Submitted on: March 15, 2013

Semantically Guided, Situation-Aware Literature Research

Timm Heuss
Darmstadt University of Applied Sciences, Darmstadt, Germany.
timm.heuss@h-da.de

Bernhard Humm
Darmstadt University of Applied Sciences, Darmstadt, Germany.
bernhard.humm@h-da.de

Tilman Deuschel
Darmstadt University of Applied Sciences, Darmstadt, Germany.
tilman.deuschel@h-da.de

Torsten Fröhlich
Darmstadt University of Applied Sciences, Darmstadt, Germany.
torsten.froehlich@h-da.de

Thomas Herth
media transfer AG, Darmstadt, Germany.
therth@mtg.de

Oliver Mitesser
University and State Library, Darmstadt, Germany.
mitesser@ulb.tu-darmstadt.de

Abstract:

Searching literature in bibliographic portals is often a trial and error process, consisting of manually and repeatedly firing search requests. This paper presents an automatic guidance system that supports users in their literature research with a stepwise refinement process, based on their needs. Algorithms operate on various different data sets, including valuable world knowledge in form of Linked Open Data, which is streamlined and indexed by a Semantic Extraction, Transform and Load process. Search results are composed dynamically and are visualized in an innovative way as a topic wheel. In this paper, we describe the prototype of this system and our current work in progress, including first user evaluations.

Keywords: Linked Open Data, Semantic Information Visualization, Library Guide and Search System, Information Retrieval, HTML5

1 INTRODUCTION

In 2009, Tim Berners-Lee introduced Linked Open Data (LOD) as a paradigm to exchange data in order to allow others to reuse and to refer to this data without barriers (Berners-Lee, 2009). Since then, many governments[1] and organizations[2] have understood the LOD-vision as a public service providing a new level of transparency and launched portals that offer most different data sets in LOD, to anyone who is interested.

Publishing data that everybody can use is a great benefit for a citizen. However, the pure availability does not necessarily lead to new and exciting applications. In fact, advanced engineering and domain specific skills are required to build applications that benefit from LOD.

In the project Mediaplatform, we are researching new and enhanced ways of searching and displaying media stocks, for example books in a library. Thereby, we observed that a classic book search is often a trial and error process, consisting of firing a search request, receiving either too many or too few results and then manually finding a suitable refinement or generalization of the search terms. We think that, today, there is a chance to perform literature research much more intelligently.

Compare the above mentioned trial and error process with the way well informed human librarians recommending suitable literature to a customer. Librarians would usually not wait for customers to refine their book requests. Instead, they would rather ask questions in order to get a better understanding of the customers' needs, make preselections and recommendations. Customers often only roughly describe what they are looking for. With the competent help of librarians, they will find the intended literature - or other alternatives that meet the requirements much better. There would be no trial and error but a systematically guided, stepwise refinement process.

To pursuit the vision of this human-like guidance, we develop an application that supports library users semantically in their literature research, i.e., based on the understanding of their search inputs. It has to cope with different situations, either if there are no, too many or too few results.

The remainder of this paper is organized as follows. In Section 2, we define the different challenges such an application is faced with. In the Section 3, we first clarify the role of LOD, and then we introduce the application's logical components. In the Section 4, we present technical insights. Section 5 evaluates the approach in consideration of the criteria from Section 2. Section 6 presents related work. Section 7 concludes the paper and outlines future work.

[1] A famous LOD government portal is, for example, the British data.gov.uk (last accessed 2013-05-06).

[2] For example, the Project Gutenberg dataset, `http://datahub.io/dataset/fu-berlin-project-gutenberg` (last accessed 2013-05-06)

2 PROBLEM STATEMENT

The goal of this work is to enable semantically guided, situation-aware literature research via an application. The requirements are as follows.

1. Literature research: It shall be possible to find and retrieve literature, e.g., books, articles, or monographs, which are relevant[3] to the user. Users may be students or researchers.
2. Semantically guided: The application shall assist users to find relevant literature[4], i.e., by guiding the users. The guidance shall be based on knowledge about the users' interests and about the semantic content of the literature. Users shall have the impression that the application understands them similarly to human librarians. The guidance shall be goal-oriented, i.e., guide the user as quickly as possible to relevant literature.
3. Situation-aware: The application shall be aware of the situation of the user and of the retrieval process and adapt the guidance strategy accordingly. For example, if the user provides a general search term with too many search results, the application suggests sensible specializations. On the other hand, if the search terms are too specific to find any research result, the application suggests related terms.
4. Intuitive: Users shall be able to use the application without training or studying a user manual.
5. Device-independent: The application shall be useable on all state-of-the-art devices, i.e., mobile phones, tablet computers, and personal computers including touch and pointer interaction, even at the same time.
6. Good performance: The application shall allow users to work in their own pace in a pleasant way. In particular, response times shall be below 1 sec. for common use cases.

3 AN APPLICATION FOR SEMANTICALLY GUIDED, SITUATION-AWARE LITERATURE RESEARCH

3.1 The Role of LOD

The concept of Linked Open Data (LOD) is most important for the success of endeavors like the one described here. This is for the following reasons:

1. Large communities formalize knowledge in very large data sets, which would be impossible for individual development projects. This includes quality assurance and permanently keeping the data sets up to date.
2. Standardized formats allow processing the data sets.
3. The free use of the data sets is ensured by appropriate licenses.

