide users with user-friendly interfaces to modify the structure (classes and properties) of ontologies [19,26,30,29]. In this case, the system also needs to deal with consistency issues which might arise between ontologies and annotations with respect to ontology changes (a.k.a. *Ontology Maintenance* [19]).
- *Supporting heterogeneous document and content formats.*
  Supporting heterogeneous document and content formats is a prerequisite for integrating semantic authoring and annotation into the existing work practices [19,35]. A generic SCA system should be able to import documents in different formats such as word processor files, spreadsheets, graphics files and complex mixtures of them. It also needs to provide appropriate semantic annotations for different content types. For example, during the content annotation, a data table should be treated differently then raw text, because a table implicitly expresses relationships between the entries of a row (or column).

5.4. Collaboration

Collaboration refers to the ability of a system to support cooperation between different users of the system. An SCA system should support collaborative semantic authoring, where the authoring process can be shared among different authors at different locations. This is a key requirement of knowledge sharing between users from different fields who are contributing to and reusing intelligent documents [19,26,53]. Web 2.0 applications and related technologies provide incentives to their users for collaboration and lead to rapidly growing amounts of content. Triggered by the success of the Web 2.0 phenomenon the *Social Semantic Web* idea has gained momentum yielding tools that allow collaboration and participation incorporating semantics by lay users. As a result, many collaborative and community-driven approaches to semantic content creation have been proposed. Examples are *Semantic Wikis* and *Semantic Tagging Systems* (e.g. Faviki[25]) which exploit Web 2.0 principles and technologies to facilitate broad user participation and collaboration in the process of creating semantically enriched or annotated content [20,47]. [30] divides semantic wikis into two main categories according to their connections with the ontologies: *wikis for ontologies* and *ontologies for wikis*. The classification is very similar to our proposed top-down (cf. Section 4.2) and bottom-up approaches (cf. Section 4.1).

Access control and *supporting standard formats* are two additional independent prerequisites of collaboration in an SCA system [19,43,37]. The SCA system should allow to distinguish between writeable and non-writeable content based on the users permission level. It also needs to support standard formats which promote the collaboration and make it possible to share and reuse the generated content.

To realize collaboration, an SCA system should provide appropriate UI elements for *meta-level interactions* around different types of semantically created content such as rating, tagging and discussing. Supporting social networking features such as following other authors, subscribing to changes for watching the evolution of content [39] as well as reusing and re-purposing of content are also important to increase the collaboration in an SCA system.

5.5. Portability

Portability is the ability of a system to run under different environments. The user of an SCA system should be able to use the system at any location without installing any special software [26,41]. When focusing on Web-based UIs, compatibility between different existing web browsers and access technologies becomes an important issue. As a requirement for UI, cross-browser compatibility should ideally be ensured in an SCA system. Designing suitable UIs for mobile and ubiquitous devices is another aspect which needs to be taken into the account as powerful mobile computing devices are becoming common among the users [45].

5.6. Accessibility

Accessibility describes the degree to which a software system is available to as many people as possible. It can be viewed as the ability to access and benefit from some system. Accessibility is often used to focus on people with disabilities or special needs and their right of access to system. As mentioned in [7], papers discussing accessibility are clearly lacking in the context of Semantic Web UIs.

5.7. Proactivity

Proactivity is the ability of a system to act in advance of a future situation, rather than just reacting. It means taking control and making things happen rather than just adjusting to a situation or waiting for something to happen. An SCA system should provide users with pre-filled form fields, suggestions, default values etc. These facilities simplify the authoring process, as they reduce the number of actions users have to perform. Moreover, they reduce the possibility that users provide incomplete or empty metadata [30].

The following features of UIs are proposed for improving the proactivity of SCA systems:

- *Real-time semantic tagging.*
  Real-time tagging means creating annotations while the user is typing [35]. This will significantly increase the annotation speed [36]. Users are not distracted since they do not have to interrupt their current authoring task. This type of UI needs a client-side component which interacts with the server asynchronously.
- *Resource suggestion.*
  An SCA system should provide users with a set of entity (i.e. URI) suggestions to facilitate the annotation process for non-expert users [34,40].
- *Concept reuse.*
  An SCA system becomes increasingly advantageous, if once defined concepts (e.g. classes, properties, or instances) are as much reused and interlinked as possible [29]. Suggesting already defined concepts to users (particularly new and inexperienced users) will facilitate their contribution to the system.
- *Real-time validation.*
  When the annotation is completed by user, the SCA system should apply validation mechanisms to check the correctness of the values. Validating metadata while they are being created improves the overall quality of the documents and does not require further consistency checks, which might be difficult or even impossible once the provider of metadata has completed the job [26,30].

5.8. Automation

Automation is the ability of a system to automatically perform its intended tasks thereby reducing the need for human work. In the context of semantic authoring it means the provision of facilities for automatic mark-up of documents to facilitate the economical annotation of large document collections [19]. The automatic process of annotating is composed basically of finding terms in documents, mapping them against an ontology, and disambiguating common terms. There are a wide range of approaches that carry out automatic annotation of texts. Most of them employ natural language processing and information extraction techniques. These approaches differ in architecture, information extraction tools and methods, initial ontology, amount of manual work required to perform annotation, as well as performance [20,25].

Existing automated SCA systems can be divided into two categories: semi-automatic and fully-automatic systems. In semi-automatic systems [26,34], the user is provided with a set of suggestions to select from. So, disambiguation is performed with the help of user (i.e. incorporating user feedback to enhance the automation results). In fully-automatic systems [44,46], annotations are generated without any intervention by users. Fully-automatic systems can generally be regarded as falling into three categories [19]. The most basic kind use rules or wrappers written by hand that try to capture known patterns for the annotations. Then there are two kinds of systems that learn how to annotate. Supervised systems learn from sample annotations marked up by the user. A problem with these methods is that picking enough good examples is a non-trivial and error-prone task. In order to tackle this problem unsupervised systems employ a variety of strategies

[25] http://www.faviki.com/.

to learn how to annotate without user supervision, but their accuracy is still limited.

Automated SCA systems should take into account user interface design issues related to minimizing intrusiveness while maximizing accuracy. Completely automated systems which do not involve any user interaction in the process of semantic content creation are out of scope of this paper. User interaction is required to supervise, assess or evaluate the automated annotation thereby creating accurate semantic content.

5.9. Evolvability

Evolvability is defined as the capacity of a system for adaptive evolution. An SCA system should support evolution of the annotated document [19,35,41,43,30]. To achieve this goal, it should take into account the following consistency constraints:

- *Resource consistency*.
  If users annotate the same resource in different texts, it is important to reference the same resource in the generated RDF statements. Otherwise, we obtain many resources that are not interlinked and the statements in the repository are not very useful and meaningful [35].
- *Document and annotation consistency*.
  Supporting Ontology Modification as discussed in Section 5.3 is an important feature for the generalizability of an SCA system. In this case, the system also needs to deal with consistency issues which might arise between ontologies and annotations. One of the important issues for the design of a semantic authoring environment is to determine how changes should be reflected in the knowledge base of annotated documents and whether changes to ontologies create conflicts with existing annotations [19]. Ontologies change sometimes but some documents change many times. So, it is crucial for an SCA system to track the annotation evolution.

An SCA system should provide appropriate UIs for versioning and change tracking to deal with document and annotation evolution.

5.10. Interoperability

Interoperability is the ability of a system to work and interact with other systems. An SCA system should provide mechanisms to interoperate together with other systems which generate or consume the semantic content created. The following features of UIs are proposed for improving the interoperability of SCA systems:

- *Support of standard formats*.
  To minimize the problems of interoperability the SCA system should be built on standards. There are already many standards for semantic content serialization (e.g. typical RDF serializations and particular RDFa), representation (e.g. RDF/RDF-S/OWL/RIF and established vocabularies such as *SIOC*, *SKOS*, *FOAF*, *rNews*, etc.) and exchange (e.g. Linked Data, Web Services, REST). Supporting standard formats and avoiding proprietary formats are essential for compatibility of data with other systems [19, 26,40,35,34,43,37,44].
- *Semantic syndication*.
  Semantic syndication supports the distribution of information and their integration into other applications by providing mechanisms such as *Semantic Atom* [54] and *Semantic Pingback*[26] [29].

