A Personalization Layer for Mediation Systems

Imane Zaoui, Faouzia Wadjinny, Dalila Chiadmi, Laila Benhlima
Mohamed V-Agdal University
Computer Sciences Department
Mohammadia Engineering School
BP765, Rabat Agdal, Morocco
{zaoui, wadjinny, chiadmi, benhlima}@emi.ac.ma

ABSTRACT: Mediation systems allow integration of disparate and heterogeneous data sources. Unfortunately, with the increasing number of integrated sources, especially in the context of web, mediator’s responses time increases considerably while the answers are not adapted to users profiles. In this paper, we present a personalization layer that makes the mediation system more sensitive to user’s preferences. The layer sits between the user and the mediation system. It captures the user profile and performs a smart source selection according to user’s interests and quality preferences. Selection is based on similarity measurement between users and sources models. Our solution for personalizing data source selection provides the mediator with a set of relevant data sources. These are then involved in the rewriting process to give more satisfying response.

Categories and Subject Descriptors:
C.2: Computer-Communication Networks Data communications
E.2: Data Storage Representations: K 8 [Personal Computing];
H.3.4 [Systems and Software]: User profiles and alert services

General Terms: Personal Data Management, Mediation Systems, User Profile

Keywords: Personalization, User profile, Mediation systems, Source selection

Received: 18 August 2011, Revised 2 October 2011, Accepted 7 October 2011

1. Introduction

The aim of a mediation system is to provide a uniform access to multiple distributed and heterogeneous data sources [1]. A user accessing a mediation system with the intention to satisfy an information need, may have to reformulate many times his query and to analyze the returned answers, trying to find the relevant information. This is a typical behavior caused by the lack of personalization features into mediation systems. Indeed, consider a simple case where two users, A and B, are seeking for a destination to spend their holiday. User A prefers a cold region whereas user B prefers a hot one. Suppose their query is identical: “retrieve touristic regions where to spend holiday”. User A and B expect different answers from the system but unfortunately they got the same exhaustive list of regions in the same order. The mediator returns the same answer for the same query although user’s goals and preferences are different and are sometimes contradictory. Furthermore, with a huge number of data sources involved, especially in the context of the Web, a user query addressed to the mediator takes a lot of time to be executed and produces an information overload as well. That means that even if existing data integration projects succeed in achieving a common technology platform, they are usually rejected by the user communities because of the irrelevant information or the presence of poor data quality [2]. Based on the above, one of the key issues to insure user satisfaction and make mediation systems competitive is personalization. This paradigm is about building a meaningful one-to-one relationship by understanding the needs of each individual [3], and delivering content and services that are tailored to individuals based on knowledge about their preferences and behaviors [4]. In this paper, we present our solution for increasing user satisfaction in mediation systems. It’s a software layer that sits between the user and the mediator. It offers many personalization functionalities including user profiling, data source modeling and data source selection. User modeling gives to the layer a structured knowledge about the user interests and preferences. Data source modeling provides a common description of candidate data sources. Finally, data source selection returns the most relevant data sources that respect both user interests and quality preferences.

The reminder of this paper is structured as follows. In section 2, we present the personalization layer functionalities and architecture. Then, in section 3 we describe how we model users through a user profile. Section 4 gives our source profile. Section 5 explains our personalized source selection procedure. In section 6, we present some related works. We conclude in section 8.
2. The Personalization Layer

Mediation systems return an integrated response to each user query. This answer is not customized, so user may spend significant time trying to distinguish relevant information. Furthermore, the mediator doesn’t differentiate between users, so the responses are not adapted to individual’s needs and preferences. To tackle these limitations, we add a layer that we call personalization layer on the top of the mediation system. This layer offers many personalization functions. It interacts with the user, the mediator and the sources following a cyclic process.