However, there are major challenges in using such data sets in applications as described here. Bowker and Star (1999) correctly state: "Classifications that appear natural, eloquent, and homogeneous within a given human context appear forced and heterogeneous outside of that context". This may result in the following problems.

1. Structure and content of the data sets do only partially match the requirements of an application. Take for example an application that needs to provide the epoch in which

[3] A first survey we conducted showed that relevant information for making a decision which book to borrow is the table of content, description and title.

[4] The survey showed that users know what they want to know before they start their search, but that it is more difficult to find appropriate key words.

an artist worked, e.g., Michelangelo worked in the epoch Renaissance. The YAGO Ontology (Suchanek, Kasneci & Weikum, 2007] contains this information, but not explicitly via a property "epoch" but implicitly via the property "rdf:type" with the value "wikicategory_Rennaissance_artists". This is also a key problem when implementing a search mechanism across different vocabularies [Schreiber et al. 2008].
2. Structure and content of different data sets to be integrated are only partially compatible. For example, in medicine ontologies like the NCI thesaurus (Golbeck et al., 2003), concepts are modelled as OWL classes and the relation with broader concepts are modelled via "rdfs:subclassOf". Other ontologies model concepts as instances and the relation with broader concepts explicitly via "skos:broader" [SKOS]. There are no uniform queries over such ontologies when they are simply merged.
3. The structure of data sets does not allow efficient query access in certain situations. For example, Heuss (2013) describes a conceptually simple query to the SPARQL endpoint of DBpedia which was rejected with error message: "The estimated execution time 7219 (sec) exceeds the limit of 3000 (sec)". The reason being that the structure of DBpedia is not optimized for this kind of query.

To leverage the advantages of LOD while coping with the problems described, we utilize a "Semantic Extraction, Transformation, and Loading (Semantic ETL)" approach.

3.2 Semantic ETL

Semantic ETL consists of the steps (1.) *extraction*, (2.) *transformation / semantic enrichment*, and (3.) *loading*.

3.2.1 Extraction

In the first step, data sources are acquired which have the following characteristics:

1. *Relevant*: The data sources must contain information which is relevant for the respective use cases, here semantically-guided literature research. For example, the GND[5] data set of the German National Library contains a taxonomy of subject terms which allows finding broader and narrower terms for a given term.
2. *Sufficient quality*: The information contained in the data sources must be of sufficient quality. For example, GND has been manually edited and quality controlled by experts over a period of several decades.
3. *Structured*: The information is provided in a structured, machine-processable way. For example, GND is provided in RDF. Any other structured, documented format like, e.g., Pica+[6] is suitable, too.
4. *Accessible*: GND is accessible as linked open data. However, commercially available data source are suitable, too.

The careful selection of suitable data sources is a design-time step to be performed by human experts. During execution time, the extracted data sources are stored in a staging area.

[5] `http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html` (last accessed 2013-05-07)

[6] German description of data format `http://www.gbv.de/wikis/cls/PICA-Format` (last accessed 2013-05-15)

3.2.2 Transformation and semantic enrichment

In the second step, the data sources are pre-processed, including the following sub-steps.

1. *Format transformation*: In this step, the formats of the different data sources are transformed into a common, structured data format, e.g., in JSON notation[7]. Source formats are XML dialects, RDF, CSV, etc.
2. *Semantic enrichment*: In this step, heuristics are used in order to deduce additional information from the data sources. For example, subject terms in GND lack a ranking with respect to their relevance. So, the term `Java <Programming language>` has numerous narrower terms, including more relevant ones like `Java Enterprise Edition` and less relevant ones like `Visual J++`. With the heuristic that the relevance of a term increases with the number of publications with this term as subject, a term ranking can be derived from the publication stock. More complex semantic enrichment necessitates the matching and merging of corresponding entities from different data sources via heuristic methods.
3. *Performance tuning*: In some cases, it is advantageous to denormalize data and to influence the indexing process for high-performance access.

3.2.3 Loading

In the last step, the transformed and semantically enriched data is loaded into a data store. The data store must provide the following characteristics:

1. *Sufficient query functionality*: e.g., it should be possible to query for literature which has a given term as subject, including narrower terms, ignoring case, and allowing for phonological ambiguities (e.g., "Meyer" versus "Maier").
2. *Sufficient performance*: The data store must allow for high-performance access to large data sets.

3.3 Retrieval

The Retrieval component delegates incoming search queries to the involved data stores, which have been previously filled by the Semantic ETL process. There are three stores:

1. *The DocumentStore* contains meta data about the media entities, for example books.
2. *The AuthorStore* contains meta data for persons that are connected to the media entities in the DocumentStore. For books, this connection is usually an authorship.
3. *The TermStore* contains general purpose, cross-domain technical terms and their senses, stored in a hierarchical structure. Thus, for a book, not only its respective category can be found as an entry, but also broader or narrower categories.

The following two sections describe a typical search scenario and how the different stores are involved. Store results are never directly returned to the user, they are just collected as evaluation input for the next component, the Guiding Agent.

