User-profiles for Information Retrieval

Bas van Gils basvg@cs.kun.nl
Eric D. Schabell erics@cs.kun.nl

University of Nijmegen, Computing Science Institute, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Abstract

In this position paper we will investigate a novel architecture for profile-based retrieval on the Web called `Vimes`[1]. This architecture is based on the fact that resources found on the Web should not only be topically relevant to a searcher's query; other characteristics (such as the file format or structural format) of a resource are equally important. Furthermore, `Vimes` uses profiles to deal with user characteristics and constraints.

1 Introduction

In today's "information society" information plays an increasingly important role. The trick is to get the right information at the right time and in an appropriate format for a given goal. Finding the right information has been researched extensively in the IR-field over the last decades. Already in the 1970's people tried to devise computer programs to assist them in their search for information. These computerized searches started with searching in homogeneous document collections such as in the STAIRS-project (Salton and McGill, 1983). The search process became more elaborate with the apparent rise of the Web. It led to the introduction of search engines such as GOOGLE which not only indexes (hyper)text, but also images, PDF-documents and interactive databases such as Citeseer (Citeseer, 1997). In other words, search engines attempt to retrieve relevant *resources*, rather than documents alone.

The importance of the timing aspect is particularly obvious when investment decisions are involved, such as on the stock market. Getting some information late could have huge (financial) consequences. Implementing a strategy for getting information in time often depends on many things such as choosing the right partner/supplier: some news sites are 'faster' then others in picking up news.

The third aspect mentioned deals with formats in the broad sense. It refers to "file format" (e.g. PDF, or HTML) as well as "structural format" (e.g. "abstract",

[1] Vimes is the Commander of the City Watch in Terry Pratchett's Discworld series™. We hope that our system will show the same ingenuity, tenacity and have the same success in information retrieval as Vimes has had at solving crimes.

or "photograph")[2]. The file format issue has been around since the early days of computing. Since people use different tools for jobs such as text processing a need for conversion tools between the file formats arose. Many of these conversions are available today. This is not (yet) the case for the latter issue, even though attempts have been made. A good example of this type of software is a computer program that generates abstracts for expository text (see e.g. (Barzilay and Elhadad, 1997)).

It is apparent that these factors vary for different users of IR-systems. For some users it is ok if certain financial records arrive slightly late, whereas for others it might have unpleasant consequences, some people would prefer an abstract of a (large) report over its full text etcetera. In other words, each of these factors can be seen as a *characteristic* of a searcher. Loosely defined, a *profile* is the collection of all characteristics of a searcher that are relevant for the retrieval process. The goal of this position paper is to investigate how profiles can be used to improve the IR-process and define the architecture of `Vimes`.

2 Overview

One of the basic functions of any information retrieval (IR) system is *relevance ranking*: the (characterizations of) resources are ranked such that the resources that are "most relevant" are listed first, and the ones that are least relevant are listed last. In (Dhyani et al., 2002) an overview is given of metrics that are used to determine the relevancy of a Web-document with regard to a query. Furthermore, it is pointed out that relevancy involves more than *topical relevance*; other attributes of resources (such as its quality and price) are important as well.

2.1 Relevance

In (Gils et al., 2003) a *conceptual model* for information supply is presented. This model is based on the notion that similar information can be conveyed by multiple representations, leading to the notion that several representations (resources on the Web) can belong to a single information service (provide access to their underlying representations). Based on this work, we define:

Definition 2.1 (representation format) *It is enforced that each representation has exactly one type. Examples of these types are:* PDF, HTML *and Webservice.*

Definition 2.2 (structural format) *Not all representations that belong to a single information service have to convey the same amount of information. For example, one conveys the "full content" and another is merely an "abstract". These (types of) structural format are modeled as* feature types *in (Gils et al., 2003). In this article we refer to them as the structural format.*

Using these definitions we can introduce our notion of relevance. Apart from topical relevance, which is the 'traditional' way of measuring relevance, we define that other constraints must be met as well. Examples of such constraints are

[2] In (Gils et al., 2003) the *structural format* is modeled as *feature types*.

its format (as explained in the previous section), but also price, quality etcetera. It may very well be that a searcher is willing to pay a certain amount of money in order to get his hands on a high-quality resource! Hence, we define relevance as follows:

Definition 2.3 (Relevance) *Resources are relevant with regard to a query if and only if this resource meets all the criteria that a searcher poses on it. These criteria can be formulated in either the query, or the user-profile.*

This definition resembles the notion of functional versus non-functional requirements in Software Engineering (Sommerville, 1989). It is now well accepted that non functional requirements and functional requirements are equally important to any software engineering project (see e.g. (Cysneiros and do Prado Leite, 2002; Barrett, 2002) for a discussion on the importance of non functional requirements).