[26] http://aksw.org/Projects/SemanticPingBack.

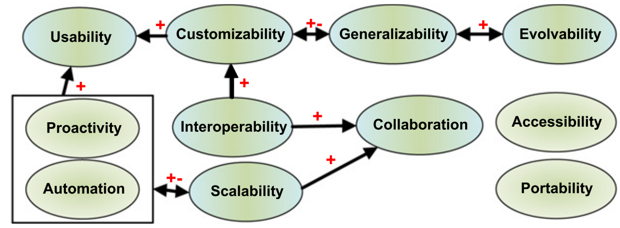

Fig. 6. Quality attributes dependences ('+': positive effect, '+−': reciprocal effect).

5.11. Scalability

Scalability refers to the capability of a system to maintain performance under an increased work load. An SCA system should support scalability as for example, the number of users, data or annotations increase. Support of *caching* and implementing a suitable *storage strategy* play an important role in achieving an scalable SCA system [29,19,41,47,34]. Annotations can be directly stored in the document or stored separately in a triple store. Most of the current SCA systems adopt a dual storage strategy of semantic annotations. In this case, annotations are stored in a server-side triple store and also embedded in the same document where annotations are undertaken in a way that is completely transparent to the user. A dual storage approach poses a redundancy but allows information from heterogeneous resources to be queried centrally and in real-time as a knowledge base [29,19].

6. Quality attributes dependences

The aforementioned quality attributes are not completely isolated and independent from each other but have overlaps and relations with each other. Fig. 6 shows an overview of these quality attributes with their interrelations. Proactivity, automation and customizability will improve the usability of an SCA system. Proactive and automatic systems provide users with helpers which facilitate the usage of the system. Customized systems are configured based on the user needs thereby increase the overall usability of the system.

Scalability will enhance the level of system collaboration. A scalable system will support more users and annotations thereby more collaboration in the system. Interoperability will also enhance the collaboration support of a system, since an interoperable system supports users of different systems. It can also support importing user data from other systems which will play a positive role in enhancing the customizability.

Evolvability and generalizability are directly related. The more evolvable to change a system is, the more generic it will be and vice versa. Customizability and generalizability share a reciprocal relation. A generic system will decrease its customization and a customizable system needs to focus on specific user needs and thus lacks generalizability. Scalability also has a reciprocal relation to proactivity and automation. Having scalability with larger data, computing proactivity and automation actions may become too heavy and complex to handle.

7. User types

Table 3 shows the list of tools discussed in our primary studies. The following tools were described in the primary studies: OntoWiki [29], SAHA [33], OWiki [30], SemCards [31], DataPress [24], RDFaCE [52], Loomp [34], Semantic MediaWiki [47], SweetWiki [40], Information Workbench [13], RDFAuthor [37], FLERSA [41], LinkedBlog [43], SemiBlog [27], HayStack semantic blogging [32], Reflect [36], Ontos-feeder [44], Epiphany [42], Linkator [25], Tabulator [39].

Table 3
User types, domain and authoring approach of the surveyed SCA systems.

Tool	Ref.	User type	Domain	Authoring approach
OntoWiki	[29]	Domain expert	Domain-independent	Top-down
OWiki	[30]	Domain expert	Domain-independent	
SAHA	[33]	Domain expert	Governmental data	
SemCards	[31]	End-user (non-expert)	Domain-independent	
RDFAuthor	[37]	End-user and domain-expert	Domain-independent	
Tabulator	[39]	End-user and domain-expert	Domain-independent	
Reflect	[36]	End-user (researchers)	Chemistry	Bottom-up
Epiphany	[42]	End-user	CMS	
Ontos-feeder	[44]	End-user (journalist)	Journalism	
DataPress	[24]	End-user (blogger)	Blogs and wikis	
RDFaCE	[52]	End-user and domain-expert	Domain-independent	
Loomp	[34]	End-user (journalist)	Journalism	
Semantic MediaWiki	[47]	End-user (wiki users)	Domain-independent	
LinkedBlog	[43]	End-user (blogger)	Blogs	
SweetWiki	[40]	End-user (wiki users)	Domain-independent	
Linkator	[25]	End-user	CMS	
FLERSA	[41]	End-user	CMS	
Information Workbench	[13]	End-user (researchers)	Paper review and publishing	
HayStack semantic blogging	[32]	End-user (blogger)	Blogs	
SemiBlog	[27]	End-user (blogger)	Blogs	

Table 4
User interface evaluation methods.

Evaluation method	Definition
Empirical methods (case study)	An empirical inquiry that investigates a contemporary phenomenon within its real-life context; A usability evaluation specialist tests a well defined hypothesis by measuring subject (user) behavior while he manipulates variables.
Discussion	Provided some qualitative, textual, opinion-oriented evaluation. e.g., compare and contrast, oral discussion of advantages and disadvantages.
Example application	Authors describe an application and provide an example to assist in the description, but the example is "used to validate" or "evaluate" as far as the authors suggest.
Observation (experience)	The result has been used on real examples, but not in the form of case studies or controlled experiments, the evidence of its use is collected informally or formally. A usability evaluation specialist acts as the observer of users as they interact with computers, noting user successes, difficulties, likes, dislikes, preferences and attitudes.
Questionnaire	The use of a set of items (questions or statements) to capture statistical data relating to user profiles, skills, experience, requirements, opinions, preferences and attitudes.
Interview	A formal consultation or meeting between a usability evaluation specialist and user(s) to obtain information about work practices, requirements, opinions, preferences and attitudes.
User groups	Availing of the wealth of knowledge and experience of organized (user forum) and selected (beta site) end users.
Cognitive walkthroughs	A step by step evaluation of a design by a cognitive psychologist in order to identify potential user psychological difficulties with the system.
Heuristic methods	The use of a team of usability evaluation specialists to review a product or prototype in order to confirm its compliance with recognized usability principles and practice.
Review methods	The review and reuse of the wealth of experimental and empirical evidence in the research literature and in the de-facto standards established by the software industry.
Simulation	Execution of a system within artificial data, using a model of the real world.
Modeling methods	Using models like GOMS (Goals, Operations, Methods and Selection) and KLM (Keystroke Level Modeling) to predict and provide feedback on user interactions and difficulties.

For each tool, we extracted the type of user, domain of the tool and the authoring approach employed in the tool. There are two general types of users mainly discussed in the studies:

- End user or normal users which have no or limited knowledge of the domain on which the annotations or semantic structures are applied. They constitute the majority of the population using the Internet to browse for information and communicate with others.
- Domain experts which have a broad knowledge of the domain on which the annotations or semantic structures are applied. They are usually consisted of the researchers or engineers with a top-down view of problems.

As our results revealed, the majority of studies (i.e. all the tools which employed the bottom-approach) were addressing tools which are appropriate for end users. Tools which were adopting the top-down approach needed users to have knowledge of the corresponding domain as well as ontology concepts. It is worth mentioning that by domain-independent, we mean that the tool is not limited to any predefined domains and is flexible enough to be applied in arbitrary domains. For instance, *OntoWiki* is domain-independent while it requires a domain-expert as user. This is due to the fact, that when a user wants to create semantic content using OntoWiki, he should have a broad view on the selected domain to define the required ontologies (i.e. the available classes, their possible relationships, constraints, class properties, data types, etc.) and to populate the data accordingly. Otherwise he cannot create semantic content with the tool.

8. User interface evaluation

In this section we briefly outline various methods for user interface evaluation and report about their usage in the surveyed papers. Table 4 lists existing methods for user interface evaluation adopted from [55,56].

Among the primary studies, the majority of studies [29,30,34, 43,31,37,13,25,41,27,44,39,47] were only using an *Example Application* as their evaluation method. A few studies [33,40,36,42,32] were also using the *Discussion* method together with an example application. A very few ones [24,52] used *Interview* and *Questionnaire* methods for UI evaluation. Other papers were either survey papers or the papers which only mentioned some user interface features and did not provide any UI evaluation method. The results

Table 5
Comparison of OntoWiki, SAHA 3, Loomp and RDFaCE according to the quality attributes.