3.3.1 AuthorStore and DocumentStore Search Capabilities

A typical search query that involves the AuthorStore as well as the DocumentStore might be the query `Russel Norvig Artifical Intelligence`. After this query is delegated by the Retrieval component, the DocumentStore returns, among others, `Artifical`

[7] `http://www.json.org/` (last accessed 2013-05-15)

`Intelligence – A Modern Approach` and the AuthorStore the according authors `Stuart Russel` and `Peter Norvig`.

3.3.2 TermStore Search Capabilities

For the TermStore, a typical input might be the single word `Java`. Thanks to the legwork done by Semantic ETL, the store finds three senses for `Java` - the island, the dance and the programming language - as well as broader and narrower concepts for each respective sense, e.g. `Object oriented programming languages` as broader concept of Java in the sense `Programming Language` or `Jakarta` as narrower concept of `Java` in the sense of `Island`.

3.4 Guiding Agent

The component providing the guiding logic follows an agent approach. For each user interaction step, it receives the current user action as input, and produces information for the user. Thereby, it performs the following steps: (1.) *accept user input*, (2.) *search data stores*, (3.) *analyse situation / react accordingly*, (4.) *process and return result*.

3.4.1 Accept User Input

The user can interact with the application in the following ways.

1. *Free text input*: The user may specify criteria in a text entry box for the literature research in natural language including, e.g., mentioning authors, subject terms, publication titles or parts thereof, publication dates, etc.
2. *Selected topics*: The user may select topics presented by the application in the previous interaction step.

3.4.2 Search Data Stores

The user input is utilized to retrieve relevant data from the data stores: AuthorStore, TermStore, and DocumentStore. The data store queries are generated depending on the user input as explained in the previous section.

3.4.3 Analyse Situation and React Accordingly

The agent analyzes the retrieved data and uses heuristics to deal with different situations. Examples:

1. If the user input results in a large number of matching documents then the agent will guide the user to *refine* the request. To this end, topics are being generated that partition the result set.
2. If the user input results in no matching documents the agent will guide the user to *broaden* the request. To this end, topics are being generated that are related with the user request but will result in matching documents.

3.4.4 Process and Return Result

The agent's result returned to the user consists of a list of documents and a set of topics that may guide to the next step of literature research.

1. *Documents* are ranked according to the degree of matching the user input. Only sufficiently relevant documents are selected for the result - the remaining documents are filtered out.
2. *Topics* are generated using heuristics. For example, to refine a request, terms which are narrower than the terms specified by the user may be retrieved from the TermStore. Those narrower terms are considered relevant if they have relevant, matching documents. Such relevant terms can be prioritized and offered to the user as further topics.

3.5 Client

The application is a combination of a search-oriented system, providing books that are relevant to the user's request, and a browsing-oriented system, enhancing the search by the Guiding Agent's adaptive navigation support.

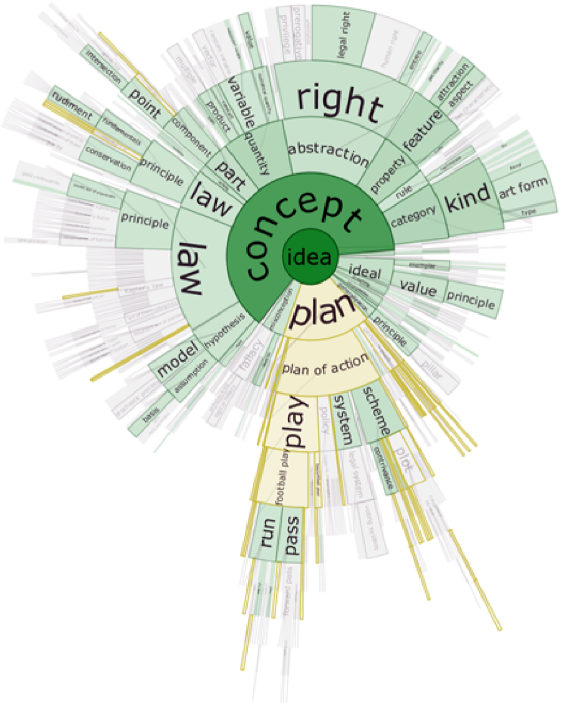
The adaptive interface is environment-aware and renders differently on different screen resolutions (Brusilovsky, 2001). The client uses CSS media queries[8] to change layout and style for three different device classes: smartphone, tablet and desktop. A state machine determines which interactor shall be rendered. Thereby the application becomes faster on mobile devices by avoiding to render all the interactors at first and hide the not required interactors afterwards. The state machine also enables graceful degradation of interactors (Vanderdonckt & Florins, 2004), based on different states. The guiding agent's visualisation is rendered on a smartphone in landscape mode differently than on a smartphone in portrait mode, because the way users interact is different then, though the screen space stays the same (Nicolau & Jorge, 2012).

The Guiding Agent's visualization is inspired by Christopher Collins, who worked in the area of semantic information visualization research. Especially the document content visualization DocuBurst is important for our work (Collins, 2009).

DocuBurst visualizes the content of documents by breaking it down to topics. These topics are represented in circular arcs, showing the superordinated topic in the center. The subordinated topics are ordered towards the edge.

Because the guiding agent's visualization looks like a wheel and can be spinned, we call it the topic wheel. In contrast to DocuBurst, the topic wheel is an interactor that does not visualize all the information available but enables the user on touch and non touch interfaces to easily select recommended topics based on the current topic. Regarding desktop computers and convertibles, the user interface needs to satisfy touch and pointer input at the same time.

