

# A User-interaction Approach in an Information-cockpit

Christian Kaiser<sup>1</sup>, Selver Softic<sup>1</sup>, Vedran Sabol<sup>2</sup> and Mario Zechner<sup>2</sup>

<sup>1</sup>*Virtual Vehicle Research and Test Center, Graz University of Technology, Inffeldgasse 21A, Graz, Austria*

<sup>2</sup>*Know-Center GmbH, Graz University of Technology, Inffeldgasse 13, Graz, Austria*

**Keywords:** Visualization, Faceted Search, Metadata, Semantics, Future Work Place.

**Abstract:** This position paper introduces an approach for user interaction in an information-cockpit. The information-cockpit in our case is an explorative visual user interface backed by a faceted search-engine. The user interface employs several visualizations helping the user to get an overview of the data and providing intuitive drill-down functionality on search results. Besides searching and visualizing of data on a personal drive, our information-cockpit supports federated searching where files are located across heterogeneous data sources in the network.

## 1 INTRODUCTION

In the last years a massive growth of online and of-line data often lead to an information overload of the user. Structuring of the information stored in different data sources greatly supports searching, but even when guidelines do exist, interpretations of the imposed structures are likely to vary. Therefore, users who are not familiar with the system face a serious challenge when they need to search for and gather information. In a future workplace scenario a user should not have to deal with such problems - the system should take over the task of presenting the information in a unified, understandable way. This idea is, to a certain degree, comparable to an online search: instead of scanning or searching different Web-pages to find information, search engines are used as a single point of access. A real-life example would be a project-manager that is supposed to collect information from several sources for a milestone report. In this case the manager needs an overview about activities of his engineers instead of detailed knowledge about every line of code as well as a single access point to all the information. Our information cockpit represents a unified visual point of access providing search and visualization of information stored in heterogeneous data sources. In this work the focus is on the design of the visual user interface, since this has been identified as a key success factor for faceted search user interfaces (Hearst, 2006). Our concept is based on visualizations offered by available visualization libraries, such as D3.js Data-Driven Documents JavaScript library (Bostock, 2013).

## 2 RELATED WORK

According to Hearst (Hearst, 2009; Hearst, 2008; Hearst, 2006) the success of a user interface for search is very highly dependent on its design. One important aspect is that people prefer to be allowed to navigate and filter along multiple metadata fields, which provides them with a sense of control over the data. In case of visual support for search the work of Kienreich (W. Kienreich and Seifert, 2008) and Smith (G. Smith and D.S., 2006) as well as the book of Fill (Fill, 2009) have to be mentioned. Smith states that the point where a simple folder hierarchy for documents is sufficient to organize, keep track and search is already long past. However, many of the existing systems providing search and browse capabilities are still textual. Given ongoing explosion of gathered content enabled by automatic data collection mechanisms, a text-based interface may become increasingly overwhelming to users, especially novices (G. Smith and D.S., 2006). Kienreich introduces visual representations for filtering search results along topical, geographic and temporal dimensions. Tvarozek (Tvarozek and Bielikova, 2008) shows the usage of novel navigation and visualization approaches for search visualization such as visual depiction of facets and restrictions, visual navigation and graph like exploration of search results. Polowinski (Polowinski, 2009) states that faceted search is particularly suitable for structured but heterogeneous data with explicit semantics.

