Integration of Visualization with Search Engines

Omar Alonso
Oracle Corp.
500 Oracle Parkway
Redwood Shores, CA 94065
USA
oalonso@us.oracle.com

Ricardo Baeza-Yates
Dept. of Computer Science
Universidad de Chile
Av. Blanco Encalada 2120, Santiago
Chile
rbaeza@dcc.uchile.cl

ABSTRACT

In this poster we analyze the dependency problem of the user interface with the information retrieval software. We propose an architecture that allows the separation of the user interface from the retrieval component. This is useful when the user wants to select an interface or visualization metaphor that could not always be available for different information retrieval systems or search engines. We present a model for visualizing large collections of documents in World Wide Web retrieval, independently of the retrieval system.

Keywords

Software architectures for WWW search, information visualization, integration with database systems, previews, overviews.

1. INTRODUCTION

The process of information resource discovery on the Web is a primary task that requires essential tools. Lately there has been a growth in the number of tools and applications in the information retrieval area that suppose to make the process easier for users. However, searching the Web is still difficult and there are still many challenges and open problems [2].

Search engines have a user interface for expressing the query and for displaying the answers. The interface is tightly coupled with the retrieval software and it varies from system to system. Given this scenario the user is not allowed to choose a user interface and a search engine independently. In other words, searching and visualizing information are not independent components.

It is important that the user has an option to select a user interface or visualization metaphor that he prefers over other ones, even the default retrieval user interface for his information seeking process. On the other hand, if a user has a preference on a retrieval system he should be able to use its services without hardware or user interface limitations. We believe this is important for producing previews and overviews of information in different systems (search engine, retrieval system, digital library, etc.) [3]

2. SOFTWARE ARCHITECTURE

The software architecture consists of a set of searchers, a set of visualizers, and an intermediate representation [1]. We defined a language called IVL (Information Visualization Language) that can be used as the intermediate representation and therefore allows searchers and visualizers to interchange data. We implemented IVL in XML.

The user has the choice of selecting a searcher and a visualizer as part of his profile. Then he issues a query to the system and reviews the search results using his selected visualization interface. Under the covers a series of transformations occurs to make this integration possible.

When the user issues a query (1), a query parser module translates the query to the selected search engine syntax (2). The results from the query can be in IVL format or not depending if the searcher supports IVL or not. If the searcher does not support IVL, a transformer will map the internal format of the searcher to IVL (3). Finally if the interface does not support IVL, a transformer will perform another transformation to generate the visualization's format (4). Figure 1 shows a simplified version of the architecture.

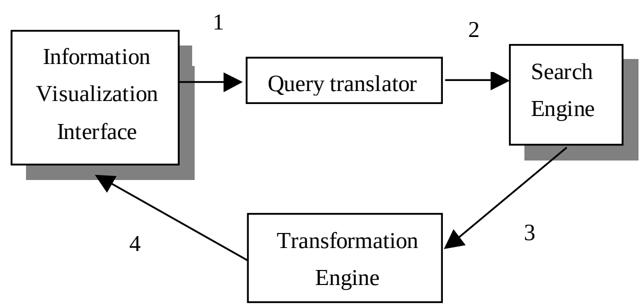

Figure 1. Software architecture

A key feature in our architecture is the transformation engine. It is possible to automate, in some cases, wrappers or adaptors generation. As an example, given a set of answers in IVL, it is very easy to generate a plug-in for Apple's Sherlock.

The following is an example in IVL of the results of a query about "Java and XML".

```
<?xml version="1.0"?>
<ivl><query>Java and XML</query>
<rank>1</rank>
<url>http://java.sun.com/xml</url>
<title>Java and XML</title>
<abstract>Sun's Java XML resource site …
</abstract>
<lm>Last modified 18-Oct-99 </lm>
<size> page size 4K </size>
<rank>2.</rank>
<url>http://www.oreilly.com/catalog/javaxml
</url>
<title>Java and XML </title>
```

```
<abstract>Shows you how to put Java and XML
together </abstract>
<lm>Last modified 27-Nov-00</lm>
<size> page size 4K </size>
<rank> 3</rank>
<url>http://technet.oracle.com/tech/xml/</url>
<title>XML page</title>
<abstract>Oracle XML resource page.</abstract>
<lm>Last modified 11-Jan-01 </lm>
<size> page size 12K </size>
</ivl>
```

3. ONGOING WORK

We implemented a prototype of the software architecture that includes Oracle *inter*Media Text as the search retrieval component [4] and several prototypes of user interfaces. We wrote the user interface prototypes in different languages: Java, JSP (Java Server Pages), and PSP (PL/SQL Server Pages) to explore different scenarios.