[8] `http://www.w3.org/TR/css3-mediaqueries/` (last accessed 2013-05-27)



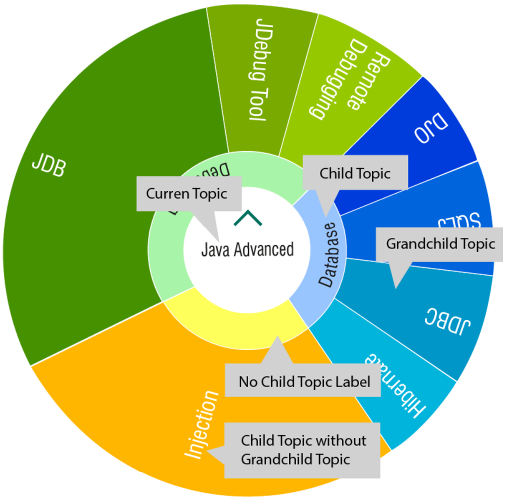
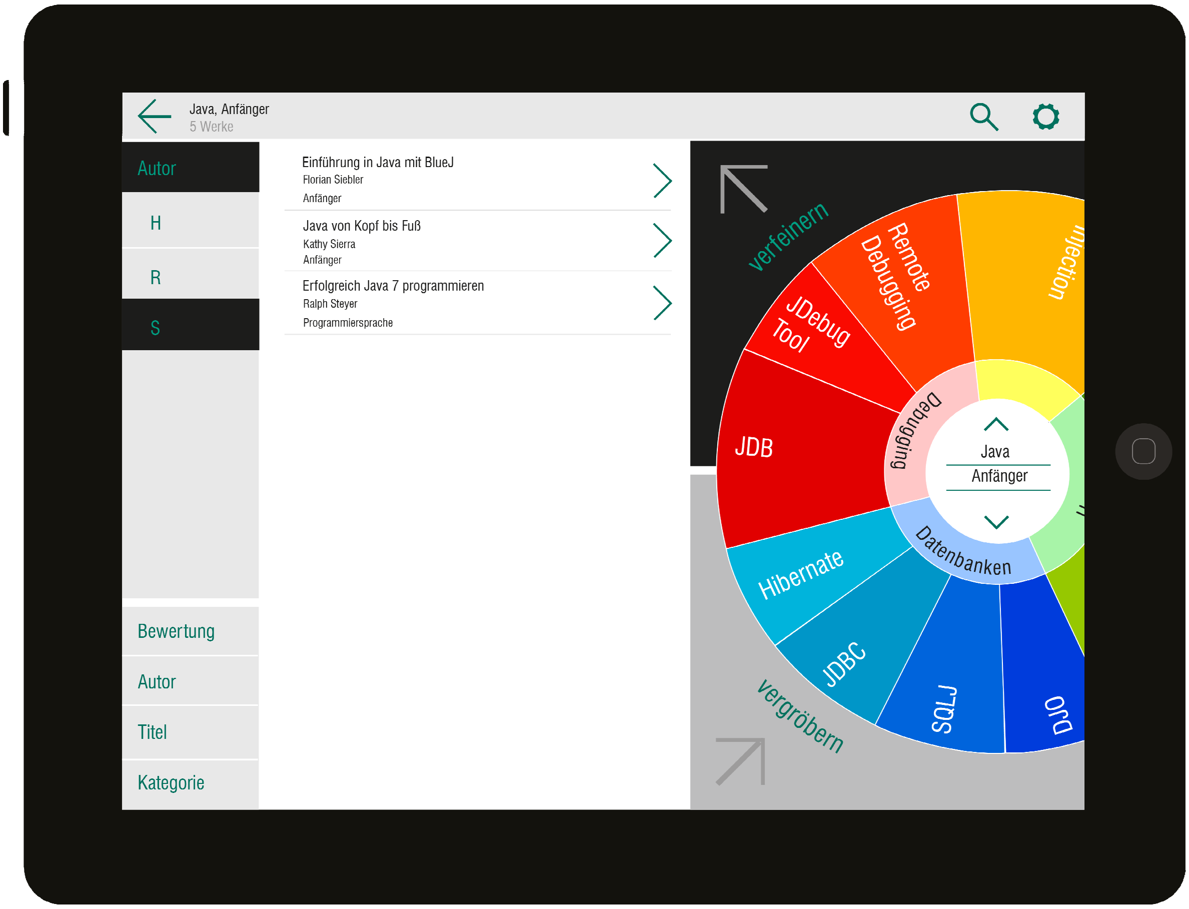


Figure 1a: On the left: DocuBurst visualization of a document, by Christopher Collins.
Figure 1b: On the right: topic wheel, visualization of the content composed by the Guiding Agent.

The topics are ordered from center to edge displaying a top-down hierarchy as in DocuBurst but showing only two degrees of relationship based on the current topic. Directly related topics are visualized as child topics between the center and the edge. If a child topic has further subordinated topics, they are the grandchildren of the current topic and placed at the edge. The total amount of child topics or grandchild topics is limited to 25. This way we can assure that the label of each topic inside the topic wheel is still readable and that the topic wheel does not become too complex for a user whose primary goal is a quick guidance in addition to the search-driven process.

