Information Systems 49 (2015) 65–101

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

A principled approach to context schema evolution in a data management perspective

Elisa Quintarelli, Emanuele Rabosio*, Letizia Tanca

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, via Ponzio 34/5, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form
21 September 2014
Accepted 9 November 2014
Recommended by: F. Naumann
Available online 2 December 2014

Keywords:
Context-awareness
Schema evolution
Evolution operators

ABSTRACT

Context-aware data tailoring studies the means for the system to furnish the users, at any moment, only with the set of data which is relevant for their current context. These data may be from traditional databases, sensor readings, environmental information, close-by people, points of interest, etc. To implement context-awareness, we use a formal representation of a conceptual *context model*, used to design the *context schema*, which intensionally represents all the contexts in which the user may be involved in the considered application scenario.

Following this line of thought, in this paper we develop a formal approach and the corresponding strategy to manage the evolution of the context schema of a given context-aware application, when the context perspectives initially envisaged by the system designer are not applicable any more and unexpected contexts are to be activated. Accordingly, when the context schema evolves also the evolution of the corresponding context-aware data portions must be taken care of. The aim of this paper is thus to provide the necessary conceptual and formal notions to manage the evolution of a context schema in the perspective of data tailoring: after introducing a set of operators to manage evolution and proving their soundness and completeness, we analyze the impact that context evolution has on the context-based data tailoring process. We then study how sequences of operator applications can be optimized and finally present a prototype validating the feasibility of the approach.

1. Introduction

The technological scenario of our era has enabled an extremely large variety of information sources to become available even to casual users: all kinds of organizations collect, maintain and use terabytes of information about their customers, suppliers and operations, while, with the help of the widespread use of mobile terminals, the WWW is becoming day by day more friendly to any kind of users. The contribution of the two recent phenomena Internet of Things and Social Networking further enriches and complicates the overall scenario.

Such an extensive repository constitutes an unprecedented opportunity for users, but at the same time risks to overwhelm them; often the only way to get the gist of the available information requires that the users know exactly how to formulate the query, a difficult task when the dataset structure and meaning are not known a priori. Moreover, really large data collections simply cannot be stored in the increasingly popular portable devices, still characterized by a relatively limited amount of memory.

The literature has coped with this research challenge by proposing techniques for summarizing [1], compressing

* Corresponding author. Tel.: +39 0223993482.
E-mail addresses: elisa.quintarelli@polimi.it (E. Quintarelli), emanuele.rabosio@polimi.it (E. Rabosio), letizia.tanca@polimi.it (L. Tanca).

http://dx.doi.org/10.1016/j.is.2014.11.008
0306-4379/

[2] and analyzing Big Data [3]. In our opinion, this problem can be cleverly solved also by applying *personalization*, so that the information provided to a user is reduced on the basis of the user's personal preferences [4], on the user's current situation [5] – i.e., her *context* – or even on both aspects [6]. Note that, even with amounts of data well below our current idea of "big", personalization constitutes an important contribution to data usability.

This work considers context-based personalization. In order to reduce a large dataset on the basis of the context, conceptual *context models* have been introduced (see [7–11] for surveys), allowing to represent the context through some perspectives (dimensions): typical such dimensions are, for example, the user's current role and her location. Also more sophisticated context parameters can be introduced, like the current activity of the user, or her main interest topic. We call *data tailoring* [12] the activity of selecting, for each specific context, the relevant information: *data tailoring* refers to the capability of the system to provide the users only with the view (over an overall data representation, like for instance a global schema) that is relevant for their current context.

We use the expression *context model* to indicate the set of constructs and constraints that allow us to represent the dimensions of context and their values at a conceptual level. The activity of designing a context-aware database requires to produce a *context schema*, which exploits the constructs provided by the context model to describe the set of *dimensions* and their *values* relevant for a certain application scenario. A *context instance*, or simply a *context*, represents a particular situation, described according to a context schema.

A context schema thus represents synthetically the *structure* of the context, and as such is useful when reasoning about the assignment of a data portion (*contextual view*) to each context. For example, the work [13] presents a very effective and efficient method to automatically assign data views to all the contexts represented by the schema, only by associating one view with each context dimension value.

Throughout the paper we use a running example in the movie domain: we consider a company offering services of video on demand and reservation of cinema tickets. In this scenario, possible perspectives useful to tailor the data are the kind of user (e.g., adult, teenager or family with children), the interest topic (e.g., cinemas or movies), the situation (e.g., alone or with friends), the time (e.g., daytime or night) and the zone. The company uses context-awareness to suggest to its customers the movie(s) which are most appropriate to their current context.

The useful dimensions for data tailoring depend on the application requirements that in current systems are intrinsically dynamic and thus can evolve. Just as in the case of database schema evolution, requirement changes can be due to various reasons, including changing business needs or application and technology developments [14]: the context representation used to perform the tailoring process should thus be smoothly adapted to the evolution of requirements over time.

Consider the movie example above. The company might change its business policy, deciding to remove the distinction between daytime and evening schedule; this would lead to removing the *time* dimension from the context representation. Moreover, at a certain point marketing researches might reveal that adult customers and teenagers show the same behavior, thus making it useless to distinguish between the two user groups: then the designer might simplify the representation of the user type, merging the two categories *adult* and *teenager*. In addition, if the impact of technological changes on the considered application grows, the designer might deem it appropriate to tailor the data also on the basis of the kind of device used for the access, thus inserting a *device* dimension in the context representation. The above changes then become out-of-sync w.r.t. the previously envisaged contexts, known by the user (and by the context-aware application) at a given moment; thus the system must be able to respond to queries and applications in a seamless way, that is, a way as similar as possible to the context-aware behavior the users and applications expect. Note that studying context schema evolution is also preliminary to understanding context sharing among different users, a need that may arise in P2P scenarios [15].

As remarked already, this problem is similar to a problem of database schema evolution, where the queries designed to run on old schema versions should be maintained in order to be still applicable in the face of database schema changes. After having studied the literature on schema evolution in various fields, we propose strategies to flexibly manage the evolution of context schemas *in a data management perspective*, i.e. keeping in mind that the context is used to perform data tailoring; the context, in fact, has been employed in the literature to manage not only data, but also many other kinds of entities, including mobile sessions [16], services [17], intelligent spaces [18], etc.

The basic idea of our approach is to introduce a set of *evolution operators* to be used by the designer for modifying the conceptual context schema when necessary. These operators are so conceived as to support the evolution of the contextual (tailored) views as well. We will show in this paper that using the techniques proposed by database schema evolution for solving some typical problems of context schema evolution can turn out to be very awkward.

Indeed, our context model provides intuitive constructs and operations that afford a high level of abstraction with respect to the application scenario and to the employed technologies, and permit the management of (hierarchical) context information of various types, possibly coming from diverse sources. As pointed out in [19,20], these are fundamental features of the modern context models, thus, we believe that the results and techniques that we present in this paper are easily generalizable to other context models and implementable by means of different programming languages and logical data models, e.g. XML or some object-oriented language. Such implementations may take advantage of previous schema evolution proposals, however this operation must be performed with some caution because the semantics do not immediately correspond to each other, as highlighted in Section 9.

Goal and contributions: Built on research already published in [21] – which provides the initial formalization of