	OntoWiki	SAHA 3	Loomp	RDFaCE
Usability	• Single point of entry UI • Faceted browsing • Inline editing/view editing	• Single point of entry UI • Faceted browsing • Inline editing	• Single point of entry UI • Faceted viewing	• Single point of entry UI • Faceted viewing • Inline editing
Customizability	• Semantic views: domain specific & generic (e.g. map, calendar)	-	-	• Semantic views: WYSIWYM, WYSIWYG, triple, source code view
Generalizability	• Multiple ontology support • Ontology modification support	• Multiple ontology support	• Multiple ontology support	• Multiple ontology support
Collaboration	• Access control • Standard formats: RDF, RDFa • Social collaboration UIs: rating and commenting UIs	• Access control • Standard formats: RDF • Social collaboration UIs: online chat	• Standard formats: RDF, RDFa	• Standard formats: RDFa
Portability	• Cross-browser compatibility • UI for mobile devices	• Cross-browser compatibility	• Cross-browser compatibility	• Cross-browser compatibility
Accessibility	-	-	-	-
Proactivity	• Resource suggestion • Concept reuse	• Resource suggestion • Concept reuse • Real-time validation	• Resource suggestion • Concept reuse	• Resource suggestion
Automation	-	-	-	• Automatic annotation: using NLP APIs
Evolvability	• Resource consistency • Document & annotation consistency • Versioning & change tracking	• Resource consistency • Document and annotation consistency	• Document and annotation consistency	• Document and annotation consistency
Interoperability	• Standard formats: RDF, RDFa • Semantic syndication: semantic pingback	• Standard formats: RDF	• Standard formats: RDF, RDFa	• Standard formats: RDFa
Scalability	• Caching support • Storage strategy: backend independent (Mysql, Virtuoso)	• Storage strategy: using a server-side triple store	• Storage strategy: using a server-side triple store	• Storage strategy: using on-the-fly client -side triple storage

distinctly exposes the lack of formal and systematic UI evaluation methods in the context of SCA systems.

Analyzing the suitability of each evaluation method for measuring the quality attributes introduced in Section 5 is beyond the scope of this paper but to bring some examples: Among the evaluation methods, *Simulation* is suitable to measure the *Scalability* of an SCA system. Most of the evaluation methods (e.g. *Empirical Methods*, *Questionnaire*, *Interview*, *Observation* and *Modeling Methods*) can be used to measure the *Usability* of an SCA system. *Observation* method seems to be suitable to measure the level of *Collaboration* in an SCA system and so on.

9. Example tools

In this section we look at four available SCA systems and compare them according to the quality attributes defined in Section 5. We will investigate their strengths and weaknesses based on our proposed taxonomy of quality attributes and UI features which are required for SCA systems. We have selected these four example tools so that top-down and bottom-up tools, domain expert and end user tools as well as domain-independent and domain-specific tools are covered. Among the tools two (i.e. OntoWiki and SAHA 3) follow the top-down approach (cf. Section 4.2) and two (i.e. Loomp and RDFaCE) follow a bottom-up approach (cf. Section 4.1) for semantic content authoring.

We used the criteria *availability of an online demo*, *availability of technical implementation details*, having *up-to-date support* and *number of quality attributes addressed in the tool* to select these 4 tools out of the 20 tools discovered in the literature (cf. Table 3). Table 5 summarizes the assessment of the tools according to the defined quality attributes.

9.1. OntoWiki

OntoWiki[27] [29] is a tool that provides support for agile, distributed knowledge engineering scenarios. OntoWiki facilitates the visual presentation of a knowledge base as an information map, with different views on instance data.

Regarding the technical realization, the system is implemented in *PHP* using the *Zend framework*.[28] It supports the *MySQL* database and the *Virtuoso* triple store[29] as storage backends and the authoring interface is built using jQuery UI.[30]

Fig. 7 shows a screenshot of OntoWiki. OntoWiki was applied in a number of use cases. Examples include: semantic content management [57], collaborative requirements engineering with *SoftWiki* [58] and historical, prosopographical knowledge engineering with the *Professor's Catalogue* [59].

OntoWiki as a single point of entry UI adopts the top-down approach for semantic authoring. It provides a semantic search feature with support for faceted browsing. It also supports two complementary knowledge base authoring strategies: (a) *Inline editing*, which enables users to edit small information chunks (i.e. statements). (b) *View editing*, which enables users to edit common combinations of information (such as an instance of a distinct class) in one single step. In order to do so, OntoWiki uses *RDFAuthor* [37]

27 http://ontowiki.net.
28 http://framework.zend.com/.
29 http://virtuoso.openlinksw.com/.
30 http://jqueryui.com/.

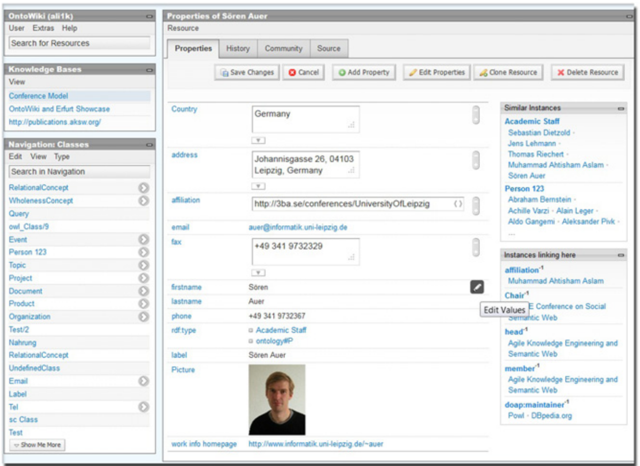

Fig. 7. Screenshot of the OntoWiki instance view with inline editing.

to make generated RDFa views editable. Regarding the customizability, OntoWiki supports different semantic views of the knowledge base which can be generic or domain-specific. It also supports editing multiple ontologies including both the instances and structures of the ontologies. As a Web-based system, it provides cross-browser compatibility and has a specific UI for mobile devices. To provide proactivity, OntoWiki uses the AJAX technology to interactively propose already defined concepts while the user types in new information to be added to the knowledge base (i.e. Concept Reuse).

OntoWiki also provides versioning and evolution features to track, review and selectively roll-back changes and supports semantic syndication (employing *Semantic Pingback* and Linked Data interfaces) to interoperate with other systems. OntoWiki is backend independent to some extend and supports two different types of storage engines. It also provides a caching component to optimize the performance of the system.

As a drawback, OntoWiki does not provide any UI elements to facilitate accessibility and automation. It supports only the editing of structured content thus lacking UIs for the annotation of unstructured or semi-structured content. Furthermore, it does not provide real-time tagging and validation for increasing the overall proactivity.

9.2. SAHA 3 metadata editor

SAHA 3[31] [33] is an RDF metadata editor for collaborative content creation and instant semantic content publishing on the Semantic Web. Regarding the technical realization, the system is implemented in Java on top of the *Spring framework*.[32] The data model is based on the *Jena TDB*[33] RDF database and the editor interface is built using *DWR*[34] and the *Dojo*[35] AJAX components.

Fig. 8 shows a screenshot of SAHA 3. *DataFinland*[36] as a semantic portal for open and linked datasets is one use case of SAHA 3.

Like OntoWiki, SAHA 3 uses the top-down approach for semantic authoring and a single point of entry UI with inline editing features. It supports faceted browsing when integrated into

[31] http://demo.seco.tkk.fi/saha/saha3/.

[32] http://www.springsource.com/.

[33] http://openjena.org/TDB/.

[34] http://directwebremoting.org/.

[35] http://www.dojotoolkit.org/.

[36] http://www.seco.tkk.fi/linkeddata/datasuomi/.