The current topic is a search term, provided by the user. If there is no grandchild topic but only child topics, they are displayed as grandchild topics without a child topic label (see Fig. 1b). This could be regarded as a logical inconsistency. However, early usability tests indicated that users understand the Guiding Agent better by visual consistency, i.e., not having gaps in the circular visualization. Therefore, the child topic label is placed at the edge. This also solves the problem of lacking screen space if many child topics without grandchild topics are supposed to be rendered in the topic wheel.

The range of color is based on the light spectrum from blue to red and finally violet. We place violet at the end, so the topic wheel will start with blue, followed by green, which creates more aesthetic visualizations as most users have a preference for blue and green (Deuschel & Vas, 2012).



Java, Anfänger
5 Werke

Einführung in Java mit BlueJ
Florian Siebler
Anfänger

Java von Kopf bis Fuß
Kathy Sierra
Anfänger

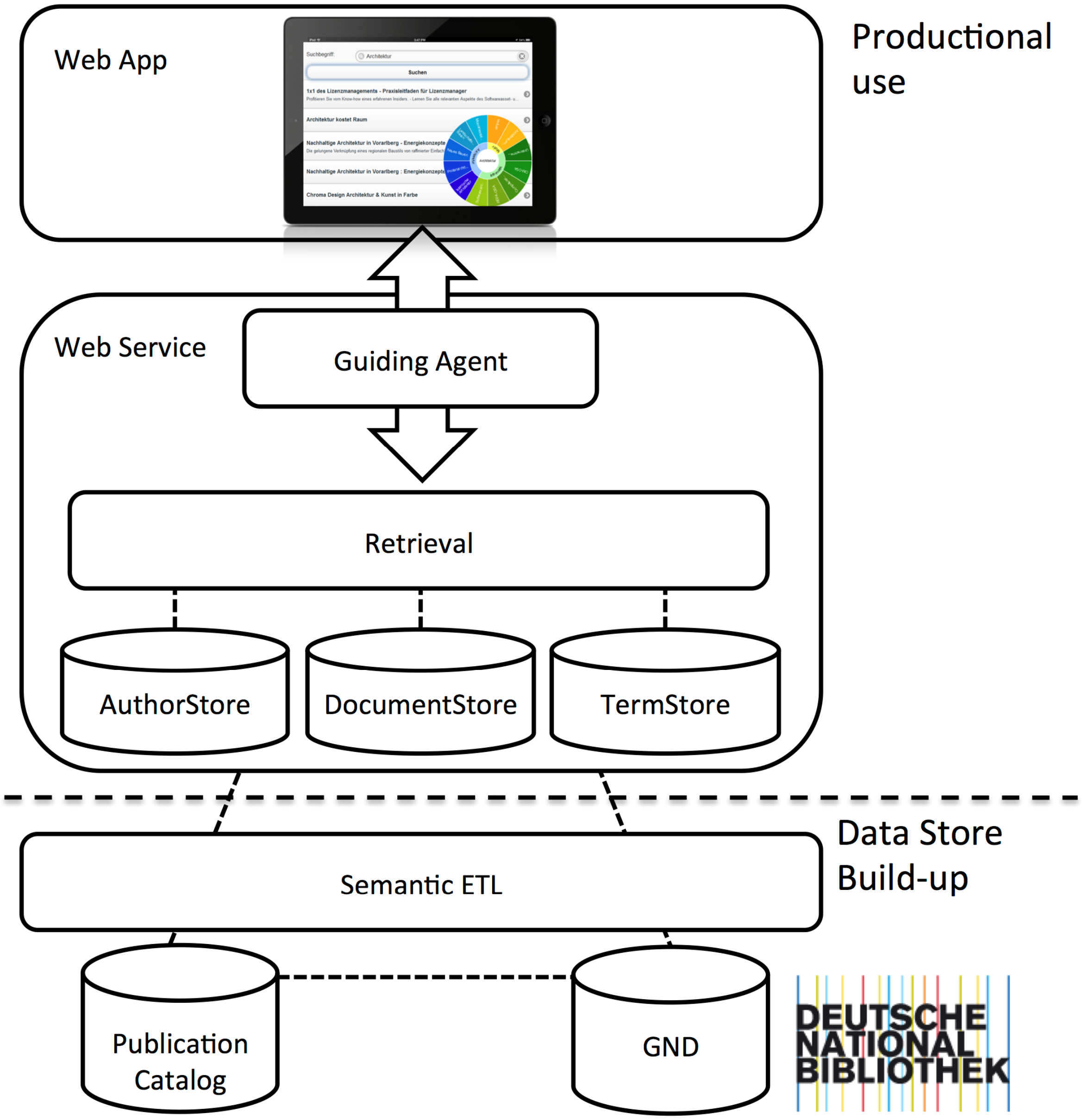
Erfolgreich Java 7 programmieren
Ralph Steyer
Programmiersprache

Figure 2: Adaptive interface of the application rendered for a tablet environment. The topic wheel is placed on the right.

25 shades of each color represent a grandchild topic. Every shade has a brighter and less saturated version for the child topic. That way the topic wheel is able to display every scenario from one child topic and 25 grandchild topics up to 25 child topics without grandchild topics. The topic wheel does not aim for complete representation of all available related topics, it displays the most relevant ones only. Therefore, it is necessary to sort them according to relevance.