2.1 The Personalization Layer Functionalities

The personalization layer belongs to the market centered category of personalization techniques where the personalization engine is considered as an infomediary that knows the consumer needs and the provider’s offerings and try to match them in the best possible ways [5]. Our personalization layer offers personalization functions which ranges from simply customizing the user interfaces (in terms of layout, colors, frequently used services, etc), to very complex services that automatically assist the user during his interaction with the system. The main functionalities are the following:

- Create and manage user profiles while ensuring the confidentiality and security of personal data
- Recognize user and loading his profile through an authentication mechanism
- Update user profile automatically by analyzing user behavior and relevancy feedback
- Provide an adaptable QBE (Query By Example) interface that allows user to enter his queries and express his judgments. User can also customize this interface.
- Create and update sources profiles which define available sources integrated by the mediator.
- Select a set of relevant sources according to user’s content goals and quality preferences from available sources. This selection is performed using matching algorithms.
- Adapt user queries based on user relevancy feedback
- Adapt mediator results according to user preferences in particular readjusting them by their relevancy.
- Offer recommendation services.

These functionalities are based on executing an iterative process composed of four steps which are: capturing knowledge about users, tailoring responses to their profiles, measuring their satisfaction and adapting responses to their feedbacks and behaviors. This is a continuous improvement cycle for improving the quality of mediation systems. To represent the key activities of the personalization process, we adopt the Deming wheel model so-called PDCA (Plan Do Check Act)[6]. Plan activities revolve around modeling user and sources profiles, customizing user interface and setting up the personalization layer functions. Do stage consists of executing personalization services in particular selecting relevant sources, adapting mediation answers to user preferences and recommend information or products according to user profile. Check stage belongs to measuring user satisfaction among the personalized responses. These measurements are either in the form of scores that user attributes to the retrieved answers or coefficients values computed by the layer like the precision and recall measures [7]. Finally, Act stage concerns readjusting the user profile and generating an expanded query that may better meet the user profile and increase his satisfaction.

2.2 The Personalization Layer Architecture

The personalization layer is directly connected to the mediation system in a way that the outputs of the personalization layer are the inputs of the mediator and vice versa. The layer interacts with users, sources and the mediator following the process described above. Thus we build our personalization layer on four interaction levels which are:

**User level:** this level is composed of two modules: 1) a QBE interface which is the front office manipulated by the user. It allows him to create accounts, open sessions via authentication, define and manage his profile, enter queries, get the personalized responses and express his feedback among the obtained results; 2) a user profiler which is the back-office agent mainly responsible for setting up the user profile, analyzing user behaviors, capturing his feedbacks and learning changes that may occur throughout the session.

**Source level:** This level is responsible of defining and managing sources profiles which describes the available sources via multiple dimensions.

**Application level:** this is the core level where personalization services are executed, in particular, selecting relevant source, reordering results and modifying queries.

In addition, these three levels manipulate data like users’ profiles and queries, sources’ profiles, and mediator’s answers. We store them into a knowledge base.

3. User Profiling

The personalization layer requires accurate modeling of user profile to insure user adaptation. This is the main role of the User level. This section highlights the user profile organization and focus on user content and quality preferences.

3.1 User Profile Organization

We organize the user profile in two entities: the persistent profile and the session profile [8].

**Persistent profile** contains general characteristics of a user which don’t change for a long period. It includes five dimensions: the personal identity (name, age, address, job, etc); the domains of interest which group all attributes
and preferences related to the general information needs of a given user; the general expected quality; the security data and the interaction history. This persistent profile is used in constructing user communities which is a key issue for mediation systems providing recommendation and collaborative filtering services [9].

Session profile is a short term profile describing the user during one session. It represents a particular instance of the persistent profile. The session profile contains four dimensions: the user context which informs about the location, time, used devices, etc; the user goals which relates to the specific user domains of interest during a particular session; the user quality preferences that covers required quality parameters and values in the current session. The session profile helps to catch the evolution of user characteristics and needs through multiple sessions. This is necessary in updating the knowledge about the user and adapting dynamically the mediator responses.

To build our user profile, we adopt an explicit approach where users describe their goals, preferences, security parameters and quality requirements using an interactive interface. Then, we update the profile through learning techniques [10]. We will not discuss these update mechanisms since it is not the scope of this paper. In addition, in the reminder of this paper, we consider only the user goals and quality preferences dimensions from the session profile.

3.2 User Goals Representation

During an interaction session, user queries revolve around a single objective which can be declined into multiple goals. Each goal represents a domain of interest related to the interaction session. Furthermore, users have preferences among those goals. For example, a session objective may be retrieving documents dealing with personalization in data integration systems. User goals during that session could be mediation systems, query rewriting, and personalization. User preferences among these goals are represented via weights, which are numerical values entered directly by the user. To formalize the user goals dimension, several approaches have been proposed in the literature, in particular ontologies and vectors. Ontologies depend on the application domain [11].