This modified view of relevance has an impact on *precision* and *recall*, for it is 'less easy' for a document to be relevant with regard to a query. For example, it may be that a resource must be converted to another format before it is really relevant. In Section 3 we explain how a retrieval system can exploit this new notion of relevance in order to achieve 'better retrieval'.

2.2 Profiles

Already in (Myaeng and Korfhage, 1986) it was recognized that information retrieval systems can be personalized for users by means of profiles. During the last few decades a lot of research has been invested in the area of user profiles. Often, these profiles are used to enhance the query by capturing the user's notions of query terms (see e.g. (Myaeng and Korfhage, 1986; Chen and Kuo, 2000; Pierra et al., 2000)). However, profiles can be used more extensively. For example, in (Gligor, 1996) profiles are used for access control. We define that:

Definition 2.4 (Profile) *A (user) profile consists of a set of preferences with regard to behavior of a search engine as well constraints on the results it presents to the user.*

To illustrate this definition, the following list are the items that make up a particular user-profile:

preferences : I prefer a maximum of 25 results per page, and by selecting a relevant resource (clicking on the link) will open a new window.

constraints : I prefer HTML and PDF formats and refuse the Microsoft DOC-format. Furthermore, the size of the resource should not exceed 25Mb.

Using this definition, there are two areas in the retrieval process where profiles can be used. Firstly, they can be used for *post-processing* the results of the ranking process. For example, an resource that was found to be topically relevant can be converted to the proper format (See Section 2.1). Furthermore, profiles can be used to make sure that the retrieval engine operates according to the user's wishes.

2.3 Format

In the previous section we explained what profiles are and what they can be used for. In this section we present a *possible* format for storing these profiles, whereas in the next section we explain how/where they are stored exactly.

Since we want the profiles to be re-used across (Web) search engines, the format should be an *open standard*. More specifically, we want our format to be machine understandable and interoperable. The eXtensible Markup Language (XML, see e.g. (Bray et al., 2000)) is particularly well suited for this task (see e.g. (Suryanarayana and Hjelm, 2002)). The following XML-fragment is an example of what a profile could look like:

```
<? xml version="1.0" ?>
<!-- ------------------------------------------------------------ -->
<!-- A profile has an owner, identified by his/her Email-address. -->
<!-- Furthermore, a check-sum is included for security purposes.  -->
<!-- This profile stores 3 characteristics.                       -->
<!-- ------------------------------------------------------------ -->
<!-- define the owner of the profile -->
<profile owner="Bas van Gils" email="bas.vangils@cs.kun.nl" cs="2768A493">
    <!-- 1st characteristic: how many results per page? -->
    <characteristic type="results"> <page> 25 </page> </characteristic>

    <!-- 2nd charactersitic: the max. size in Mb -->
    <characteristic type="max_size"> <mb> 5 </mb> </characteristic>

    <!-- 3rd characteristic: preferred file-types -->
    <characteristic type="file_type">
        <type nr="1"> HTML </type>
        <type nr="2"> PDF </type>
        <type nr="3"> PS </type>
    </characteristic>
</profile>
```

Note that this excerpt is intended to illustrate our ideas. Defining a formal DTD for profiles is part of future research.

3 Architecture

In the previous sections we explained our notion of formats, profiles and relevance. These notions are essential for the architecture of *Vimes*, which we will introduce here. The architecture uses many elements that stem from previous research, such as brokers, agents, semantic web components and web services.

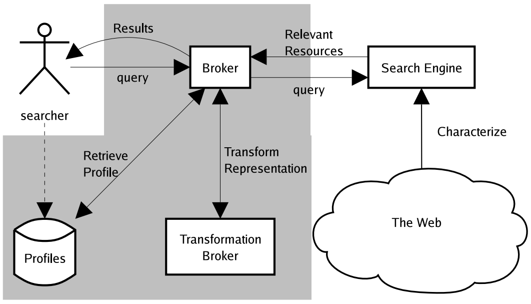
3.1 Components

User profiles will be stored in a repository for easy access and reuse by *Vimes*. The format in which this information will be maintained should be an accepted open standard, such as XML as outlined briefly in the previous section. Using such an open standard will make life easier for the user as they only need to define their preferences once. All retrieval engines that know how to deal with (this type of) profiles can re-use this single profile[3].