4 IMPLEMENTATION

The Mediaplatform implementation is a client-/server architecture, consisting of an HTML5 Web App that communicates with an HTTP/REST Web service written in Java. The following figure depicts the involved layers and the corresponding components Guiding Agent, Retrieval and the Semantic ETL.



Suchbegriff: Architektur

Suchen

1x1 des Lizenzmanagements - Praxisleitfaden für Lizenzmanager
Profitieren Sie vom Know-how eines erfahrenen Insiders. - Lernen Sie alle relevanten Aspekte des Softwareasset- u...

Architektur kostet Raum

Nachhaltige Architektur in Vorarlberg - Energiekonzepte
Die gelungene Verknüpfung eines regionalen Baustils von raffinierter Einfach

Nachhaltige Architektur in Vorarlberg : Energiekonzepte

Chroma Design Architektur & Kunst in Farbe

Figure 3: Architectural overview of the Mediaplatform, showing the components involved in the two modes "Data Store Build-up" and "Productional use"

The figure also shows that there are two main modes: (1.) data store build up and (2.) productional use.

4.1 Data Store Build-up

As a preparatory step, the Semantic ETL component is executed to build up the AuthorStore, DocumentStore, and TermStore. It processes various kinds of inputs, including the GND data set of the German National Library[9] (RDF format) and the document stock of all libraries of the state of Hesse (Pica+ format).

[9] `http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html` (last accessed 2013-05-07)

As outlined in Section 3.2, the criteria for selecting the data store technologies and products are (1). sufficient query functionality and (2.) sufficient performance. For retrieving literature, unsharp querying features such as case-insensitive search, allowing for phonological ambiguities, best guesses, support for lemma forms, and ratings are required. RDF triple stores such as Jena or OWLim do not provide such features. Such features are typically provided by search engines like Apache Lucene. Lucene also provides the necessary performance characteristics and has been chosen as data store technology.

The Semantic ETL process creates the stores as Lucene index structures. Lucene offers the suitable abstraction level to search engine interns but still gives the opportunity to manipulate delicate details, e.g., the exact kind of indexing of certain information. We use this to optimize the index structures towards the typical usage patterns of the Mediaplatform. For example, to create the contents of the topic wheel, the TermStore must find a given term with its corresponding broader and narrower concepts out of half a million records within a few ten milliseconds, as the subsequent Guiding Agent will conduct a time-consuming reasoning and optimization step prior to the final delivery to the client. The terms (subjects) in the input source GND, however, just refer to their respective broader terms, so the transitive relation to narrower terms is created during the Semantic ETL. It takes a few minutes to build up the index but in production, the transitive relations can then be accessed in about 20 milliseconds.

4.2 Productional Use

In productional use, the Retrieval component fires queries on all three Lucene index structures, the AuthorStore, DocumentStore and the TermStore. The resulting entities from the stores are then processed in the Guiding Agent component, which conducts several heuristics, for example a prioritization or a preselection. We designed this reasoning of the Guiding Agent as an isolated step from the store queries, not just to have results from three stores as input for the reasoning, but also to incorporate statistics or to preview store results, and to take this valuable information in consideration, too.

On the client side, the data sets sent from the Web service are displayed by an HTML5 Web App. Due to the use of CSS media queries, it may be used on various devices, especially on mobile phones and tablets. We also use a number of common JavaScript frameworks, like jQuery mobile[10], in order to respect the individual needs of the top three browsers Firefox, Chrome and Safari. We use Knockout[11] to implement a Model-View-ViewModel (MVVM) pattern and KineticJS[12], to build the canvas-based topic wheel which is rotated by CSS3 transitions.

5 EVALUATION

In this section, we evaluate our approach with respect to the goals outlined in section "Problem Statement".

1. *Literature research*: The current prototype implementation allows users to find literature in the consolidated stock of all university libraries of the German state of Hesse including, overall, about 15 million publications.
2. *Semantically guided*: The guiding agent component implements logic to assist the user in finding relevant literature, i.e., by guiding the user. This guidance is based on

[10] `http://jquerymobile.com/` (last accessed 2013-05-07)

[11] `http://knockoutjs.com/` (last accessed 2013-05-07)

[12] `http://kineticjs.com/` (last accessed 2013-05-07)

user input and on content of the data store, established by the Semantic ETL process and is employed in the current prototype. First analyses of user logs suggest the successful goal-oriented guidance of users to relevant literature. However, more usage data needs to be collected and analyzed systematically before we can make statements whether the user is guided as quickly as possible. Also, user surveys are necessary to evaluate whether users have the impression that the application understands them similarly to a human librarian. Those evaluations are future work.

3. *Situation-aware*: The guiding agent component provides the logic to assess the situation of the user and adapts the guidance strategy accordingly. For example, if the user provides a general search term with too many search results, the application suggests sensible specializations. On the other hand, if the search terms are too specific to find any research result, the application suggests related terms.
3. *Intuitive*: First experiences with the prototype suggest that users are able to use the application without training or studying a user manual. However, systematic user surveys will be needed to evaluate this aspect.
4. *Device-independent*: The current prototype implementation is usable on most state-of-the-art devices and platforms, in particular Nexus 10 (Android 4.2.2 Tablet), Samsung Galaxy S3 (Android 4.1.2 Smartphone), iPad4 (iOS 6.1.3 Tablet), iPad3 (iOS 5.1.1 Tablet) and iPhone 4 (iOS 6.1 Smartphone).
5. *Good performance*: The current prototype implementation exhibits a response time below 1 sec. for common use cases.

