INTRODUCTION

Traditional database and information retrieval systems have followed a \textit{query-based} information access paradigm (i.e., information is returned to the user on the basis of a query issued). As a result, users issuing the same query are provided with the same answer. With the advent of the World Wide Web and hand-held electronic devices such as palmtops and cellular phones, information access entered a new era. Increasing amounts of information become available to a growing mass of untrained lay users through various access media. A user searching Web-resident information may have to reformulate queries issued several times and sift through many results until a satisfactory, if any, answer is obtained. As purely query-driven approaches may be inappropriate in this context, the need for a shift towards a more user-centered information access paradigm arises. To this end, different approaches aim to the personalization of the overall user experience at different levels: content selection, content presentation, and user interaction. There is no generally accepted definition of personalization, so I adopt a broad one as follows:

- **Personalization** is the approach of providing an overall customized, individualized user experience by taking into account the needs, preferences, and particular characteristics of a user or group of users.

Focusing on the level of personalized content selection, several distinct lines of research exist. There are two broad categories: filter-based and personalized approaches.

- **Filter-based** approaches filter system responses on the basis of a user profile, storing long-term user interests. In particular, information filtering methods employ profiles comprised of keywords (Foltz & Dumais, 1992). Anatagonomy is a representative example of customized Web-based newspapers employing information filtering methods (Sakagami, Kamba, & Sugiura, 1997). Web-accessible databases employ \textit{continuous queries} to allow users to obtain new results from the underlying collection or stream without issuing the same query repeatedly (Chen, DeWitt, Tian, & Wang, 2000; Liu, Pu, & Tang, 1999). User preferences are also used for providing \textit{recommendations} (Karypis, 2001; Shahabi, Banaei-Kashani, Chen, & McLeod, 2001), and for ranking search results (Glover, Lawrence, Birmingham & Lee Giles, 1999; Smyth, Bradley, & Rafter, 2002).

It is worth noting that query-based and filter-based approaches have been characterized as two sides of the same coin (Belkin & Croft, 1992). Filter-based approaches deal only partially with the problem of information overload. This problem still haunts Web searches in which users may issue various queries expressing different information needs.

- **Personalized** approaches are based on the observation that different people find different things relevant; therefore, they may expect different answers to the same query. Consider a simple example: John and Ann access a Web-based movies database, searching for comedies. John is a fan of director W. Allen, and Ann is not. Traditional query-based systems would consider only the query issued and return the same, exhaustive list of comedies to both users. Focusing on the user enables a shift from what is called \textit{consensus relevancy} in which the computed relevancy for the entire population is presumed relevant for each user, toward \textit{personal relevancy} in which relevancy is computed based on each individual’s characteristics (Pitkow et al., 2002). Personal agents and query personalization approaches belong here. \textit{Personal agents} represent virtual assistants that learn user preferences by dialoguing in natural language with customers. These preferences are used for the formulation of personalized queries and generation of individual recommendations (André & Rist, 2002; Semeraro, Degemmis, Lops, Thieli, & L’Abbate, 2003). \textit{Query personalization} approaches exploit user preferences stored in profiles to dynamically enhance any user query by integrating preferences relevant to it (Koutrika & Ioannidis, 2004b; Pitkow et al, 2002). Personalized results may be ranked according to user preferences.
Figure 1. Information access paradigms

<table>
<thead>
<tr>
<th>Query-based paradigm</th>
<th>Filter-based paradigm</th>
<th>Personalized paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Content Selection</td>
<td>Preference Selection</td>
</tr>
<tr>
<td>e.g., Traditional IR &amp; Databases</td>
<td>Profile</td>
<td>User Profile</td>
</tr>
<tr>
<td>Query</td>
<td>Profile</td>
<td>Preference Integration</td>
</tr>
<tr>
<td>e.g., Information Filtering</td>
<td>Content Selection</td>
<td>User agent</td>
</tr>
<tr>
<td>Query</td>
<td>Profile</td>
<td>Preference Execution</td>
</tr>
<tr>
<td>e.g., Query Personalization</td>
<td>Content Selection</td>
<td>Personalized Query Execution</td>
</tr>
</tbody>
</table>

Figure 1 summarizes the aforementioned information access paradigms.

BACKGROUND

Storing user preferences in user profiles gives a system the opportunity to return more focused personalized (and hopefully smaller) answers. The primary ways to personalize a search for an active searcher are query augmentation and result ranking. Returning to the previous example, John’s personalized results would include W. Allen’s comedies, and Ann’s would not. Which preferences are relevant to a specific request and how they affect the final answer are dynamically determined based on the query, the profile, and the personalization philosophy adopted. For example, when Ann is searching for a theatre to go to, the system should also consider her preference for downtown theatres. Query personalization approaches have recently attracted interest in both information retrieval and databases research communities.

*Outride* is a personalized IR system (Pitkow et al., 2002) that exploits user profiles defined upon the ontology of the open directory project (ODP). For query augmentation, the similarity between the query term and the user model is computed to decide which, if any, of the stored keywords are relevant to and should be included in the query. For example, if an individual is interested in coffee information and then searches for java, the system may dynamically augment the query by adding the relevant term *coffee* to provide the user with results about Java coffee. If another user interested in programming searches for java, the system may augment the query by adding the term *programming* or *language* to provide this user with results about the Java programming language. To rank search results based upon the user profile, metadata from the pages are compared via vector methods against the profile.

Another personalized IR system is described by Liu, Yu, and Meng (2002). A query is mapped to a set of categories stored in a user profile. A category is comprised of a set of terms with weights. A term weight reflects the significance of the term in representing the user’s interest in that category.

Koutrika and Ioannidis (2004b) have implemented a first personalized database system and provide a query personalization framework that specifies which preferences from a profile should affect a query issued. Based on this framework, the top-K preferences are integrated into the query producing one that returns results satisfying at least L of them. Preferences are stored as degrees of interest in atomic query elements. Consider a user that accesses a movies database looking for films released in 2004. The corresponding SQL query could look like this:

```
Select title from MOVIES where year = 2004
```

Assuming the user is interested in comedies, the system may execute a personalized query that could look like this:

```
Select title from MOVIES, GENRES
where MOVIES.id = GENRES.id
and year = 2004 and genre = "comedy"
```

The aforementioned efforts have shown that the benefits of personalized search can be significant, appreciably decreasing the time it takes people to find information. In the following section, I focus on personalization of database queries.

DATABASE QUERY PERSONALIZATION

A personalized database system keeps a repository of user profiles (Koutrika, 2003). Information in profiles is either inserted explicitly by the user or collected implicitly by monitoring user interaction with the system (profile creation). When a new query is issued, query personalization proceeds in three stages: (a) preference selection (preferences relevant to the query are extracted from the user profile); (b) preference integration