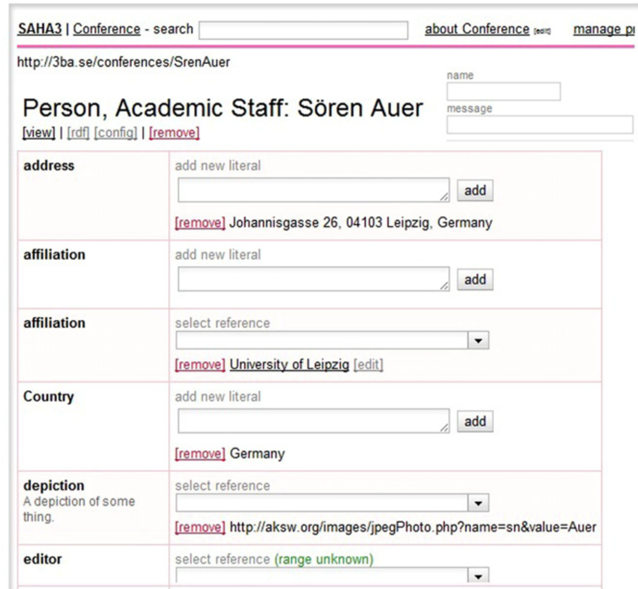

Fig. 8. Screenshot of the SAHA 3 inline editing.

the faceted portal engine HAKO[37] for content publishing. SAHA 3 supports multiple ontologies as well as collaborative simultaneous editing. Resources that are being edited by one user are locked for editing by other users. A chat facility has been integrated into the editor to facilitate instant discussions between peer editors.

Regarding proactivity, SAHA 3 also provides real-time semantic validation and concept reuse. As shown in [60], SAHA 3 has proven a good level of scalability to support large projects.

As a drawback, SAHA 3 does not provide any UI elements to provide customizability, accessibility and automation. Like OntoWiki, it only supports structured content authoring thus lacking the appropriate UIs for unstructured or semi-structured content annotation. Although it provides some simple UIs for supporting collaboration but lacks sophisticated features regarding social interactions. Since it does not provide any UI elements for versioning and change tracking, evolvability is not well addressed. Scalability could be further improved by adding support for caching and alternative storage backends (i.e. client-side RDF processing).

9.3. Loomp

Loomp[38] [34] is a tool representing a prove-of-concept for the *One Click Annotation* (OCA) strategy. The Web-based OCA editor allows for annotating words and phrases with references to ontology concepts and for creating relationships between annotated phrases [35].

Regarding the technical realization, Loomp is a typical Web application built on the LAMP stack.[39] It serves content either in RDF (e.g. for linked data clients) or in XHTML/RDFa (e.g. for Web browsers).

Fig. 9 shows a screenshot of Loomp. *Data-driven journalism* is mentioned as one of the primary uses cases of Loomp.

Loomp provides a WYSIWYG editor as a single point of entry UI which adopts the bottom-up approach for semantic content authoring. It supports a faceted viewing feature which highlights

[37] http://www.seco.tkk.fi/tools/hako/.

[38] http://loomp.org.

[39] LAMP: Linux operating system, Apache web server, MySQL database, PHP scripting language.

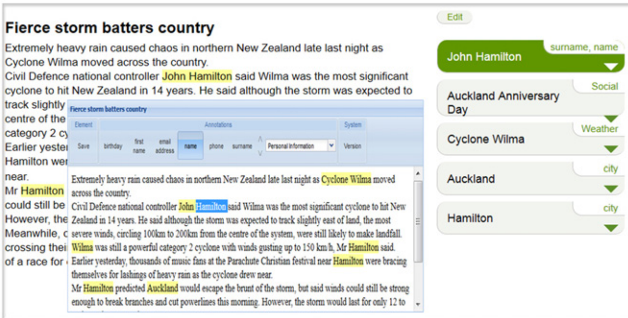

Fig. 9. Screenshot of the Loomp faceted viewing UI.

user-selected annotations in the Web browser. Loomp facilitates concept reuse and suggestion in order to reduce non-expert user efforts during the annotation process. It also employs RDF and RDFa standard formats which make it interoperable with other systems.

As a drawback, Loomp lacks appropriate UI elements to support customizability, accessibility and automation. It does not provide any UI features for faceted browsing and inline editing of annotations. It also does not allow to directly edit the underlying ontologies thereby extending the annotation domain. Furthermore, Loomp lacks appropriate UI elements for real-time tagging and validation as well as versioning and change tracking. Regarding scalability, no information could be found on how Loomp supports large amounts of users and annotations.

9.4. RDFaCE

RDFaCE[40] [52] (RDFa Content Editor) is a Web-based RDFa content editor based on the TinyMCE[41] rich text editor. It supports different views for semantic content authoring and uses existing Semantic Web APIs to facilitate annotating and editing of RDFa content. RDFaCE introduces the WYSIWYM (What-You-See-Is-What-You-Mean) concept for semantic text authoring which is an extension (incorporating semantic views) to the WYSIWYG (What-You-See-Is-What-You-Get) editing model.

Regarding the technical realization, RDFaCE is written completely in JavaScript and utilizes the *jQuery UI library* for the authoring user interface. Annotations are created on-the-fly using the client-side triple store RDFQuery,[42] which makes a separate backend for storing annotations obsolete.

Fig. 10 shows an screenshot of RDFaCE. Use cases of the RDFaCE include the annotation of news articles with metadata using the *rNews vocabulary*[43] and semantic blogging with *Wordpress*.

Like Loomp, RDFaCE consists of a single point of entry UI which supports faceted viewing and inline editing of annotations. It provides different semantic views for different personas involved in the process of semantic content authoring. Furthermore, RDFaCE supports resource suggestion and automatic content annotation using external NLP APIs. Since RDFaCE processes the annotations client-side within the user's browser and does not require any central storage backend, RDFaCE is highly scalable.

As a drawback, RDFaCE lacks the appropriate UI elements to support accessibility. It also does not provide a faceted browsing UI and does not allow to directly edit the underlying annotation ontologies. Beside supporting the RDFa standard format, it does not provide any UI features for social collaboration as well as versioning and change tracking.

[40] http://rdface.aksw.org.
[41] http://tinymce.moxiecode.com.
[42] http://code.google.com/p/rdfquery/.
[43] http://dev.iptc.org/rNews.

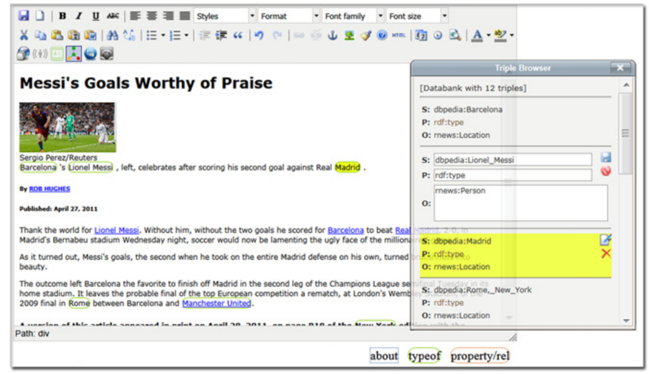

Fig. 10. Screenshot of the RDFaCE triple viewer and editor.

10. Research and technology challenges

The results of our systematic review revealed some research and technology gaps and corresponding challenges with regard to the development of SCA UIs.

i. *Accessibility.* There is a clear research gap in addressing accessibility issues during the design of SCA UIs. Many semantic authoring tools remain inaccessible to people with disabilities. Providing people with disabilities and special needs with appropriate SCA UIs can facilitate their tasks of information seeking. Semantically annotated content allows alternatives or conditional content in different modalities to be selected based on the type of the user disability and information need. For example, visually impaired people, need significantly more time to grasp the structure and gist of a Web site, since visual navigation and structuring elements are not accessible to them. Once content is semantically annotated, visually impaired people can use this semantic annotation as a means to access and explore the content faster.