6 RELATED WORK

6.1 Question Answering

(Ferucci et al., 2010) gives an overview of IBM’s DeepQA Project and Watson. Watson is a Question Answering (QA) application which defeated human champions in the American TV quiz show Jeopardy. Due to the real-time requirements, particular attention is given to an architecture allowing for high performance. Watson uses numerous data sources including, for example, the YAGO ontology. Similar to our approach of semantic ETL, the data sources are pre-processed and loaded into the system in an offline process to allow for high-performance online-access. While the goals and capabilities of Watson are far beyond the ones described here, we regard this as a confirmation of our Semantic ETL approach.

6.2 Search Result Clustering

The topic described in this paper is related to search result clustering. Because we exploit meta data, the initial task of “discovering subject of objects” (Carpineto, Osiński, Romano & Weiss, 2009, page 5) is a much more straightforward process, however, visualisation is challenging in both disciplines. In search result clustering, a hierarchical “folder layout” is usually employed (Carpineto et al., 2009, page 16), while other, graphically sophisticated alternatives usually suffer usability problems (Carpineto et al. 2009, page 18). Instead, with the topic wheel, we developed a novel approach of displaying search result clusters with a high usability.

6.2.1 ETL Component

Many media stock projects describe an ETL-component, including those which employ a pure Semantic Web technology stack, because “in practice, many data sources still use their own schema and reference vocabularies which haven’t been mapped to each other” (Mäkelä, Hyvönen & Ruotsalo, 2012). For example, (Schreiber et al., 2008) describes a considerable effort in the “harvesting, enrichment and alignment process of collection metadata and vocabularies”, which takes about “1-3 weeks to include a new [media] collection”. This time is comparable with the integration effort of the Semantic ETL process, too.

6.3 Query Expansion

We currently do not employ a classic query expansion mechanism, even though others like (Ruotsalo, 2012) made very good experiences in comparable problem domains. However, with the Guiding Agent, we have a similar mechanism that modifies queries, e.g. to refine a certain search request. In contrast to the classic query expansion, our approach is based on heuristics operating on result sets. Admittedly, we see some potential in query expansion in our future work.

6.4 Adaptive Guides

According to (Brusilowsky, 2001) the landscape of adaptive guides and adaptive recommendation systems can be divided into closed-web and open-web systems. Our application is a closed-web system, it displays only content which is directly linked by authory files and library information systems such as HeBiS[13].

The Guiding Agent makes suggestions, similar to Letizia (Liebermann, 1995) and SiteIF (Stefani & Strapparava, 1999). SiteIF is a closed-web and Letizia an open-web recommendation system. Letizia estimates the value of a certain link for the searching user and recommends them to the user. The recommendations are preference ordered and computed regarding the persistence of interest phenomenon (Liebermann, 1995) and can even react on serendipitous connections, which is a major goal in browsing-driven behaviour. Letizia uses information retrieval as well as information filtering. Both systems do not ask the user for keywords of interest, but create a user model based on the visited pages. Unlike them, the Guiding Agent does not employ user models, because the combination of search-oriented and browsing-oriented systems implies that the user provides the Guiding Agent with a keyword of interest, the search term. The recommendations are then made based on the structure provided by the authority files. However we see potential in user modeling in our future work.

6.5 Semantic Visualization

As mentioned, Collins (2009) has major influence on our work, with the difference that Collins’ DocuBurst visualizes the complete content of a document enabling the user to compare large amounts of text at once. DocuBurst uses WordNet (Fellbaum, 1998) a lexical database, to match synonyms in a text and to detect their relationships. It visualizes structure and counts of word occurrences in a document using a sunburst diagram, which displays hierarchy and quantity distribution in circular arcs. In contrast to DocuBurst, the topic wheel described in this paper is an interactor - therefore usability weights more than utility. It limits the displayed information and appears differently on landscape and portrait mode as well as on different device classes like smartphones and tablets.

[13] `http://www.hebis.de/eng/englisch_index.php` (last accessed 2013-05-07)

7 CONCLUSIONS AND FUTURE WORK

Releasing Linked Open Data (LOD) is currently very much in the spirit of the age. While this paradigm allows developers to incorporate world knowledge in an as yet unprecedented scale into their applications, building complex knowledge-based systems is still a challenging task.

In pursuit of recreating the utility and usability provided by of a human librarian, we introduce the Mediaplatform application for situation-aware, semantically guided literature research. The application helps users to find and retrieve books, assists them by understanding their needs, and provides recommendations. The application is intuitive and shows a good performance across various different devices and platforms.

We showed that a number of different data sources and formats are to be considered. We also showed how to tackle format-specific issues and data consolidation challenges with a Semantic ETL process.

The so created storage structures are the foundation of the Guiding Agent component, which intelligently composes query results and suggestions with the help of various heuristics. Results are displayed by a HTML5 client with an innovative, rotatable topic wheel.

