A Contextualization Service for a Personalized Access Model

Sofiane ABBAR*, Mokrane BOUZEGHOUB*,
Dimitre KOSTADINOV**, Stéphane LOPES*

*Laboratoire PRiSM, Université de Versailles,
45 avenue des Etats-Unis, Versailles, 78035
firstname.lastname@prism.uvsq.fr,
**Alcatel Lucent Bell Labs Villarceaux
Route de Villejust, 91620 Nozay, France
dimitre.kostadinov@alcatel-lucent.fr

Abstract. Personalization paradigm aims at providing users with the most relevant content and services according to many factors such as interest center or location at the querying time. All this knowledge and requirements are organized into user profiles and contexts. A user profile encompasses metadata describing the user whereas a context groups information about the environment of interaction between the user and the system. An interesting problem is therefore to identify which part of the profile is significant in a given context. This paper proposes a contextualization service which allows defining relationships between user preferences and contexts. Further, we propose an approach for the automatic discovery of these mappings by analyzing user behavior extracted from log files.

1 Introduction

Personalization paradigm aims at adapting applications as much as possible to the user preferences and to the user context. Adaptation may concern several aspects, such as system reconfiguration, communication protocols, data sources selection, query reformulation, data layout, or users feedback handling. Data personalization refers to the set of techniques which allow providing users with the most relevant content. There exist two approaches for adapting and customizing application interactions: User Centric Personalization and Context-Aware Application.

Considering only one of the previous approches may not be satisfactory for many applications. Indeed, the same user, with different profiles, may prefer listening news during breakfast and listening Rn'B music while driving a car. Alternatively, the same user, at his home context, may have different domains of interest related to his hobbies or to his job. Thus, allowing applications to combine both approaches leverages their adaptability to the benefit of the users.

The goal of this paper is to show, through the definition of a specific service called contextualization, how a Personalized Access Model (PAM) can operate on both profile and context. Given a profile model and a context model, Contextualization is defined as a cross-filtering process, run periodically over the user's interaction log file to extract possible associations between profile and context instances (called contextual preferences). The contextualization service is part of a set of services composing the PAM and aiming to provide high level mechanisms to build personalized applications.

This paper is organized as follows. Section 2 gives an overview of the PAM architecture and services. Section 3 defines the contextualization service. Section 4 reports on related works. Section 5 concludes the paper with current results and further research.

2 PAM: The Personalized Access Model

Our definition of a personalized access model aims to provide a generic set of concepts and techniques which can be deployed over a given architecture to make applications adaptable to users' profiles and contexts. The figure 1 gives an overview of the main components of a PAM.

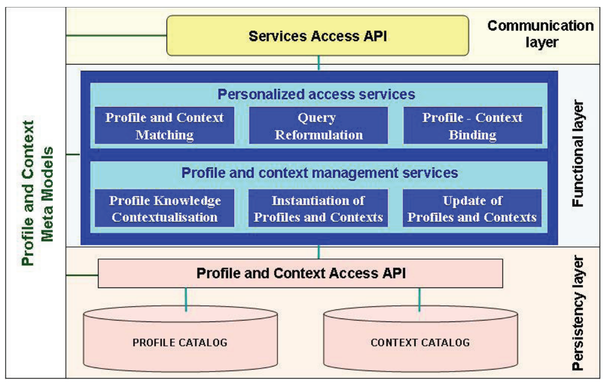

FIG. 1 – *Personalized Access Model Architecture*

The *persistency layer* deals with the storage and the access to the profiles and to the contexts. It includes the profile and the context catalogs. The *functional layer* is composed of services for profile and context management, and personalized access services. Finally, the *communication layer* provides a communication interface between the PAM and users or applications. The role of this layer is, on one hand, to give access to the profiles and the contexts database and, on the other hand, to enable calling the PAM services. The components of the three layers of the PAM are built around profile and context meta models which are generic enough to be adapted to a wide range of applications and which are open to integrate specific knowledge not included initially. More details on the PAM can be found in Abbar et al. (2008).