Furthermore, in the context of mediation, integrated sources may have different ontologies. To take advantage from this semantic representation, user goals ontology and all sources ontologies requires alignment. This is a hard task especially in the context of the web. Since we propose a user model that could be adopted in a large variety of domains, it is clear that the model of the ontology does not meet our expectations. We choose the vector model [12]. So the user goals dimension is a weighted vector of key words, where key words are user goals (domains of interests during the session), and weights reflects user preferences. For convenience to theoretical studies, we consider for the weights, a range of [0,1]. In reality, user preferences are normalized according to a given scale so that the user preferences are values into [0,1]. For example, the user could set his preferences via a slider into the user interface. The slide position is then transformed to the corresponding value.

The user goals dimension is given by definition 1.

Definition 1: The user goals of a user \( U_i \) is a weighted vector of key words given by: \( UG(U_i) = (t_{ui1}, w_{ui1}; t_{ui2}, w_{ui2}; ..., t_{uin}, w_{uin}) \); where \( t_{uij} \) are key words and \( w_{uij} \) their respective weights.

3.3 User Quality Preferences

In mediation systems, user satisfaction depends on the quality of returned responses. Since mediation systems combine responses from different integrated data sources, the quality of the final response relies on the quality of the involved sources. Consequently, we believe that indicating the user quality preferences among data sources in the user profile makes the personalization process more accurate. The user quality preferences dimension is the aggregation of multiple quality criteria, such as accuracy, popularity, completeness, freshness, etc. [13]. To build this dimension, we propose a model where the user chooses his desired quality criteria from a global list in each session. Then, he associates a weight and a value for each criterion. Weighting quality criteria helps the system to emphasize the priority of the quality criterion to satisfy. To simplify our model, we use the following scale for weights: (0.4: mandatory criteria, 0.3: desirable, 0.2: not desirable, 0.1: indifferent). Concerning quality values, user express them with a numerical score in the
appropriate unit of measure.

The user quality preferences dimension is formalized by definition 2.

**Definition 2:** The user quality preferences of a user $Ui$ is a vector given by: $UQP(Ui)$ = $(qu1, wu1; qu2, wu2; ... ; quin, wuin)$, where $quj$ are quality criteria, $wu$ their respective weights and $vuij$ their desired values.

### 4. Source Profiling

Data sources in mediation systems are heterogeneous, distributed and autonomous. To describe them, local schemas and ontologies are generally used. But these descriptions are not sufficient to give details about the content and the quality criteria of the integrated data sources. For this reason, we refine the sources description by building a source profile. The source profile is managed by the source level into the personalization layer. It contains a variety of information including source location, identity, owner, content, quality criteria and so on, which helps in the source selection process. In the following, we present our source profile and we focus on the dimensions describing the content and the quality of the source.

#### 4.1 Source Profile Organization

The source profile we propose is generic and multidimensional [14]. It can be used in a variety of application domains. We adopt the same multidimensional representation used for the user profile. Our objective by doing so is to perform the matching between the source profile and the user profile. A brief description of our source profile dimensions is given bellow.

**Source identity** describes the source identity with the following attributes: Id, name, URL, port, owner, size, principal languages and principle types of content.

**Source content** represents the most important topics treated by the sources. This information is available in the form of weighted key words or concepts.

**Source ontology** represents the concepts used in the data source and describes the semantic relationship between them. This dimension is used by the mediator to insure the semantic interoperability during the rewriting process [15].

**Source quality** describes the main quality characteristics of the source in terms of quality criteria such as freshness, popularity, response time, etc.

In the following we give our formalism for source content and source quality dimensions.

#### 4.2 Source Content Dimension

To represent the source content dimension, we use the same formalism than the user goals of the user profile. This is necessary to perform content matching between the source profile and the user profile. So, the source content dimension of the source profile is represented by a weighted vector of key words or concepts. The concepts are extracted from the source ontology if it exists or through an indexing method. The weights of the terms (key words or concepts) are calculated using the well known $TF*IDF$ schema explained in the vector model of Salton [12]. Using this formalism, we give in definition 3 the source content dimension of the source profile.

