A Context-Aware Preference Model for Database Querying in an Ambient Intelligent Environment

Arthur H. van Bunningen, Ling Feng, and Peter M.G. Apers
Centre for Telematics and Information Technology, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
{bunninge, ling, apers}@cs.utwente.nl

Abstract. Users’ preferences have traditionally been exploited in query personalization to better serve their information needs. With the emerging ubiquitous computing technologies, users will be situated in an Ambient Intelligent (AmI) environment, where users’ database access will not occur at a single location in a single context as in the traditional stationary desktop computing, but rather span a multitude of contexts like office, home, hotel, plane, etc. To deliver personalized query answering in this environment, the need for context-aware query preferences arises accordingly. In this paper, we propose a knowledge-based context-aware query preference model, which can cater for both pull and push queries. We analyze requirements and challenges that AmI poses upon such a model and discuss the interpretation of the model in the domain of relational databases. We implant the model on top of a traditional DBMS to demonstrate the applicability and feasibility of our approach.

1 Introduction

With the coming anytime/anywhere ubiquitous data access paradigm in an AmI environment, context plays an important role in information delivery and dissemination. Database and recommendation systems nowadays are more and more aware of the context while serving users’ information needs. In this paper, we investigate user’s query preferences in an AmI environment, taking their applicable contexts into account. Our design of the context-aware preference model is influenced and guided by the following AmI philosophies:

– Smartness requirement. The smartness requirement in an AmI environment implies reasoning and learning capabilities that the preference model must possess, calling for an inevitable knowledge ingredient. For example, a user may input a preference like prefer a nearby restaurant when the weather is bad. With the model, it should be able to infer the applicability of the preference no matter whether it rains or snows, since both are bad weather.

– Proactiveness requirement. Following the smartness requirement, the database systems in an AmI environment should proactively deliver anytime/anywhere useful information to their users. The designed context-aware query preference model should therefore provide sufficient flexibility and adaptiveness to the two access modes, namely pull query where users actively query databases to pull relevant information, and push query where the systems push proactively possibly relevant
information to users (e.g., querying the background information about a person when s/he enters a room).

– **Closure requirement.** To support preference propagation and deduction, the model should preferably possess the closure property so that the output preference can serve as the input context of some other preferences. For instance, suppose a user has two preferences: “prefer cheerful TV programs when having a bad mood” and “prefer channel 5 if looking for cheerful TV programs” As a consequence, when this user has a bad mood, _cheerful TV programs on channel 5_ will be the most preferable program alternatives for him/her.

– **Scalability requirement.** Performance is highly demanded at data management level to process real-time queries raised by different users anytime/anywhere in an AmI environment. The context-aware preference model must be easily interpreted and executed in a database world to achieve scalability.

– **Traceability requirement.** The behaviors of database querying systems in an AmI environment, and thus the preference model should be traceable by the users. In other words, it should be possible for a human to conveniently enter, view, and edit context-aware preferences in a way which is close to the world model of the users. An intuitive user-friendly interface for preference declaration is therefore needed.

In the following sections, we first review some closely related work in Section 2. We present our knowledge-based context modeling approach, followed by the context-aware query preference modeling using Description Logics in Section 3. We depict a framework for implanting this model on top of a traditional DBMS, and interpret the model in a relational database in Section 4. The implementation of the model in serving pushing queries is illustrated in Section 5. We conclude the paper in Section 6.

### 2 Related Work

The notion of preference query was first introduced to the database field in [1]. It extended the Domain Relational Calculus to express preferences for tuples satisfying certain logical conditions. Since its introduction, extensive investigation has been conducted, and two main lines of approaches have been formed in the literature to deal with users’ preferences, namely, quantitative and qualitative [2]. The qualitative approach intends to directly specify preferences between the tuples in the query answer, typically using binary preference relations. An example preference relation is “prefer one book tuple to another if and only if their ISBNs are the same and the price of the first is lower.” These kinds of preference relations can be embedded into relational query languages through relational operators or special preference constructors, which select from their input the set of the most preferred tuples (e.g., winnow [2], PreferenceSQL BMO [3], and skyline [4]). The quantitative approach expresses preferences using scoring functions, which associate a numeric score with every tuple of the query. Then tuple $t_1$ is preferred to tuple $t_2$ if and only if the score of $t_1$ is higher than the score of $t_2$. A framework for expressing and combining such kinds of preference functions was provided in [5]. [6] presented a more rich preference model which can associate degrees of interest (like scores) with preferences over a database schema.