3 The Contextualization Service

The contextualization service aims at identifying relationships between user preferences and contexts. To achieve this task, the process we propose takes as input the user behavior captured in log files. This process is divided into three steps: *result partitioning*, *mapping initialization*, and *context specificity check* (see figure 2).

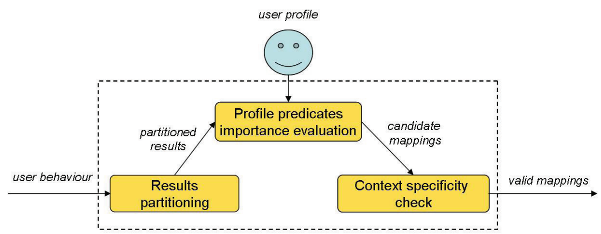

FIG. 2 – *Global profile contextualization process*

3.1 Notation and Problem definition

A user profile is a set of preferences $P_u = \{p_1, \ldots, p_n\}$. Each preference p_i is composed of a predicate pr_i and a weight w_i, i.e. $p_i = (pr_i, w_i)$. The weight w_i is a real number expressing the importance of the predicate pr_i for the user. The predicate characterizes a given concept. It is a triplet <*concept, operator, value*>, e.g. *Genre = 'Drama'*.

A context c_j is defined as a set of pairs *(attribute, value)*, e.g. *location = 'Paris'*. We consider that a context is represented by its identifier c_j. The set $\mathcal{C}$ is the set of contexts which are of some interest for the application, i.e. $\mathcal{C} = \{c_1, \ldots, c_m\}$.

For a given user whose profile is P_u, the profile contextualization consists in specifying a set $\mathcal{M}$ of mappings which relates the user preferences with the contexts of $\mathcal{C}$:

$$\mathcal{M}(P_u, \mathcal{C}) = \{m(p_i, c_j, \delta) \mid p_i \in P_u, c_j \in \mathcal{C}, \delta \in [-1, 1]\}$$

where δ is the mapping score of p_i with respect to the context c_j.

According to the value of δ, there are three interpretations of the mapping $m(p_i, c_j, \delta)$: a *positive mapping* ($\delta > 0$) indicates that the predicate pr_i has to be satisfied in the context c_j, a *negative mapping* ($\delta < 0$) points out that the predicates pr_i would not be satisfied in the context c_j, and a *neutral mapping* ($\delta = 0$) specifies that the predicate pr_i is irrelevant in the context c_j.

Mappings are discovered by analyzing the user history $\mathcal{H}$ captured in log files. The profile contextualization process takes as input a non-contextualized user profile P_u, the user history $\mathcal{H}$ and the set $\mathcal{C}$ of contexts handled by a given application. It returns the set of mappings $\mathcal{M}(P_u, \mathcal{C})$.

The remaining of this section details each steps of the process as presented in figure 2.

3.2 Result Partitioning

This first step consists in identifying which content the user liked and which he disliked. This phase produces two sets containing respectively relevant contents and irrelevant contents for each context. In the following, $POS(P_u, c_j)$ and $NEG(P_u, c_j)$ denote respectively the set of relevant and irrelevant contents in the context c_j for a user profile P_u.

The identification of relevant and irrelevant contents in the log is done by analyzing actions that the user performed on those contents. Further, relevance of contents in the POS and NEG sets may differ with regard to the type of the action applied to them. In POS set for example, a bought product is more important for a given user than visualized one even if user spent a lot of time on it. Thus we propose to organize all possible actions in a relational table which describes actions with their eventual weights. Actions with positive weights are said positive actions and contents on which they are performed are considered as being relevant to the user, whereas actions with negative weights are said negative actions, and contents on which they are applied are considered as being irrelevant.

3.3 Mappings initialization

The second step consists in initializing the possible mappings between the user profile predicates and contexts. This is done by computing the occurrence frequencies δ_{ij}^{+} and δ_{ij}^{-} of each profile predicate in $POS(P_u, c_j)$ and $NEG(P_u, c_j)$ respectively. The value of δ_{ij} depends on whether the frequency δ_{ij}^{+} (respectively δ_{ij}^{-}) satisfies a given condition $\gamma^{+}(\delta_{ij}^{+})$ (respectively $\gamma^{-}(\delta_{ij}^{-})$ as for instance a threshold.