The W3C's Web Content Accessibility Guidelines (WCAG)[44] explain how to make Web content more accessible to people with disabilities. As part of WCAG, *Authoring Tool Accessibility Guidelines* (ATAG)[45] and more specifically *Accessible Rich Internet Applications* (WAI-ARIA)[46] suite, define how authoring tools should support accessibility requirements. Consequently, a challenge is to apply and extend the series of accessibility guidelines proposed in ATAG for the purpose of designing accessible SCA UIs.

ii. *Handling complexity in UIs.* One important concern when designing SCA UIs is how to make complex functionality available to the user in a simple way. There are two issues in this context which need to be addressed. The first one is how to properly map complex functions and algorithms (e.g. entity disambiguation, recommendation and other machine learning algorithms) to corresponding user interface elements. The second issue is how to flatten the user's learning curve by providing adaptive and intelligent UIs which take user knowledge into account. Many current SCA systems bear a bewildering amount of functions and algorithms which confuses both the novice and expert users. This problem causes a significant impediment for a broader use of SCA systems.

Addressing the complexity problem requires the creation of abstract models for complex tasks as well as modeling the user characteristics and behavior. Ideally, the SCA UI should present the users with concepts that are consistent with both designer and users' mental models of that phenomenon in the real world.

[44] http://www.w3.org/WAI/intro/wcag.php.
[45] http://www.w3.org/WAI/intro/atag.php.
[46] http://www.w3.org/TR/wai-aria/.

The above mentioned issues are well addressed in designing Geographic Information System (GIS) UIs [61,62]. Now it is a challenge to rethink these issues for the purpose of designing adaptive and flexible SCA UIs.

iii. *Formal and systematic methods for user interface evaluation.* The results of our survey clearly reflects the lack of formal and systematic UI evaluation methods in evaluating SCA systems. As described in Table 4, there are several UI evaluation methods which can be used in this context. Nielsen and Molich [63] enumerate four general categories of systematic user interface evaluation methods: formally by employing an analysis technique; automatically by a computerized procedure; empirically by testing users performing experiments; and heuristically.

In *heuristic evaluation*, evaluators inspect a user interface against a guideline to identify usability problems that violate any items on the guideline [64]. Our list of quality attributes and UI features (cf. Section 5) can be used as a guideline for heuristic evaluation of SCA system UIs. This will require less resources than testing with users and can be applied to the system during the design phase.

iv. *Support of crowdsourcing.* One of the missing aspects of developing collaborative SCA systems is the support of crowdsourcing. There is a huge amount of amateur and expert users who collaborate on and contribute to the Social Web. Harnessing the power of such crowds can significantly enhance and widen the results of semantic content authoring and annotation. Crowdsourcing as a distributed problem-solving and production model is defined to address this aspect of collective intelligence [65].

In order to support crowdsourcing, an SCA system needs to provide appropriate UIs. In [66], Geiger et al. analyze the respective characteristics and requirements related to the design of crowdsourcing systems. Providing small contributions with instant gratification, altruism and a way to establish a reputation are some of these requirements. As a new challenge, it is worth to consider these characteristics when designing SCA UIs.

v. *UIs for ubiquitous devices.* As discussed in Section 5.5, creating UIs for mobile and ubiquitous devices is an issue which is not well addressed in the literature yet. Mobile and ubiquitous devices are rapidly becoming the central computing and communication devices in people's lives. Ubiquitous computing (a.k.a *everyware* [67]) presents new challenges in user interface design. Emerging ubiquitous devices are programmable and come with a growing set of facilities including multi-touch screens and cheap powerful embedded sensors, such as an accelerometer, digital compass, gyroscope, GPS, microphone, and camera [68]. Utilizing these rich set of UI facilities when developing SCA systems can improve the user experience in the process of semantic content authoring and annotation. For example, users can easily share their real-time activities with SCA system using mobile sensors or can use some gestures for annotating the content.

Another challenge here is the ability to provide offline functionality with local updates for later synchronization with a web server. SCA systems should take into account the reconciliation problem—the problem of potentially conflicting updates from disconnected clients.

11. Conclusions and future work

In this paper we reported on the results of a systematic literature review on user interfaces for semantic content authoring comprising initially 175 papers. The review aimed to answer the five research questions defined in Section 2.1 by thorough analysis of the 31 most relevant papers. Before addressing the defined research questions, we drew a terminology which defines the basic concepts used in the literature as well our survey. To answer the RQ1, we classified existing approaches for SCA into two categories namely top-down and bottom-up discussed in Section 4. Furthermore, Our data analysis revealed a set of quality attributes for SCA systems together with the corresponding user interface features which are suggested for their realization. These quality attributes plus the UI features are used to answer the RQ2 and RQ3. In order to answer RQ4 and RQ5 we extracted the types of users as well as user evaluation methods discussed in the primary studies and reflected the results in Section 8. Open research and technological challenges in the domain of SCA systems were discussed as well. Additionally, to show the applicability of the results, we performed an in-depth comparison of four existing SCA systems according to the defined quality attributes and described their strengths as well as their weaknesses. Table 6 shows an overview of the results surveyed in this paper.

Essential, foundational quality attributes for an SCA system are, in particular, usability, generalizability, customizability and evolvability. A basic SCA system should fulfill a reasonable level of user-friendliness and adopt to different situations or use cases while providing mechanisms to tailor its functionality based on specific user needs. It also should take into account issues such as consistency constraints and content evolution which are required for change management. Support of collaboration, interoperability and scalability are quality attributes required when an SCA system is employed in a community-driven environment with large amount of users, systems and interactions. An SCA system should support standard formats and provide appropriate UI elements for social networking including both human-to-human as well as system-to-system interactions. Additionally, it should maintain performance under an increased work load by supplying appropriate storage and caching mechanisms. Automation and proactivity are quality attributes which facilitate usability of SCA systems especially for non-skilled users. Portability and accessibility are, as our survey indicates, not well addressed by the literature so far. The demands for suitable UIs for mobile and ubiquitous devices are increasing as powerful mobile computing devices are becoming more common. Furthermore, providing accessible UIs for people with disabilities or special needs is another requirement which should be taken into account when designing SCA systems.

While there are many benefits of systematic reviews, they also bear some limitations and validity threats originating from human errors. The main threats to validity of our systematic review are twofold: correct and thorough selection of the studies to be included as well as accurate and exhaustive selection of quality attributes together with their corresponding UI features. With the increasing number of works in the area of semantic content authoring we cannot guarantee to have captured all the material in this area. The scope of our review is restricted to the scientific domain. Therefore, some tools or approaches employed in the industry might have not been included in our primary studies. Furthermore, since the review process was mainly performed by one researcher a bias is possible. In order to mitigate a potential subjective bias, the review protocol and results were checked and validated by a senior researcher and other colleagues experienced in the context of Semantic Web.

We see this effort and in particular the identification of a comprehensive set of quality attributes as a crucial step towards developing more effective and user-friendly authoring tools for realizing the Social Semantic Web. New approaches and tools can be evaluated in the light of these quality attributes, thus revealing additional aspects to be taken into consideration. As a result, more user-friendly tools will enable more people to interact with the Semantic Web thereby facilitating the realization of the intelligent Web vision.

As future work, we envision strategies to semi-automatically improve the realization of the quality attributes, for example, using

Table 6
Overview of the results.