**Definition 3:** The source content $SC(Si)$ of the source $Si$ is a weighted vector of concepts given by: $SC(Si)$ = $(csi1, wsi1; csi2, wsi2; ... ; csin, wsin)$, where $csi$ represents the concepts and $wsi$ their respective weights.

#### 4.3 Source Quality Dimension

We define the source quality dimension as the main quality criteria that make a significant difference between data sources. Several quality factors exist in the literature [13][16][17]. For example, understandability, credibility, precision, correctness, etc. All these factors could be added in our model easily. The user chooses those meeting his quality requirements. In this paper, we focus on four information quality metrics which are reputation, timeliness, completeness and time of responses. Reputation means the degree to which a source is in high standing [18]. We consider that the reputation of a source “$S$” is expressed by a set of users following a scale from 1 (bad reputation) to 5 (very high reputation). Timeliness [19][17], measures the time elapsed since data was updated. For example, “Timeliness=2 years” means that the source contains documents updated 2 years ago. Completeness is the extent to which data is not missing and are of sufficient breadth, depth, and scope for the task at hand [18]. For instance, we measure the completeness of a bibliographic source “$S$” as the percentage of relevant documents according to a given query out of the total size of this source. Response time is the time that a source takes to answer a given query. Response time could be very high if the source is saturated or doesn’t have the capability to answer the query [20]. In this paper, we suppose that all sources could answer all queries so that the problem of source capabilities is resolved. Consequently, the time of response depends only on the communication process with the source. We use sample queries to determine this factor.

### 5. Personalized Source Selection

Let’s remind that in most mediation systems, source selection is done during the rewriting process. Once the user submits a query, formulated in terms of mediated schema, the mediator decomposes it into a set of sub-queries targeted to the appropriate sources. This method gives the same results for the same queries and does not respect user preferences. To tackle this limitation, the personalization layer performs at the application level a source selection procedure, based on matching the user profile and the sources profiles in terms of content and quality. Only selected sources will be involved while integrating results. In this section we present the content matching and the quality matching algorithms.
5.1 Content Profile Matching.
In this step, we are interested in the user goals \( UG (Ui) \) and the source content \( SC(Sj) \) dimensions. To perform a content matching, we face two major problems. First, \( UG \) and \( SC \) vectors have different cardinalities and contain different terms. So, they could not be compared since they are not homogenous. To overcome the problem of vector homogeneity, we propose to take into account the common terms with their relative weights and add absent concepts in each \( SC \) vector with a weight of zero. Second, we have to rank the sources from the most to the less relevant ones according to the user goals and preferences. The ranking is given by a similarity score which measures the degree of content matching between user profile and source profile. Similarities between two homogenous vectors could be calculated in different manners: using distances like the Euclidian distance, Manhattan distance, Hamming distance etc. [21], or using similarity measures based on the inner product of the vectors. The most popular ones are the Cosine angle, Jaccard coefficient, and Dice Coefficient [22]. Distances and similarity measures are equivalent. Indeed, the more the vectors are similar, the less is the distance between them. For next, we focus on similarity measures instead of distances. The Cosine angle is the most simple and widely used similarity measure. The main advantage is its sensitivity to the relative importance of terms in each vector. Jaccard and Dice measures were initially defined for binary vectors, thus, measuring the similarity as a proportion of common terms (having a weight of 1) and non common ones (having a weight of 0). These measures have been extended to support weighted vectors. Similarities between \( SC \) and \( UG \) depend on the number of common terms and also on their weights. Furthermore, we need to emphasize the importance to the common terms instead of terms that are not shared. For these reasons, we propose in definition 3 to calculate the similarity score as the average of the Cosine, Jaccard and Dice coefficients.