If only $\gamma^{+}(\delta_{ij}^{+})$ holds, i.e. a sufficient part of the relevant results satisfies the predicate, a positive mapping $m(p_i, c_j, \delta_{ij}^{+})$ is created. Similarly, if only $\gamma^{-}(\delta_{ij}^{-})$ holds, a negative mapping $m(p_i, c_j, \delta_{ij}^{-})$ is created. If both $\gamma^{+}(\delta_{ij}^{+})$ and $\gamma^{-}(\delta_{ij}^{-})$ are not satisfied, a neutral mapping $m(p_i, c_j, 0)$ is created. Finally, if both $\gamma^{+}(\delta_{ij}^{+})$ and $\gamma^{-}(\delta_{ij}^{-})$ hold, the predicate corresponds to a user preference which is always valid. Thus, a positive mapping $m(p_i, c_j, \delta_{ij})$ is created with δ_{ij} equal to the average of p_i frequencies in the set union of relevant and irrelevant results ($\delta_{ij} = freq(p_i, POS(P_u, c_i) \cup NEG(P_u, c_i))$).

3.4 Context specificity check

This step prunes mappings which relate the same profile predicate to all contexts. These mappings correspond to the last case discussed in section 3.3 and concern profile predicates which have not to be contextualized as they have to be considered in all contexts. For example if the user watches only movies in English, then all relevant results will satisfy the predicate “language="English"”, but the same will be true for all irrelevant results too. This kind of predicates has to be taken into account in all contexts as it corresponds to a general user preferences. The result of this step is a set of valid mappings between the user profile and contexts.

3.5 Mapping construction algorithm

Discovering automatically contextual preferences is a complex task. Therefore, we made a simplifying hypothesis for the design of the algorithm : *All user actions express positive preferences and are of the same importance*. As explained before, the algorithm takes as input a user profile P_u, logs of user $\mathcal{H}$, a set of contexts $\mathcal{C}$ and a threshold γ for mapping validation. The main step of the algorithm is to compute the frequency in the log of each user predicate for each context. Then, following the value of those frequencies, mappings will be created.

If the predicate has a occurence frequency that satisfies γ in all contexts, then a positive mapping $m(p_i, \mathcal{C}, AVG(F))$ is created, attesting that predicate pr_i holds in all contexts of the set $\mathcal{C}$, with $AVG(F)$ the average of all frequencies of apparition. This predicate is always satisfied independently of the context, thus, it can be seen as a user view characteristic.

Algorithm 1 Automatic mapping discovery

Require: the user profile $P_u = \{p_1, \ldots, p_n\}$, the user behavior $\mathcal{H}$, the set of possible contexts $\mathcal{C} = \{c_1, \ldots, c_m\}$, the threshold γ.
Ensure: the set of contextualization mappings $\mathcal{M}$

$\mathcal{M} \leftarrow \emptyset$
for all $p_i \in P_u$ **do**
  Compute $F = f_1, \ldots, f_M$ such as $f_k = Freq(pr_i, c_j, \mathcal{H})$ {where $Freq(pr_i c_j, \mathcal{H})$ is the frequency of the predicate pr_i in $\mathcal{H}$ when the context is c_j.}
  if $\exists i, j \mid f_i \geq \gamma \wedge f_j < \gamma$ **then**
    for all $f_k \in F$ **do**
      if $f_k \geq \gamma$ **then**
        $\mathcal{M} \leftarrow \mathcal{M} \cup m(p_i, c_k, f_k)$ {the predicate holds in the context}
      else
        $\mathcal{M} \leftarrow \mathcal{M} \cup m(p_i, c_k, 0)$ {the predicate does not hold in the context}
  else
    if $\forall f_k \in F, f_k < \gamma$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \cup m(p_i, \mathcal{C}, 0)$ {predicate which does not hold in any context}
    else
      $\mathcal{M} \leftarrow \mathcal{M} \cup m(p_i, \mathcal{C}, AVG(F))$ {predicate that holds in all contexts (user view)}
return M

If any occurence frequency of a given predicate satisfies γ, then a neutral mapping $m(p_i, \mathcal{C}, 0)$ is created attesting that the predicate pr_i is not contextual. In other words, the predicate is not relevant for contexts of the set $\mathcal{C}$. That is the case of rarely (occasionally) used predicates.