Quality Attribute	UI Feature	User Type		Authoring Approach		Domain	
		Expert	Non-expert	Top-Down	Bottom-Up	Domain-independent	Domain-specific
Usability	Single Point of Entry Interface	OntoWiki, RDFaCE, SAHA	HayStack semantic blogging, RDFaCE, Loomp	OntoWiki, SAHA	HayStack semantic blogging, RDFaCE, Loomp	OntoWiki, RDFaCE	HayStack semantic blogging, SAHA, Loomp
	Faceted Browsing	OntoWiki, SAHA	-	OntoWiki, SAHA	-	OntoWiki	SAHA
	Faceted Viewing	RDFaCE	Reflect, Loomp, RDFaCE	-	Reflect, Loomp, RDFaCE	RDFaCE	Reflect, Loomp
	Inline Editing and View Editing	OntoWiki, RDFauthor, SAHA, RDFaCE	RDFauthor, RDFaCE	OntoWiki, RDFauthor, SAHA	RDFaCE	OntoWiki, RDFauthor, RDFaCE	SAHA
Customizability	Living UIs	-	-	-	-	-	-
	Providing Different Semantic Views	OntoWiki, Owiki, Tabulator, RDFaCE	Tabulator, Information Workbench, RDFaCE	OntoWiki, Owiki, Tabulator	Information Workbench, RDFaCE	OntoWiki, Owiki, Tabulator, RDFaCE	Information Workbench
Generalizability	Supporting Multiple Ontologies	OntoWiki, , RDFaCE, SAHA	SweetWiki, FLERSA, Epiphany, LinkedBlog, , RDFaCE, Loomp	OntoWiki, SAHA	SweetWiki, FLERSA, Epiphany, LinkedBlog, , RDFaCE, Loomp	OntoWiki, SweetWiki, , RDFaCE	FLERSA, Epiphany, LinkedBlog, SAHA, Loomp
	Supporting Ontology Modification	OntoWiki, Owiki	SemCards	OntoWiki, SemCards, Owiki	-	OntoWiki, SemCards, Owiki	-
	Supporting Heterogeneous Document/Content Formats	-	-	-	-	-	-
Collaboration	Access Control	OntoWiki, RDFauthor, SAHA	RDFauthor, LinkedBlog	OntoWiki, RDFauthor, SAHA	LinkedBlog	OntoWiki, RDFauthor	LinkedBlog, SAHA
	Support of Standard Formats	OntoWiki, RDFauthor,RDFaCE, SAHA	SweetWiki, LinkedBlog, RDFauthor, RDFaCE, Loomp, SemiBlog	OntoWiki, RDFauthor, SAHA	SweetWiki, LinkedBlog, RDFaCE, Loomp, SemiBlog	OntoWiki, SweetWiki, RDFauthor, RDFaCE	LinkedBlog, SAHA, Loomp, SemiBlog
	UIs for Social Collaboration	OntoWiki, SAHA	Tabulator	OntoWiki, SAHA, Tabulator	-	OntoWiki, Tabulator	SAHA
Portability	Cross-browser Compatibility	OntoWiki, SAHA	FLERSA, Loomp	OntoWiki, SAHA	FLERSA, Loomp	OntoWiki,	FLERSA, SAHA, Loomp
	UIs for Mobile Devices	OntoWiki, RDFaCE	RDFaCE	OntoWiki	RDFaCE	OntoWiki, RDFaCE	-
Accessibility	Accessible UIs	-	-	-	-	-	-
Proactivity	Resource Suggestion	OntoWiki, RDFaCE, SAHA	Loomp, SweetWiki, RDFaCE	OntoWiki, SAHA	Loomp, SweetWiki, RDFaCE	OntoWiki, SweetWiki, RDFaCE	Loomp, SAHA
	Real-time Semantic Tagging	-	Reflect	-	Reflect	-	Reflect
	Concept Reuse	OntoWiki, SAHA	Reflect, SweetWiki, Loomp	OntoWiki, SAHA	Reflect, SweetWiki, Loomp	OntoWiki, SweetWiki	Reflect, SAHA, Loomp
	Real-time Validation	Owiki, SAHA	-	Owiki, SAHA	-	Owiki	SAHA
Automation	Automatic Annotation	RDFaCE	Linkator, Ontos-Feeder, Epiphany, RDFaCE	-	Linkator, Ontos-Feeder, Epiphany, RDFaCE	RDFaCE	Linkator, Ontos-Feeder, Epiphany
Evolvability	Resource Consistency	OntoWiki, SAHA	-	OntoWiki, SAHA	-	OntoWiki	SAHA
	Document and Annotation Consistency	OntoWiki, RDFaCE, SAHA	Loomp, RDFaCE	OntoWiki, SAHA	Loomp, RDFaCE	OntoWiki, RDFaCE	Loomp, SAHA
	Versioning and Change Tracking	OntoWiki	Semantic MediaWiki	OntoWiki	Semantic MediaWiki	OntoWiki, Semantic MediaWiki	-
Interoperability	Support of Standard Formats	OntoWiki, RDFauthor, RDFaCE, SAHA	SweetWiki, LinkedBlog, RDFauthor, RDFaCE, Loomp, SemiBlog	OntoWiki, RDFauthor, SAHA	SweetWiki, LinkedBlog, RDFaCE, Loomp, SemiBlog	OntoWiki, SweetWiki, RDFauthor, RDFaCE	LinkedBlog, SAHA, Loomp, SemiBlog
	Semantic Syndication	OntoWiki	-	OntoWiki	-	OntoWiki	-
Scalability	Support of Caching	OntoWiki	Semantic MediaWiki	OntoWiki	Semantic MediaWiki	OntoWiki, Semantic MediaWiki	-
	Suitable Storage Strategies	OntoWiki, RDFaCE, SAHA	Loomp, FLERSA, RDFaCE	OntoWiki, SAHA	Loomp, FLERSA, RDFaCE	OntoWiki, RDFaCE	Loomp, FLERSA, SAHA

active machine learning for better integration with approaches delivering automatic suggestions. Also extending the support for integration of multimedia and multi-modal semantic annotation (e.g. of images and multimedia content) is a promising research direction.

Integrating SCA systems into other applications like speech recognition and question-answering systems for improving the accuracy and quality of results is another important area of future work. At the moment, intelligent mobile assistants (e.g. *Siri*[47] for the iPhone) only allow delegation of certain programmed tasks (e.g. making restaurant reservations, getting movie tickets, etc.) by invoking certain predefined web services. Employing semantically enriched content in the UI of mobile personal agents will extend their capability to inquiry the open Web of Data thereby achieving more efficient and effective results.

Addressing open research and technology challenges such as accessibility, handling complexity in UIs, formal and systematic methods for user interface evaluation, support of crowdsourcing and UIs for ubiquitous devices discussed in the paper are other interesting areas for future research.

Acknowledgments

We would like to thank our colleagues from AKSW research group for their helpful comments and inspiring discussions during the conducting of this systematic review. This work was supported by a grant from the European Union's 7th Framework Programme provided for the project LOD2 (GA no. 257943).

References

[1] Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert Nuffelen, Claus Stadler, Sebastian Tramp, Hugh Williams, Managing the life-cycle of linked data with the LOD2 stack, in: Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, JosianeXavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, Eva Blomqvist (Eds.), The Semantic Web—ISWC 2012, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2012, pp. 1–16.

[2] Mohamed Morsey, Jens Lehmann, Sören Auer, Axel-Cyrille Ngonga Ngomo, DBpedia SPARQL Benchmark—performance assessment with real queries on real data, in: ISWC 2011, 2011.

[3] Axel-Cyrille Ngonga Ngomo, Lars Kolb, Norman Heino, Michael Hartung, Sören Auer, Erhard Rahm, When to reach for the cloud: Using parallel hardware for link discovery, in: Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, Sebastian Rudolph (Eds.), The Semantic Web: Semantics and Big Data, in: Lecture Notes in Computer Science, vol. 7882, Springer, Berlin, Heidelberg, 2013, pp. 275–289.

[4] Sören Auer, Jan Demter, Michael Martin, Jens Lehmann, Lodstats—an extensible framework for high-performance dataset analytics, in: Annette Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Acquin, Andriy Nikolov, Nathalie Aussenac-Gilles, Nathalie Hernandez (Eds.), Knowledge Engineering and Knowledge Management, in: Lecture Notes in Computer Science, vol. 7603, Springer, Berlin, Heidelberg, 2012, pp. 353–362.

[5] Benjamin Heitmann, Sheila Kinsella, Conor Hayes, Stefan Decker, Implementing semantic web applications: reference architecture and challenges, in: 5th International Workshop on Semantic Web-Enabled Software Engineering, 2009.

[6] H. Paulheim, F. Probst, Ontology-enhanced user interfaces: A survey, Int. J. Semant. Web Inf. Syst. 6 (2) (2010).

[7] Ghislain Hachey, Dragan Gašević, Semantic web user interface: a systematic survey, Master's Thesis, Athabasca University, 2011.