Definition 4: Given \( UG(Ui) = \langle tu1, wu1; tu2, wu2; ... tuin, win \rangle \) and \( SC(Sj) = \langle cs1, ws1; cs2, ws2; ... csjn, wsnj \rangle \), the homogenized vector representing the source content dimension of source \( Sj \). We note \( sim(Ui,Sj) \) the similarity score between \( UG(Ui) \) and \( SC(Sj) \). It is given by formula 4:

\[
sim(Ui, Sj) = \frac{a}{3}\left( \frac{1}{\sqrt{bc}} + \frac{1}{b+c-a} + \frac{2}{b+c} \right)
\]

where

\[
a = \sum_{k=1}^{n} wu_{ik} \cdot ws_{jk} \quad b = \sum_{k=1}^{n} wu_{ik}^2 \quad and \quad c = \sum_{k=1}^{n} ws_{jk}^2
\]

The similarity score measures how closest are the User Goals and the Source Content vectors. By definition, \( sim(Ui,Sj) \) is based on the Cosine, Dice and Jaccard coefficients. Egghe prove in [22], that all these coefficients respect the properties of OS-Similarity measures. That means that they allow ordering sets of comparable items, which is the purpose of assumption 1. It is easy to verify that \( Sim \) score, based on the average of the cosine, Dice and Jaccard coefficients, is an OS-similarity. Consequently, the higher the score is, the higher the vectors are close. Also we can consider that a source is relevant if the \( SC \) and the \( UG \) vectors are close. This proves assumption 1 which allows ranking the sources from the most relevant to the less relevant according to their similarity scores.

Assumption 1: Given a user \( Ui \) and two sources \( Sj \) and \( Sk \), \( Sj \) is more relevant than \( Sk \) if \( sim(Ui,Sj) > sim(Ui,Sk) \). We note: \( Sj > Sk \).

But this relevancy ranking is not sufficient to select only the most relevant sources. According to Pareto principle [23], we can say that 80% of user satisfaction comes only from 20% of available sources. That means we can considerably reduce the number of integrated sources while getting a satisfying result. To attempt this goal, user sets a parameter so-called relevancy threshold \( (Rt) \). For example, \( Rt= 60\% \) means that the sources having a similarity score less than 0.60 are considered irrelevant. This relevancy threshold depends on user preferences. To select only the most relevant sources, so-called \( m- \)relevant, we use definition 4.

Definition 5: Given a user threshold \( Rt \) and a similarity score \( sim(Ui,Sj) \). \( Sj \) is \( m- \)relevant if \( sim(Ui,Sj) \cdot 100 \geq Rt \).

5.2 Quality Profile Matching.
In the previous section, we explain how we select sources according to user goals. The selected sources may have different quality characteristics, and respect more or less the user quality preferences. To perform an accurate source selection, it is important to refine the content selection by a selection based on quality criteria. This is ensured through the quality profile matching. Since the source quality is measured with several criteria, the quality profile matching could be studied as a multi-attribute decision making problem (MDMP). In the literature, several methods have been developed to resolve this problem such as SAW, TOPSIS and AHP [24]. We choose to apply SAW (Simple Additive Weighting) [25], because it is one of the most simple but nevertheless a good decision making procedure. SAW results are also usually close to more sophisticated methods [24]. The basic idea of SAW is to calculate a quality score for each source using a decision matrix and a vector of preference weights. Although SAW solves the problem of the heterogeneity of quality criteria by scaling their values, this method ranks sources considering only the user quality preferences weights. This ranking is based on the priority and importance of quality criterion but does not consider the preferences values. Consequently, we could not select the best sources unless the user defines a limit of the acceptable scores or a number of sources. To overcome these limitations, we develop a selection and ranking algorithm that respects both the user quality preferences weights and values. Our algorithm is performed in two stages: 1) source selection which consists of filtering sources according to user’s quality values. Only the quality criteria having the highest weight are involved during
this phase, 2) source ranking using SAW method which calculates for each source a score based on all quality criteria. The quality matching algorithm is given in the following.