Early evaluations indicate that we are on the right track. The storage can cope with the meta data of about 15 million publications and first algorithms guide users to relevant literature with the help of the automatically composed topic wheel. Early user tests also show that the application can be used easily, on most state-of-the-art devices and platforms, with a good performance.

With the research and development described in this paper, we did a first important step in establishing core concepts and creating a powerful and sustainable architecture of the Mediaplatform. The prototype implementation is a proof-of-concept.

We plan to extend the functionality of components and introduce new ones. We see potential in preprocessing the user input with methods of Natural Language Processing (NLP). Thereby, entities like book titles or authors could be identified. This could then be used to conduct a query expansion, as described by (Ruotsalo, 2012), e.g. to find terms across different languages.

In a continuous process, we will enrich our data stores with new data sets extending the knowledge-driven guidance, as well as we will adjust the underlying heuristics. Employing a persistent user-model, as described in (Roes, Stash, Wang & Aroyo, 2009), could also significantly improve the guidance mechanisms.

We will continue to constantly conduct feedback loops and usability tests with selected users, as well as we plan to systematically evaluate their expectations, experiences, and usage patterns.

We aim at transforming the research prototype towards a commercial product within the next years.

8 ACKNOWLEDGEMENTS

Our work is funded by the LOEWE initiative of the German state of Hesse under contract number HA 321/12-11. Cooperating partners are the University and State Library Darmstadt as customer as well as Software AG, media transfer AG, nterra GmbH and the House of IT.

9 REFERENCES

Berners-Lee, T. (June 2009). Linked Data [Webpage]. Retrieved from: http://www.w3.org/DesignIssues/LinkedData.html.

Bowker, G. C. & Star, S. L., (1999). *Sorting Things Out: Classification and its consequences*. Cambridge: MIT Press.

Brusilovsky, P. (2001). Adaptive Hypermedia. In *User Modeling and User-Adapted Interaction, 11*, 87-110. doi:10.1023/A:1011143116306

Carpineto, C., Osiński, S., Romano, G., & Weiss, D., (2009). A survey of Web clustering engines. *ACM Computing Surveys, 41*(3). doi:10.1145/1541880.1541884

Collins, C., Carpendale, S. & Penn, G. (2009). DocuBurst: Visualizing Document Content Using Language Structure. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization (EuroVis '09)), 28*(3), 1039-1046.

Deuschel, T. & Vas, R., (2012). *Das Hörspielbrett: Produktion und Feldtest eines nutzerzentrierten, interaktiven Hörspiels für Kinder im Alter von 8-11 Jahren* [unpublished master thesis]. Darmstadt: Hochschule Darmstadt University of Applied Sciences.

Fellbaum, C. (ed.) (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA, 1998.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., … Welty, C. (2010): Building Watson: An Overview of the DeepQA Project. In *AI Magazine, 31*(3), 59-79. Association for the Advancement of Artificial Intelligence, 2010.

Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J., & Parsia, B. (2003). The National Cancer Institute's Thesaurus and Ontology. *Web Semantics: Science, Services and Agents on The World Wide Web, 1*(1), 75-80. doi:10.1016/j.websem.2003.07.007.

Heuss, T. (2013). Lessons learned (and questions raised) from an interdisciplinary Machine Translation approach. Position paper for the W3C Workshop on the Open Data on the Web, 23 - 24 April 2013, Google Campus, Shoreditch, London [PDF]. Retrieved from: http://www.w3.org/2013/04/odw/odw13_submission_18.pdf

Liebermann, H. (1995). Letizia: an agent that assists web browsing, Bd. 1. In *IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1* (924–929). San Francisco: Kaufmann.

Mäkelä, E., Hyvönen, E. & Ruotsalo, T. (2012). How to deal with massively heterogeneous cultural heritage data – lessons learned in CultureSampo. *Semantic Web 3*, 85–109. doi:10.3233/SW-2012-0049

Nicolau, H. & Jorge, N. (2012). Touch typing using thumbs: understanding the effect of mobility and hand posture. In *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2683–2686. doi:10.1145/2207676.2208661

Roes, I., Stash, N., Wang, Y., & Aroyo, L. (2009). A personalized walk through the museum: the CHIP interactive tour guide. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 3317–3322. doi:10.1145/1520340.1520479

Ruotsalo, T. (2012). Domain Specific Data Retrieval on the Semantic Web. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*. doi: 10.1007/978-3-642-30284-8_35

Schreiber, G., Amin, A., Aroyo, L., van Assem, M., de Boer, V., Hardman, L., ... Wielinga, B. (2008). Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Web Semantics: Science, Services and Agents on the World Wide Web, 6*(4). doi:10.1016/j.websem.2008.08.001

Stefani, A. & Strapparava, C. (1999). Exploiting NLP techniques to build user model for Websites: the use of WordNet in SiteIF. In *Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto and Banff, Canada. Computer Science Report 99-07*, 95-100. Eindhoven: Eindhoven University of Technology.

Suchanek, F. M., Kasneci, G. & Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*, 697-706. New York: ACM. doi:10.1145/1242572.1242667

Vanderdonckt, J. & Florins, M. (2004). Graceful degradation of user interfaces as a design method for multiplatform systems. In *IUI '04 Proceedings of the 9th international conference on Intelligent user interfaces*, 140–147. doi:10.1145/964442.964469