[8] Daniel Tunkelang, Faceted Search, in: Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan and Claypool Publishers, 2009.

[9] Vanessa Lopez, Victoria Uren, Marta Sabou, Enrico Motta, Is question answering fit for the semantic web? A survey, Semantic Web—Interoperability, Usability, Applicability 2 (2) (2011) 125–155.

[10] Gregoire Burel, Amparo E. Cano, Vitaveska Lanfranchi, Ozone browser: augmenting the web with semantic overlays, in: Sören Auer, Chris Bizer, Gunnar AAstrand Grimnes (Eds.), Proc. of 5th Workshop on Scripting and Development for the Semantic Web at ESWC 2009, CEUR Workshop Proceedings, vol. 449, ISSN: 1613-0073. June 2009. www.CEUR-WS.org/Vol-449.

[11] Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, Denny Vrandecic, The two cultures: mashing up web 2.0 and the semantic web, in: WWW'07: Proceedings of the 16th International Conference on World Wide Web, ACM Press, New York, NY, USA, 2007, pp. 825–834.

[12] Melike Sah, Wendy Hall, Nicholas M. Gibbins, David C. De Roure, Semport? A personalized semantic portal, in: 18th ACM Conference on Hypertext and Hypermedia, Sheridan Printing, 2007, pp. 31–32.

[13] Wolfgang Müller, Isabel Rojas, Andreas Eberhart, Peter Haase, Michael Schmidt, A-R-E: the author-review-execute environment, Procedia Comput. Sci. 4 (2011) 627–636. Proceedings of the International Conference on Computational Science, ICCS 2011.

[14] Tore Dyba, Torgeir Dingsoyr, Geir K. Hanssen, Applying systematic reviews to diverse study types: An experience report, in: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, ESEM'07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 225–234.

[15] B. Kitchenham, Procedures for performing systematic reviews, Technical Report, Keele University and NICTA, 2004.

[16] Barney G. Glaser, Anselm L. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine de Gruyter, New York, NY, 1967.

[17] Carolyn B. Seaman, Qualitative methods in empirical studies of software engineering, IEEE Trans. Softw. Eng. 25 (4) (1999) 557–572.

[18] Matthew B. Miles, Michael Huberman, Qualitative Data Analysis: An Expanded Sourcebook, second ed., Sage Publications, Inc., 1994.

[19] Victoria Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, Fabio Ciravegna, Semantic annotation for knowledge management: requirements and a survey of the state of the art, Web Semantics: Science, Services and Agents on the World Wide Web 4 (1) (2006) 14–28.

[20] Katharina Siorpaes, Elena Simperl, Human intelligence in the process of semantic content creation, World Wide Web 13 (1-2, SI) (2010) 33–59.

[21] Hui-Chuan Chu, Ming-Yen Chen, Yuh-Min Chen, A semantic-based approach to content abstraction and annotation for content management, Expert Syst. Appl. 36 (2, Part 1) (2009) 2360–2376.

[22] Phillip Sheu, Heather Yu, C.V. Ramamoorthy, Arvind K. Joshi, Lotfi A. Zadeh, Semantic Computing, Wiley, IEEE Press, 2010.

[23] Kôiti Hasida, Semantic authoring and semantic computing, in: Proceedings of the 2003 and 2004 International Conference on New Frontiers in Artificial Intelligence, JSAI'03/JSAI04, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 137–149.

[24] Edward Benson, Adam Marcus, Fabian Howahl, David Karger, Talking about data: sharing richly structured information through blogs and wikis, in: Peter Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Pan, Ian Horrocks, Birte Glimm (Eds.), The Semantic Web—ISWC 2010, in: Lecture Notes in Computer Science, vol. 6496, Springer, Berlin, Heidelberg, 2010, pp. 48–63. http://dx.doi.org/10.1007/978-3-642-17746-0-4.

[25] Samur Araujo, Geert-Jan Houben, Daniel Schwabe, Linkator: enriching web pages by automatically adding dereferenceable semantic annotations, in: Boualem Benatallah, Fabio Casati, Gerti Kappel, Gustavo Rossi (Eds.), Web Engineering, in: Lecture Notes in Computer Science, vol. 6189, Springer, Berlin, Heidelberg, 2010, pp. 355–369. http://dx.doi.org/10.1007/978-3-642-13911-6-24.

[26] Onni Valkeapaeae, Olli Alm, Ero Hyvoenen, An adaptable framework for ontology-based content creation on the semantic web, Journal of Universal Computer Science 13 (12) (2007) 1835–1853. 28th Annual Meeting of the Society-of-Behavioral-Medicine, Washington, DC, MAR 21-24, 2007.

[27] Knud Moller, Uldis Bojars, John C. Breslin, Using semantics to enhance the blogging experience, in: Y. Sure, J. Domingue (Eds.), Semantic Web: Research and Applications, Proceedings, Lecture Notes in Computer Science, vol. 4011, pp. 679–696. Sekt, Dip, Knowledgeweb, Asg, Ontotext, iSOCO, Luisa, AKT, Oasis, 2006, 3rd European Semantic Web Conference, Budva, SERBIA MONTENEG, JUN 11-14, 2006.

[28] Vincent Quint, Irène Vatton, Structured templates for authoring semantically rich documents, in: Proceedings of the 2007 International Workshop on Semantically Aware Document Processing and Indexing, SADPI'07, ACM, New York, NY, USA, 2007, pp. 41–48.

[29] Sören Auer, Sebastian Dietzold, Thomas Riechert, Ontowiki—a tool for social, semantic collaboration, in: Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, Lora Aroyo (Eds.), The Semantic Web—ISWC 2006, in: Lecture Notes in Computer Science, vol. 4273, Springer, Berlin, Heidelberg, 2006, pp. 736–749. http://dx.doi.org/10.1007/11926078-53.

[30] Angelo Di Iorio, Alberto Musetti, Silvio Peroni, Fabio Vitali, Ontology-driven generation of wiki content and interfaces, New Rev. Hypermedia Multimedia 16 (1–2, SI) (2010) 9–31.

[31] Kristinn Thórisson, Nova Spivack, James Wissner, The semantic web: from representation to realization, in: Ngoc Nguyen, Ryszard Kowalczyk (Eds.), Transactions on Computational Collective Intelligence II, in: Lecture Notes in Computer Science, vol. 6450, Springer, Berlin, Heidelberg, 2010, pp. 90–107. http://dx.doi.org/10.1007/978-3-642-17155-0-5.

[32] David R. Karger, Dennis Quan, What would it mean to blog on the semantic web? Web Semantics: Science, Services and Agents on the World Wide Web 3 (2-3) (2005) 147–157. Selcted Papers from the International Semantic Web Conference, 2004—ISWC, 2004.

[47] http://www.siri.com.

[33] Matias Frosterus, Eero Hyvönen, Joonas Laitio, Datafinland—a semantic portal for open and linked datasets, in: Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, Jeff Pan (Eds.), The Semanic Web: Research and Applications, in: Lecture Notes in Computer Science, vol. 6644, Springer, Berlin, Heidelberg, 2011, pp. 243–254. http://dx.doi.org/10.1007/978-3-642-21064-8-17.

[34] Ralf Heese, Markus Luczak-Roesch, Linked data authoring for non-experts, in: Proceedings of the WWW09, Workshop Linked Data on the Web, LDOW2009, 2009.

[35] Ralf Heese, Markus Luczak-Rösch, Radoslaw Oldakowski, Olga Streibel, Adrian Paschke, One click annotation, in: Scripting and Development for the Semantic Web, SFSW, 2010.

[36] Seán I. O'Donoghue, Heiko Horn, Evangelos Pafilis, Sven Haag, Michael Kuhn, Venkata P. Satagopam, Reinhard Schneider, Lars J. Jensen, Reflect: a practical approach to web semantics, Web Semantics: Science, Services and Agents on the World Wide Web 8 (2-3) (2010) 182–189.