Input:  
$S = \{S_1, S_2, ..., S_n\}$: Set of candidate sources  
$Q = \{Q_1, Q_2, ..., Q_m\}$: set of sources quality metrics.  
$M = [v_{ij}] (n \times m)$: the decision matrix, where $v_{ij}$ is the value of $Q_j$ measured on source $S_i$  
$W = [w_i]:$ the vector of user quality preference weights  
$Q(t_{Qi})$: Quality threshold defined by user for each $Q_i$

Output:  
$S' = \{S'_1, S'_2, ..., S'_k\}$: Set of selected and ranked sources

// Stage 1: Source Selection

1. for all $Q_i$ select the one having the highest weight and call it $Q_{max}$  
2. from $S$, select $S_i$ having $Q_{max}$ value e’” $Q(t_{Qi})$ (Qmax)

// Stage 2: Sources Ranking using SAW Algorithm

3. Scale $v_{ij}$ to make them comparable using some transformation function. With this scaling all source’s quality values are in $[0, 1]$. We obtain a scaled decision matrix $M' = [v'_{ij}] (n \times m)$ where:

$$v'_{ij} = \frac{v_{ij} - \min(v_{ij})}{\max(v_{ij}) - \min(v_{ij})}$$

4. Apply $W$ to $M'$

5. Calculate sources scores; the score of source $S_i$ is given by:

$$Score(S_i) = \sum_{j=1}^{m} v'_{ij} \cdot w_j$$

6. Rank sources according to the sources scores obtained in step 3.

End

6. Related Works

To the best of our knowledge, few mediation systems deals with personalization and proposes solution for dynamic adaptation to user profile. Recent works on personalizing data integration systems can be classified according to three approaches: The first one is about extending querying languages to support user preferences. New operators have been proposed to integrate user preferences in the query such as skyline[26], best [27], winnow[28], prefer[29], and the Preference SQL language[30]. The second approach involves the reformulation of queries using enrichment and rewriting techniques. For example, the APMD project[31] proposes a personalization platform for accessing multiple databases. This platform plays the role of a mediator which first enrich the user query using elements from user profile, executes the query rewriting process to generate sub queries, these latter are enriched once again before being sent to the targeted sources [32]. Although this solution enables personalizing results, it remains costly in term of response time and bandwidth. Moreover, this technique is not applicable for other mediators. The last approach concerns optimizing query evaluators. Take for example the QBF framework[33] which proposes a method to build adaptive query evaluators. Contrary to previous literature where user preferences and profile are used in a single phase (formulating, enrichment, rewriting or optimization), our solution integrates the user profile upstream and downstream of the mediator results. With this, the personalization layer could be used in a large variety of mediation systems. Furthermore, the layer is built on several levels, which makes it extensible. Last but not least, the layer offers recommendation services neither proposed by APMD nor QBF.

7. Case Study: WASSIT

To illustrate our solution, we consider a set of bibliographic sources integrated by the mediation system WASSIT [34]. In this section, we present first the WASSIT mediation framework. Then, we explain through a running example, how the personalization layer selects the most relevant sources according to user profile.

7.1 The Mediation Framework WASSIT

Developed by our laboratory, WASSIT is a mediation framework that relies on the well-known mediator architecture [1]. It is mainly made up of two components: Mediator which is the query processing core of the framework WASSIT and Wrappers which translate the mediator requests so that they are understood by the sources. The Mediator is mainly responsible of the Query rewriting process. WASSIT supports rewriting following the most popular approaches which are LAV (Local as View) [34] and GAV (Global as View) [15]. In this case study, we adopt a GAV approach where the global schema is expressed in terms of the local schemas. Concretely, WASSIT’s mediator takes as input a user query, and produces a query execution plan (QEP) which is presented in the form of a tree. It contains the sub queries to be sent to the corresponding sources via the associated wrappers and the reconstruction operators (for example: union and jointure) that will be used to integrate results delivered by the wrappers. After combining results, WASSIT delivers a final response to the user; it neither differentiates between users, nor adapts responses to each profile. Furthermore, the mediator involves all sources figuring in the execution plan. This is costly because the response time of the mediator depends on the number of integrated sources. By inserting the personalization layer between the user and the mediator, users’ profiles are used to select a reduced set of relevant sources that will be involved in the query rewriting process. Let us consider a running example.

7.2 Running Example

In this example, WASSIT integrates, 13 bibliographic sources dealing with computer science domain such that: “$S_1$” to “$S_7$” contains documents concerning data integration; “$S_8$”, “$S_9$” and “$S_{10}$” contains all documents...
with their prices; 

S11", "S12", and S13 contains all documents reviews.