### 3 DATA PREPARATION

The data used for this example has its origin in the "Tankia" project of the TU Graz racing-team, an international engineering competition of student groups that compete in the formula student. In this section the preparation of the data is briefly described. The data is a mixture of office documents and files created by special engineering tools. Before the data can be indexed by the search engine it is necessary to first extract searchable information from the files and store it in a pre-defined format. A data preparation program was implemented for extracting useful information and metadata, and storing the results in XML<sup>1</sup> format. After that it is possible to build a Web-service with a REST (Representational State Transfer)-API (Application Programming Interface) which provides data for given questions. A JSON<sup>2</sup> (JavaScript Object Notation) formatted response of the API is preferred. The following code illustrates a possible JSON response.

```
{
  "result_information": {
    "#results": "500",
    "facets": {
      "person": "100",
      "car": "300",
      "security": "130"
    },
    "navigation": {
      "page_id": "1",
      "next_page_id": "2",
      "query": "http://gui.org/search&query=
Hans&facets=person+car+security"
    },
    "results": [
      {
        "uri": "file://path/xy.extension",
        "source": "N:",
        "metadata": {
          "modified": "2013:01:25:19:36:43",
          "authors": "Hans Huber",
          "created": "2012:12:21:15:36:43",
          "type": "text",
          "title": "car instructions"
        },
        "snippet": "[...]<b>car</b>[...]",
        "facets": [
          "person",
          "car"
        ],
        "relevance": "0.000134342"
      }
    ]
  }
}
```

<sup>1</sup>see <http://www.w3.org/XML/>

<sup>2</sup>see <http://www.w3.org/TR/json-ld-api/>

For the purpose of designing the REST-API which will be used by the visual user interface a set of 500 fictive search results is defined. The data structure returned as a fictive search result in JSON format can be seen above (for shortness just 1 result is shown). The extracted keywords are called facets, e.g. "person". According to Hearst (Hearst, 2006) facets are defined as following: Facets refer to categories which are used to characterize information items in a collection. Each facet has a name, such as person or car for a car-sharing collection. Facets can be flat or hierarchical. Some navigation information is provided to allow pagination. A query with page-id "2" added to the original query would lead to a response with the next page of the result-list. The results provide information about the unique URI, the source, some metadata like file-created date, file-modified date, the author, etc. Additionally a HTML (Hypertext Markup Language) styled snippet of the occurrence of a keyword is provided as well as the matched keywords and a calculated relevance.

### 4 GRAPHICAL USER INTERFACE (GUI)

At this point it is assumed that the data is prepared and searchable through the REST API. The next and most important step is the design (and later implementation) of the graphical user interface. There are two design guidelines that we wish to follow:

- ensure that usability of the GUI is high as this is a crucial point in regard to user acceptance.
- use different visualizations, each being adapted to the type of faceted metadata

GUI design shown in Figure 1 should, for the most part, be self-describing and its usage should be clear and intuitive. In the rest of this section a motivation and discussion of design decisions is provided.

#### 4.1 Search

As the main goal of our GUI is searching there is a text field on the very top of it. The user can type one or more keywords into the text field an hit enter or push the "search"-button to submit the search query to the system. A visually supported faceted search, which is discussed in the next subsection, can be used to filter the result set according to user's criteria.