The prototype works as follows. The user issues a query from the browser. A Java servlet is responsible for opening a JDBC connection to the Oracle8*i* database, issuing the text query, and retrieving the rows. The search server returns the results in IVL and with the XML supported features from the product, a component transform the results into the visualization's format. In case the user interface has a proprietary format, another transformation is applied.

Figure 2 shows a screenshot of a two-view search interface. The left side shows structure and the right side shows services (e.g. highlighting, summary, etc.). The same user interface can be used with a different search engine. Figure 3 shows an AltaVista-like user interface using *inter*Media Text as the search engine. We are also exploring the automatic generation of simple user interfaces and previews/overviews with our architecture.

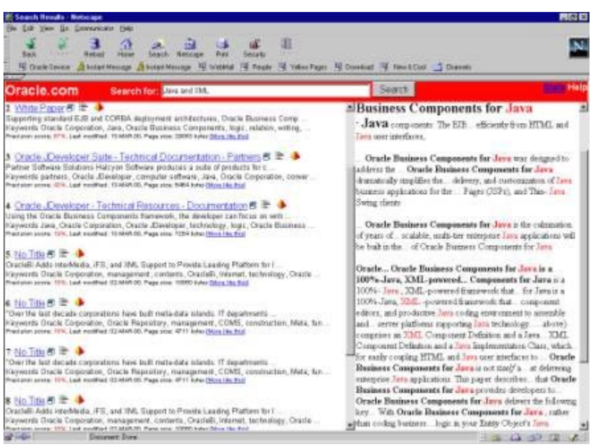

Figure 2. A Two-View search results interface

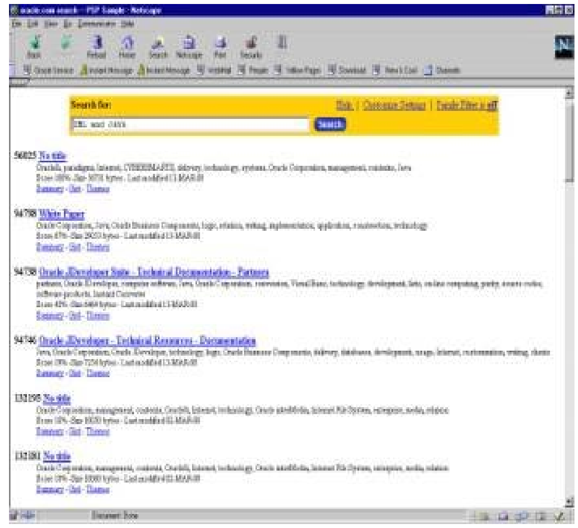

Figure 3. AltaVista-like user interface

4. CONCLUSIONS

With so many systems and tools for web retrieval, our model defines a way to share the output of those so that the visualization is more independent and therefore an option of several to the user.

Future work includes experiments with different hardware devices (e.g. cell phones, personal assistants, etc.) and integration with different visualization products that are available in the market.

5. REFERENCES

[1] O. Alonso and R. Baeza-Yates. "A Model and Software Architecture for Search Results Visualization on WWW". *VII Symposium on String Processing and Information Retrieval (SPIRE)*, A Coruna, Spain. IEEE Computer Society Press (September 2000).

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman, Edinburgh Gate, London (1999).

[3] S. Greene *et al*. "Previews and Overviews in Digital Libraries: Designing Surrogates to Support Visual Information Seeking". UMIACS-TR-97-73 ISR-TR (1999).

[4] Oracle *inter*Media Text Reference (8.1.6). Oracle Corp. Redwood Shores, CA (December 1999).