The global schema used by the mediator according to the GAV approach is the following:

Documents: (id, title, author, abstract, key words, editor, year, price, type)

Price: (id, title, price, editor, year)

Eval: (id, title, author, year, evaluation, score)

7.2.1 User Interaction

Consider two users U1 and U2 with different profiles. U1 is a student who prefers popular sources (popularity >3), with completeness score 40%. His session goals are: rewriting, personalization, and optimization. U2 is a professor preferring recent publications published in the last 3 years, in most popular journals (popularity >3). U2 session goals are mediator, wrapper, rewriting, Bucket algorithm. Both users enter the same query Q: "looking for evaluated documents dealing with data integration field and ordered by increasing prices".

WASSIT executes Q and generates the QEP in figure 1:

As we can see, both users will have the same response because they didn't express their preferences in their query. The QEP is the same; it involves all sources that contain documents about data integration without making any distinction between U1 and U2 profiles.

7.2.2 Using the Personalization Layer

To personalize WASSIT’s responses, let’s use the personalization layer in 4 steps which are:

• Step 1: representing user profiles

We suppose that users have entered their persistent profiles while creating their account. In the following, we only represent in table 1 the session profiles.

<table>
<thead>
<tr>
<th>User goals</th>
<th>Quality preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1 rewriting, 0.9; personalization, 0.8; optimization, 0.7; data integration, 0.6</td>
<td>Reputation (score) 0.4; Timeliness (years) 0.1; Completeness (%) 0.3; Resptime (s) 1</td>
</tr>
<tr>
<td>U2 mediator, 0.9; wrapper, 0.8; rewriting, 0.7; Bucket, 0.5; data integration, 0.6</td>
<td>Reputation (score) 0.3; Timeliness (years) 0.3; Completeness (%) 0.1; Resptime (s) 1</td>
</tr>
</tbody>
</table>

Table 1. Users profiles: Session goals and quality preferences

• Step 2: representing sources profiles

We only give in table 2 an illustrative example of two sources profiles.

<table>
<thead>
<tr>
<th>S1 Identity</th>
<th>S2 Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id = 1, Name, ConfProc</td>
<td>Id = 2, Name: BibComputer</td>
</tr>
<tr>
<td>URL: <a href="http://www.conferecnesproc.gov.ma">www.conferecnesproc.gov.ma</a></td>
<td>URL: <a href="http://BibComputer.com">http://BibComputer.com</a></td>
</tr>
<tr>
<td>Owner: Mohammed V University</td>
<td>Owner: EMI School digital library</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S1 Content</th>
<th>S2 Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rewriting, 0.5; optimization, 0.8; mediator, 0.1; ontology, 0.4; classification, 0.3)</td>
<td>(rewriting, 0.9; mediator, 0.8; wrapper, 0.6; Bucket, 0.7)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S1 Quality</th>
<th>S2 Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reputation= 5, Timeliness= 20 years, Completeness= 50%, Response-time= 1s</td>
<td>Reputation= 4, Timeliness= 5 years, Completeness= 40%, Response-time= 2s</td>
</tr>
</tbody>
</table>

Table 2. Example of sources profiles
• **Step3: Content-Matching**

All sources profiles, particularly, Content and Quality dimensions are involved during the source selection procedure. Due to the lack of space, we only give in Table 3 the results of executing the Content-Matching algorithm according to U1 profile.

From Table 3, we select the most relevant sources so-called m-relevant using definition 4. Suppose U1 relevancy threshold is 50%, then the ordered set of m-relevant sources is: \( S = \{ S_9 > S_{11} > S_3 > S_8 > S_6 > S_{13} \} \).