#### 4.2 Visualized Faceted Search

As a search query can retrieve lots of results a possi-

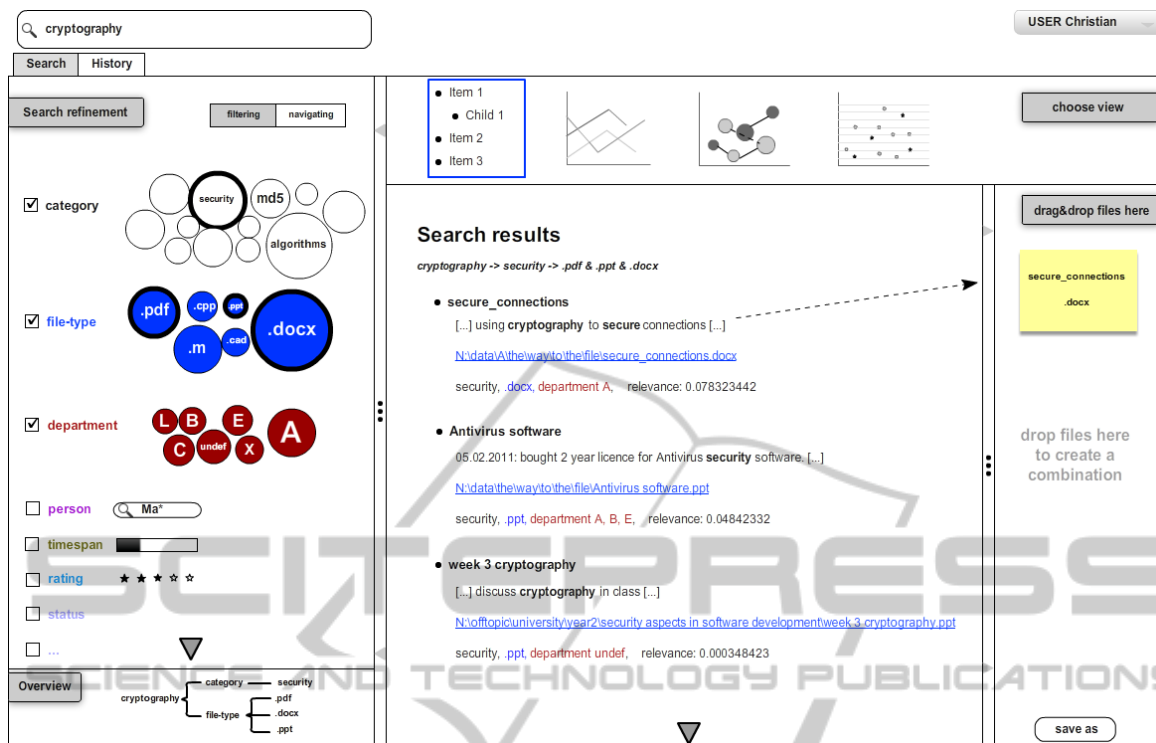


Figure 1: A mock-up of the GUI.

bility to narrow down and optimize the results is provided by using a faceted search (shown on left in Figure 1) - which is a well-known, proven exploration and discovery technique (Hearst, 2008). The intent is to use the faceted search additionally to an initial keyword-search, but it should also be possible to start with all existing results as base-amount for the filtering. When the user clicks on a facet a new search query is generated and sent to the Web-service, which filters the result-set with regard to the chosen facets. Filtering using any number of different facets is possible. As mentioned before, data type-dependent visual representations are used to provide a visually enhanced access to faceted metadata. The facet categories, e.g. "file-type" and "department", rely on the data indexed previously during data preparation. It is the job of the system administrator to identify such categories and make sure the chosen categories are extractable. The bubbles represent the identified possibilities of each category. The bigger a bubble is, the more results share this facet. Note that there is a minimum bubble-size to keep the describing text inserted into the bubble readable. The bubbles share the same color if they are members of the same category. Alternative representations are used for non-textual data, for example a star-representation for "rating" or a timeline (or a progress bar) for temporal information.

### 4.3 Result Visualization

Whenever the search query changes, either because of typing in a new keyword or selecting facets, a new query is sent to the API. The response will be displayed in the result box (center of the display). Per default a list is used to show the results, as this is the most commonly used and widely known format of presenting them. Above the result list there is a palette with available views for visualizing the results. Per default the list view is shown, with a blue-rectangle around the list button showing that this view is in use at the moment. Other result visualizations, such as the bar chart showing aggregations, a line chart displaying numerical values, or a graph visualization showing relations between search results will provide visual analysis capability to the user.

### 4.4 Collecting and Saving Results

A user might not be interested in only one single search result, but in a current status of a group of files, e.g. collect all necessary data for a milestone report. Therefore a so called "collection-box" is integrated on the right side of the GUI. The box content is persistent and does not change with new search-queries allowing the user to collect files over time. Adding hits from the search-result list (or visualization) to the

collection works via drag and drop. The user can click on the "save as" button and provide a destination directory where the collected files will be saved.

#### 4.5 History View

While a user has the possibility to copy the current status of files as discussed in the last section, he might need an overview of all previously used search-queries too, e.g. to search again for the very same by clicking on the saved query instead of retyping the keywords and reselecting facets again. Therefore the view of the GUI can be changed by clicking on the "History" tab below the text-field. In this view the used search queries are represented as trees. The root element is the keyword used in the query or "no keyword" if no keyword was used. The child elements are all the facets where an element was chosen, e.g. if only the file-extensions ".m" and ".pdf" have been chosen, then the child element of the keyword would be "file-extension" and again the child elements would be ".pdf" and ".m". The History tab contains a list of search-trees. If the user clicks on a search-tree, the query will be opened in the search-tab. The "Overview" box on the bottom left edge shows the current search-tree, which is saved in the history tab.