[3] Open issues that we need to work out still are the management of these profiles: where to store them? how to achieve an acceptable level of security?

In the previous section we introduced a new notion of relevance. We also explained how, in some cases, resources may have to be transformed before they are considered to be relevant. To cater for these transformations we introduce a *transformation broker*[4]. The broker will be a networked service, encapsulating functionality of all available transformation tools on the network and provide for multiple methods of transport. For example, a request to the transformation broker includes the form desired is PDF and the resource document is a postscript document. This particular conversion can be achieved by a transformation broker on the network that provides the tool *ps2pdf*. For similar reasons as before, we choose open standards for transport, such as FTP and HTTP. An additional benefit is that other parties can more easily participate/contribute by submitting transformation routines to the broker.

The broker component will be `Vimes`' main interface for users seeking information. It will interact with the user-profile repositories and search engines on the Web. Essential to our architecture is the broker's ability to interact with not only the well known web search engines (Yahoo, GOOGLE, AltaVista, Excite, etc.), but also with such enabling technologies as static agents, mobile agents, web services and services using the Semantic Web or Resource Description Framework (see e.g. (Google, 2003; Fünfrocken and Mattern, 1999; Berners-Lee et al., 2001; Miller et al., 2003)). Our broker component will also provide interaction with different forms of user-profile repositories, both local and remote. This will allow interaction with other profile systems on the Web (see e.g. (Pierra et al., 2000) for an agent-based approach along these lines). This leads to the following architectural diagram, with the components in the shaded area making up the `Vimes`-system.



Please note that the components will be loosely coupled so that they (especially the profile repository and the transformation broker) can also be accessed by other systems via the Web.

3.2 Example session

In this section we describe what the retrieval process could look like, based on the architecture as defined in the previous section. The first thing to be done is that the user creates a profile, preferably via an intuitive Web-interface, after which

[4] This transformation broker flowed out of the earlier work done on resource access for generic information retrieval (Schabell, 2002).

it can securely be stored in the repository. The second step is to browse to the broker, which functions as the main interface for the rest of the process. The user identifies himself (either automatically via e.g. a cookie, or more explicitly via a login-screen) after which the relevant profile is retrieved from the repository.

When the profile is retrieved, the user can enter his query into the system. Two things can happen at this point: either the broker decides to reformulate the query based on the user-profile, or it leaves the query untouched. Subsequently, the query is submitted to one of the search engines. This can be one of the well known web search engines, but others are possible such as an agent, a web service or other external services as described above. Based on the user's profile, the broker may decide to post-process discovered resources. The returned list of discovered resources would then be transformed using the transformation broker. If this is indeed the case, the resources are processed and ranked again before they are presented to the user.

3.3 Future work

The goal is to implement this architecture within the PRONIR research project (Proper, 2002) in the coming years. We will have to look deeper into the enabling technologies and will have to make a decision to narrow the possibilities down for our prototype. In either case, we will attempt to use open standards (XML, SOAP, WSDL, UDDI, etc) and open protocols (HTTP, FTP, Jabber, etc) throughout the entire architecture. Currently we are under way with regards to the transformation broker. We have setup a Conversion Clearinghouse that is web accessible, allowing users to search through our available conversions. In the near future we hope to have this online for extern input and usage. Development continues on this part of the project (Schabell, 2003).

4 Conclusion

In this article we started out by introducing a new way of measuring relevance. We feel that the relevancy of a resource with regard to a user query should be a combination of several factors such as topical relevance (the traditional measurement), but also things like its format, price, quality etcetera. To be able to deal with this more elaborate relevancy metric, we need to know a lot more about the user. Part of this additional knowledge is fixed over time for individual users. For example, a search may *always* prefer a certain file-type over others, and wants the search engine to list a maximum of 25 results per page. This lead to the introduction of profiles, which consist of constraints on the resources that are presented to the user as well as preferences with regards to the behavior of the search engine.

The main contribution of this article is the architecture of *Vimes*, introduced in Section 3. *Vimes* is intended to be a broker that assists users in querying the Web. There are three important components in this architecture. The *profile repository* stores the profiles of all users in an open format such as XML. There are still some open issues in this area, such as specifying a language for storing the profiles, enforcing that they are stored securely, etcetera.