[37] Sebastian Tramp, Norman Heino, Sören Auer, Philipp Frischmuth, Rdfauthor: employing rdfa for collaborative knowledge engineering, in: Philipp Cimiano, H. Pinto (Eds.), Knowledge Engineering and Management by the Masses, in: Lecture Notes in Computer Science, vol. 6317, Springer, Berlin, Heidelberg, 2010, pp. 90–104. http://dx.doi.org/10.1007/978-3-642-16438-57.

[38] Peter Haase, Andreas Eberhart, Sebastian Godelet, Tobias Mathäß, Thanh Tran, Günter Ladwig, Andreas Wagner, The information workbench. Interacting with the web of data, in: 3rd Future Internet Symposium, FIS2010, September 2010.

[39] T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Pru d'ommeaux, M.C. Schraefel, Tabulator redux: writing into the semantic web, Tabulator Redux Tech Report, 2007.

[40] Michel Buffa, Fabien Gandon, Guillaume Ereteo, Peter Sander, Catherine Faron, Sweetwiki: a semantic wiki, Web Semantics: Science, Services and Agents on the World Wide Web 6 (1) (2008) 84–97. Semantic Web and Web 2.0.

[41] José L. Navarro-Galindo, José Samos, Manual and automatic semantic annotation of web documents: the flersa tool, in: Proceedings of the 12th International Conference on Information Integration and Web-Based Applications & Services, iiWAS'10, ACM, New York, NY, USA, 2010, pp. 542–549.

[42] Benjamin Adrian, Jörn Hees, Ivan Herman, Michael Sintek, Andreas Dengel, Epiphany: adaptable RDFa generation linking the web of documents to the web of data, in: Philipp Cimiano, H. Pinto (Eds.), Knowledge Engineering and Management by the Masses, in: Lecture Notes in Computer Science, vol. 6317, Springer, Berlin, Heidelberg, 2010, pp. 178–192. http://dx.doi.org/10.1007/978-3-642-16438-5-13.

[43] Iván Ruiz-Rube, Carlos M. Cornejo, Juan Manuel Dodero, Vicente M. García, Development issues on linked data weblog enrichment, in: Salvador Sánchez-Alonso, Ioannis N. Athanasiadis (Eds.), Metadata and Semantic Research, in: Communications in Computer and Information Science, vol. 108, Springer, Berlin, Heidelberg, 2010, pp. 235–246. http://dx.doi.org/10.1007/978-3-642-16552-8-22.

[44] Alex Klebeck, Sebastian Hellmann, Christian Ehrlich, Sören Auer, Ontosfeeder—a versatile semantic context provider for web content authoring, in: Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, Jeff Pan (Eds.), The Semanic Web: Research and Applications, in: Lecture Notes in Computer Science, vol. 6644, Springer, Berlin, Heidelberg, 2011, pp. 456–460. http://dx.doi.org/10.1007/978-3-642-21064-8-34.

[45] Timofey Ermilov, Norman Heino, Sebastian Tramp, Sören Auer, Ontowiki mobile–knowledge management in your pocket, in: 8th Extended Semantic Web Conference, ESWC2011, June 2011.

[46] Nadzeya Kiyavitskaya, Nicola Zeni, James R. Cordy, Luisa Mich, John Mylopoulos, Cerno: light-weight tool support for semantic annotation of textual documents, Data Knowl. Eng. 68 (12) (2009) 1470–1492. Including Special Section: 21st IEEE International Symposium on Computer-Based Medical Systems (IEEE CBMS 2008)—Seven selected and extended papers on Biomedical Data Mining.

[47] Daniel Herzig, Basil Ell, Semantic mediawiki in operation: experiences with building a semantic portal, in: Peter Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Pan, Ian Horrocks, Birte Glimm (Eds.), The Semantic Web—ISWC 2010, in: Lecture Notes in Computer Science, vol. 6497, Springer, Berlin, Heidelberg, 2010, pp. 114–128. http://dx.doi.org/10.1007/978-3-642-17749-1-8.

[48] Soren Lauesen, User Interface Design: A Software Engineering Perspective, Addison Wesley, 2005.

[49] Jakob Nielsen, Introduction to usability, 2012.

[50] Ahmed Seffah, Mohammad Donyaee, Rex B. Kline, Harkirat K. Padda, Usability measurement and metrics: a consolidated model, Softw. Qual. Control 14 (2) (2006) 159–178.

[51] Ferran Perdrix, Roberto García, Rosa Gil, Marta Oliva, José A. Macías, Semantic web interfaces for newspaper multimedia content management, in: New Trends on Human–Computer Interaction, Springer, London, 2009, pp. 1–10. http://dx.doi.org/10.1007/978-1-84882-352-53.

[52] Ali Khalili, Sören Auer, Daniel Hladky, The RDFa content editor—from WYSIWYG to WYSIWYM, in: IEEE Signature Conference on Computers, Software, and Applications, COMPSAC 2012, 2012.

[53] M. d'Aquin, E. Motta, M. Dzbor, L. Gridinoc, T. Heath, M. Sabou, Collaborative semantic authoring, IEEE Intell. Syst. 23 (3) (2008) 80–83.

[54] Dhiren R. Patel, Sidheshwar A. Khuba, Realization of semantic atom blog, CoRR, arXiv:abs/0912.3957, 2009.

[55] Ronan Fitzpatrick, Strategies for evaluating software usability, Methods 353 (1) (1998).

[56] Lianping Chen, Muhammad Ali Babar, A systematic review of evaluation of variability management approaches in software product lines, Inf. Softw. Technol. 53 (4) (2011) 344–362.

[57] Norman Heino, Sebastian Tramp, Sören Auer, Managing web content using linked data principles—combining semantic structure with dynamic content syndication, in: Proceedings of the 35th Annual IEEE International Computer Software and Applications Conference, COMPSAC 2011, IEEE Computer Society, 2011.

[58] Steffen Lohmann, Philipp Heim, Sören Auer, Sebastian Dietzold, Thomas Riechert, Semantifying requirements engineering—the softwiki approach, in: Proceedings of the 4th International Conference on Semantic Technologies, I-SEMANTICS'08, J. UCS, 2008, pp. 182–185.

[59] Thomas Riechert, Ulf Morgenstern, Sören Auer, Sebastian Tramp, Michael Martin, The catalogus professorum lipsiensis—semantics-based collaboration and exploration for historians, in: Proceedings of the 9th International Semantic Web Conference, ISWC2010, in: Lecture Notes in Computer Science, Springer, Shanghai, China, 2010.

[60] Jussi Kurki, Eero Hyvönen, Collaborative metadata editor integrated with ontology services and faceted portals, in: 1st Workshop on Ontology Repositories and Editors for the Semantic Web, 2010.

[61] Bo Yu, Cognitive aspects of human-gis interaction: a literature review, Interface (2006) 1–17. Cognitive aspects of human-gis interaction: a literature review.

[62] Gilberto Camara, Ricardo Cartaxo, Modesto Souza, Antonio Miguel Monteiro, Joao Paiva, Juan Garrido, Gilberto Câmara, Ricardo Cartaxo, Modesto De Souza, Antônio Miguel, Vieira Monteiro, Juan Carlos, Pinto De Garrido, Divisão Processamento, Imagens Dpi, Handling complexity in gis interface design, in: Proceedings of the I Brazilian Workshop on GeoInformatics, Campinas, São, 1999.

[63] Jakob Nielsen, Rolf Molich, Heuristic evaluation of user interfaces, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Empowering People, CHI'90, ACM, New York, NY, USA, 1990, pp. 249–256.

[64] Estelle de Kock, Judy Van Biljon, Marco Pretorius, Usability evaluation methods: Mind the gaps, Evaluation (2009) 122–131.

[65] Jeff Howe, The rise of crowdsourcing, Wired Magazine 14 (6) (2006) 06.

[66] David Geiger, Michael Rosemann, Erwin Fielt, Crowdsourcing information systems: a systems theory perspective, in: Australasian Conference on Information Systems, ACIS 2011, Sydney, Australia, 2011.

[67] Adam Greenfield, Everyware: The Dawning Age of Ubiquitous Computing, New Riders Publishing, Berkeley, CA, 2006.

[68] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, Andrew T. Campbell, A survey of mobile phone sensing, 2010.