<table>
<thead>
<tr>
<th></th>
<th>Rewriting</th>
<th>Personalization</th>
<th>Optimization</th>
<th>Data-integration</th>
<th>Sim</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>0.9</td>
<td>0.8</td>
<td>0.7</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>0.5</td>
<td>0</td>
<td>0.8</td>
<td>0</td>
<td>0.596</td>
<td>5</td>
</tr>
<tr>
<td>S2</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.433</td>
<td>8</td>
</tr>
<tr>
<td>S3</td>
<td>0.2</td>
<td>0.6</td>
<td>0.4</td>
<td>0</td>
<td>0.650</td>
<td>3</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0.7</td>
<td>0</td>
<td>0.1</td>
<td>0.430</td>
<td>9</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0.276</td>
<td>11</td>
</tr>
<tr>
<td>S6</td>
<td>0.8</td>
<td>0</td>
<td>0.2</td>
<td>0.1</td>
<td>0.593</td>
<td>6</td>
</tr>
<tr>
<td>S7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
<td>0.270</td>
<td>12</td>
</tr>
<tr>
<td>S8</td>
<td>0.2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.1</td>
<td>0.643</td>
<td>4</td>
</tr>
<tr>
<td>S9</td>
<td>0.7</td>
<td>0.6</td>
<td>0.5</td>
<td>0.3</td>
<td>0.933</td>
<td>1</td>
</tr>
<tr>
<td>S10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>S11</td>
<td>0.5</td>
<td>0.25</td>
<td>0.25</td>
<td>0.3</td>
<td>0.756</td>
<td>2</td>
</tr>
<tr>
<td>S12</td>
<td>0.2</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.370</td>
<td>10</td>
</tr>
<tr>
<td>S13</td>
<td>0.9</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0.532</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 3. Sources similarity scores and ranking according to U1 profile

• **Step4: Quality-Matching**

This step aims to select from S sources that respect user's quality preferences based on the Quality-Matching algorithm. Remind that this algorithm runs in two stages. In stage1, it selects sources that respect quality values required by the user for criteria having the highest weights (in our case: reputation). Consequently, sources S9 and S13 will not be selected. In stage2, it applies the SAW method and calculates the sources scores. These scores are presented in table 4.

We apply the same method for U2 profile and we Got the following set: \( S = \{ S_2 > S_{10} > S_{11} > S_{13} \} \).

From these results we can observe that the ordered sets of relevant sources for U1 and U2 are different. So the mediator generates different QEPs according to each user profile. Figure 3 presents these personalized QEPs.

<table>
<thead>
<tr>
<th></th>
<th>Reputation (score)</th>
<th>Timeliness (years)</th>
<th>Completeness (%)</th>
<th>Response Time (s)</th>
<th>Source Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>5</td>
<td>20</td>
<td>50</td>
<td>1</td>
<td>0.572</td>
</tr>
<tr>
<td>S3</td>
<td>5</td>
<td>30</td>
<td>80</td>
<td>1</td>
<td>0.833</td>
</tr>
<tr>
<td>S6</td>
<td>3</td>
<td>2</td>
<td>60</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>S8</td>
<td>4</td>
<td>5</td>
<td>40</td>
<td>2</td>
<td>0.31</td>
</tr>
<tr>
<td>S9</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>1</td>
<td>not-selected</td>
</tr>
<tr>
<td>S11</td>
<td>4</td>
<td>5</td>
<td>40</td>
<td>0.5</td>
<td>0.21</td>
</tr>
<tr>
<td>S13</td>
<td>2</td>
<td>4</td>
<td>60</td>
<td>2</td>
<td>Not-selected</td>
</tr>
</tbody>
</table>

Table 4. Sources scores using quality matching algorithm

The personalized QEP involves only 4 sources for U1 and 6 sources for U2 instead of 13 sources in the non personalized QEP. This reduction optimizes responses times. Indeed, it has been proved in [20] that the increasing number of data sources leads to a bottleneck and degrades considerably the mediator performances.

8. Conclusion

This paper presents the personalization layer whose objective is to adapt a mediation system to the users’ profiles. This layer is composed of four levels; the user level where user profile is managed, the source level where
sources profiles are created, the application level where personalization functions are executed, and the knowledge base level where data are stored. The layer offers diverse personalization functions according to a cyclic process. We focus in this work on building a multidimensional user profile, and source profile and matching them with the aim of selecting the most relevant sources that meet user information goals and respects his quality requirements in the same time. Our source selection procedure allows adapting query execution plans to users’ profiles while reducing the number of involved sources. This is a key issue for optimizing mediation systems. Indeed, it has been proved in [20] that the response time increases considerably with the number of integrated sources.

To illustrate our solution, we present results of a case study where we use the personalization layer to personalize the WASSIT mediation framework. Until now, integrating other functionalities like measuring user feedback and recommendation services are in progress. Another ongoing work is to adapt user queries according to user relevancy feedback. The next step is to validate our approach considering performances (cost, precision, recall, etc).

References