If the predicate is frequent in some contexts only, the predicate is contextual. Indeed, it is relevant to take it into account in some contexts, then mappings $m(p_i, c_k, f_k)$ are created for each of these context with the correspondent frequency, and to ignore it in the others where neutral mapping $m(p_i, c_k, 0)$ are created for each of the remainder contexts.

4 Related Works

Recently, contextualization of user profile and preferences has attracted attention. In (Holland and Kiessling, 2004) authors propose a framework for contextual preferences, called situated preferences. In this approach, both user profiles and situations are modelled in an Entity-relationship model, then, contextualization of preferences is modelled uniquely as M:N relationship (pid, sid), expressing that the preference pid holds in the situation sid. No detail are given about relationship construction. In their research, (Stefanidis and Pitoura, 2008) introduced context into the database field. The context which is a set of contextual attribute (e.g. Age, weather) is used to rank database tuples w.r.t a given query. Contextual preferences are explicitly specified by users and are of the form <contextState, preferencePredicate, Score>. The difference with our proposition, is that authors consider a user profile as being a part of the context while we clearly separated these two concepts, this may be due to the different point of view we consider. An automatic discovery of contextual preferences was initiated by (Agrawal et al., 2006). In (Bunningen et al., 2007) contextual preferences are also automatically generated by observing and analyzing user histories. Preference are modeled throught description-logical frame, and the discover of contextual preferences is done automatically by calculating the probability that an element be the best one in a particular context.

5 Conclusion

In this paper, we have presented a personalization service, called contextualization, which allows finding dependencies between user profiles and contexts by exploiting user-application interaction logs. The service can be used at design time or periodically as a maintenance service which keeps up to date the semantic links between profiles and contexts for a given application. The contextualization service is a service among several others which constitute a coherent personalized access model (PAM). Further research will focus on the binding service which exploit, at execution time, the mappings generated by the contextualization service to make user queries sensitive to user profile, to interaction context or to both of them.

References

Abbar, S., M. Bouzeghoub, D. Kostadinov, S. Lopes, A. Aghazaryan, and S. Betge-Brezetz (2008). A personalized access model: Concepts and services for content delivery platforms. In *Actes des 24emes Journées Bases de Données Avancées, Guilherand-Granges, France*.

Agrawal, R., R. Rantzau, and E. Terzi (2006). Context-sensitive ranking. In *ACM SIGMOD international conference on Management of data, Chicago, IL, USA*, pp. 383–394.

Bunningen, A. H. V., M. M. Fokkinga, P. M. G. Apers, and L. Feng (2007). Ranking query results using context-aware preferences. In *ICDEW'07, Istanbul, Turkey*, pp. 269–276.

Holland, S. and W. Kiessling (2004). Situated preferences and preference repositories for personalized database applications. In *Conceptual Modeling - ER 2004*, pp. 511–523.

Stefanidis, K. and E. Pitoura (2008). Fast contextual preference scoring of database tuples. In *11th international conference on Extending database technology, Nantes, France*, pp. 344–355.

Résumé

La personnalisation vise à fournir à l'utilisateur les contenus les plus pertinents par rapport à divers critères. Ces informations sont organisées en profils utilisateur et en contextes d'interaction. Un problème intéressant est alors de déterminer quelle part du profil est significative dans un contexte donné. Cet article propose un service de contextualisation qui permet de définir les relations entre les préférences utilisateurs et les contextes. Nous proposons également une approche automatique pour la découverte de ces relations en analysant les traces du comportement de l'utilisateur.