#### 4.6 Authentication

If files are used to save knowledge to a hard-drive, operating systems provide access-right handling. Pre-defined users and groups either have permission to read/write/execute or not. The approach described in this paper works with unstructured data, e.g. files of a network drive, and has to respect permissions. A simple way to do that is to force users to login with an account to authenticate. The question is, if a user should be able to see at least some metadata of the information which might solve his problem to get a hint whom to ask, even if he has not the permission to view the file. The underlying problem is that a user might be able to gather all critical informations out of the metadata, e.g. the snippet, and therefore can read a file he should not. Access is a complex problem and therefore not solved in the GUI approach yet, but a hint is given that there should be some management by placing the drop-down box on top-right which says current user is "Christian".

## 5 SUMMARY AND FUTURE WORK

In this paper a conceptional approach for user interaction in an information cockpit is presented. Hereby specific demands with regard to a future workplace scenario have been considered. As an outcome, knowledge about data-management to optimize data-extraction is retrievable in a retrospective. A drill-down oriented approach is used to access data combined along with state-of-the-art approaches like faceted- and explorative-search. It is expected to provide easy and personalized access to relevant data for users. Further interesting features for the described approach were identified and will be briefly described in this section. An important feature focuses on handling access rights and roles in the GUI. If there are no access restrictions, e.g. in a personal environment or in companies maintaining a free access policy for the employees, no problem exists: every user can see the same results. But if our information-cockpit is used in a company environment with different departments and lots of access restrictions the user shall see only what he is allowed to see and this should also include browsing through faceted metadata. Another part of the future work involves additional visualizations for browsing and analyzing the search results. Per default search results are shown as a list, but showing them within a hierarchy visualization, such as the "circle packing"<sup>3</sup>, might present an interesting option. Such a visualization would be also applicable for navigating hierarchically organized metadata facets (on the left side of the UI). For the search queries stored under the history tab an additional interesting scenario is proposed - a search agent: If a user searches for something but finds nothing or the results are not satisfying, the user might subscribe to the query and be informed when the result set changes. Finally, a heuristic evaluation of our design using a mock-up of the interface will be performed. This will allow us to unveil major usability issues and discover design inconsistencies before the implementation of the GUI begins.

## REFERENCES

- Bostock, M. (2013). Data-driven documents.  
 Fill, H. G. (2009). *Visualisation for Semantic Information Systems*. Gabler Verlag, Wiesbaden.  
 G. Smith, M. Czerwinski, B. M. D. R. G. R. and D.S., T. (2006). Facetmap: A scalable search and browse visu-

<sup>3</sup>See <http://mbostock.github.com/d3/talk/20111116/pack-hierarchy.html>

- alization. In *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12 (5), pages 797–804.
- Hearst, M. (2006). Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*.
- Hearst, M. (2008). Uis for faceted navigation: Recent advances and remaining open problems. In *Workshop on Human-Computer Interaction and Information Retrieval, HCIR*, pages 13–17.
- Hearst, M. (2009). *Search User Interfaces*. Cambridge University Press, New York, 1st edition.
- Polowinski, J. (2009). Widgets for faceted browsing. In Smith, M. and Salvendy, G., editors, *Human Interface and the Management of Information. Designing Information Environments*, volume 5617 of *Lecture Notes in Computer Science*, pages 601–610. Springer Berlin Heidelberg.
- Tvarozek, M. and Bielikova, M. (2008). Personalized view-based search and visualization as a means for deep/semantic web data access. In *Proceedings of the 17th international conference on World Wide Web (WWW08)*, pages 1023–1024.
- W. Kienreich, E. L. and Seifert, C. (2008). Apa labs: An experimental web-based platform for the retrieval and analysis of news articles. In *First International Conference on Applications of Digital Information and Web Technologies*, pages 58–62.