The second component is the *transformation broker*, which enables us to perform transformations on resources found on the Web. With these transformations we hope to be able to transform resources into a format that is convenient / wanted by individual users. For example, we can transform a HTML document into PDF, or generate an abstract of a report that is too long according to a user's profile. We are currently working on a system that performs these transformations by setting up a Conversion Clearinghouse that is web accessible, allowing users to search through our available conversions.

Last but not least, the *broker* in the `Vimes` architecture is the user-interface. It interacts with the two other components, as well as with search engines on the Web. Much work remains to be done in this are also. For example, we need to figure out what the interface will look like, which message-standards are going to be used to interface with the other components, etcetera.

All in all, this article provides insight into a novel way of thinking about retrieval. It outlines the architecture `Vimes`, without giving a full specification. Finally, we have presented a road-map for our research.

References

Barrett, M. L. (2002). Putting non-functional requirements to good use. *The Journal of Computing in Small Colleges*, 18(2):271–277.

Barzilay, R. and Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97)*, Madrid, Spain.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., and Maler, E. (2000). Extensible markup language (XML) 1.0 (second edition). Technical report, World Wide Web Consortium,
http://www.w3.org/TR/REC-xml. last checked: 19-may-2003.

Chen, P.-M. and Kuo, F.-C. (2000). An information retrieval system based on a user-profile. *The Journal of Systems and Software*, 54(1):3–8.

Citeseer (1997). *NEC ResearchIndex Citeseer*.
http://citeseer.nj.nec.com. Last checked: 19-may-2003.

Cysneiros, L. M. and do Prado Leite, J. C. S. (2002). Non-functional requirements: from elicitation to modelling languages. In *Proceedings of the 24th international conference on Software engineering*, pages 699–700, Orlando, Florida. ACM Press. ISBN: 1-58113-472-X.

Dhyani, D., Ng, W. K., and Bhowmick, S. S. (2002). A survey of web metrics. *ACM Computing Surveys (CSUR)*, 34(4):469–503. ISSN:0460-0300.

Fünfrocken, S. and Mattern, F. (1999). Mobile agents as an architectural concept for internet-based distributed applications - the wasp project approach. In Steinmetz, editor, *Proceedings of the KiVS'99 ("Kommunikation in Verteilten Systemen")*, pages 32–43. Springer-Verlag.

Gils, B. v., Proper, E., and Bommel, P. v. (2003). Towards a general theory for information supply. In *Proceedings of the 10th International Conference on Human-Computer Interaction*.

Gligor, V. (1996). Characteristics of role-based access control. In *Proceedings of the first ACM Workshop on Role-based access control*, Gaithersburg, Maryland, United States. ACM Press. ISBN: 0-89791-759-6.

Google (2003). *Google Web API's*. Google,
`http://www.google.com/apis`. last checked: 16-May-2003.

Miller, E., Swick, R., and Brickley, D. (2003). *Resource Description Framework (RDF)*. World Wide Web Consortium,
`http://www.w3.org/rdf`. last checked: 16-May-2003.

Myaeng, S. H. and Korfhage, R. R. (1986). Towards an intelligent and personalized retrieval system. In *Proceedings of the ACM SIGART international symposium on Methodologies for intelligent systems*, pages 121–129, Knoxville, Tennessee, United States. ACM Press. ISBN:0-89791-206-3.

Pierra, S., Kacan, C., and Probst, W. (2000). An agent-based approach for integrating user profiles into a knowledge management process. *Knowledge-Based Systems*, 13(5):307 – 314.

Proper, E. (2002). PRONIR proposal. Technical report, IRIS / KUN. NWO Project Proposal.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill New York, NY.

Schabell, E. D. (2002). Resource access in generic information retrieval systems. Master's thesis, Vrije Universiteit, Amsterdam, Netherlands.

Schabell, E. D. (2003). *Profile Based Retrieval Of Networked Information Resources, The Scientific Programmers Workshop*.
`http://www.pronir.nl/pub/spws`. Last checked: 16-May-2003.

Sommerville, I. (1989). *Software Engineering*. Addison-Wesley, Reading, Massachusetts.

Suryanarayana, L. and Hjelm, J. (2002). Profiles for the situated web. In *Proceedings of the eleventh international conference on the World Wide Web*, pages 200–209, New York, NY, USA. ACM Press. ISBN:1-58113-449-5.