Towards an XML-Based Query and Contextual Information Model in Context-Aware Mobile Information Systems

Essam Mansour
International University in Germany
School of Information Technology
Campus 3, D-76646 Bruchsal, Germany
Email: essam.mansour@ieee.org

Hagen Höpfner
International University in Germany
School of Information Technology
Campus 3, D-76646 Bruchsal, Germany
Email: hoepfner@acm.org

Abstract

Nowadays database systems (DBSs) are state-of-the art for managing complex data in information systems. Unfortunately, DBSs are not aware of the context of their usage. Query results are retrieved without considering the context of the user/device issuing the query. In many cases, this feature is added by a context managing middle-ware. This paper outlines a scenario for context aware mobile services. In the paper we introduce an XML-based XREAL *model for formalizing contextual information and queries highlighted in the scenario and discuss how* XREAL *can be implemented in a modern database system. By utilizing* XREAL*, it is possible to realize context aware mobile information systems based on the available DBSs without additional middle-wares.*

1. Introduction

In mobile information systems (mIS), mobile clients use caching techniques to reduce the volume of transmitted data and response time [1] and to optimize the energy consumption [2]. It is a demand to store the relationships between mobile clients and data cached by these clients in order to keep the global database consistent. Queries could be stored on the server to track the relationship between the clients and the data cached by these clients, as presented in [3]. Storing these queries assists in detecting affected caches.

Context aware mIS understand the context within which their users operate [4]. In such systems, it is required to store and maintain the contextual information related to mobile clients. This contextual information assists in processing context-aware queries and maintaining the relationships between the clients and the cached data. We assume that the queries are generated in the form of relational algebra expressions.

We focus on context-aware queries that are issued by a mobile client, which is provided context-aware services by a mobile service provider. In this paper, we address the problem of modeling the contextual information related to mobile clients and their queries executed by the server. The required model needs to be flexible enough to support the heterogeneity of mobile environments and to represent several combinations of relational algebra operations. The model also demands storage and retrieval support within database systems (DBS) utilized to manage the data at the server side.

This paper presents a scenario for context-aware mobile services. Our scenario adapts the well known application scenario published by Weiser in [5] for use by mobile users and service providers. In the paper, We present an XML-based model, called XREAL, for representing and storing the contextual information related to mobile clients and queries issued by these clients.The XREAL specifications are to be stored as XML documents in modern DBS supporting XML.

The remainder of this paper is organized as the following. Section 2 highlights the related work. Section 3 presents our adapted scenario that highlights modern context-aware mobile services. Section 4 outlines the query representation. Section 5 introduces XREAL. Section 6 demonstrates the use of XREAL. Section 7 outlines a prototype system. Section 8 discusses the merits of XREAL. Section 9 concludes the paper and outlines future work.

2. Related Work

There exist many different interpretations of the term "context" in the literature [6], [7], [8]. We here use the definition by Dey [9]: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications* *themselves."* In order to utilize context information it is a must to represent it in a machine readable format. Usually this is done by ontologies or by numerical methods known from multimedia databases or geographical information systems [10]. Höpfner and Sattler in [11] have discussed the representation of context predicates as replication criterion that consists of mobile client extension.

This paper presents an approach where mobile clients can issue or pre-register queries, which depend on contextual information, on a server. At processing these queries, their current contextual information is provided to the server, so that it can be taken into account for executing the query. Our core contribution is the XREAL model that formalizes the contextual information as well as the queries, and is managed within modern DBSs.

3. Working Scenario

Mobile information systems should be aware of the context of mobile clients to enhance the query facilities and be more receptive to users' needs, such as the support for context aware or context dependent queries. We adopted the scenario proposed by Weiser in [5], and modified it to be oriented to mobile service providers and their users. We here utilized the classification of contexts in the ubiquitous computing environment proposed by Korkea-Aho in [12].

The modified scenario is: *Sal has found a mobile service provider, called Mobility, that also provides context aware services. She has registered for such services. Mobility prepared for Sal a contextual information document. This document divides the contextual information into several contexts, physical, environmental, informational, personal, social, application, and system. For the physical context, Mobility detects the location of Sal's mobile and the corresponding time of such location. The environmental context includes information related to Sal's current location or locations of interest, such as her home and work location. Examples for information related to this context are a traffic jam, parking spots, and weather.*

Sal wants to spend more time at breakfast with her children. Therefore, she asked Mobility to consider interesting business quotes in her informational context. Sal needs to know the newspaper's name, date, section, and page number combined with the quote value. She registered a query at the Mobility server to update her by such information every morning via a MMS. As soon as Sal arrives at her office, she transmits the information of this MMS to her PC.

The personal context records Sal's plan for working days and information, such as Sal likes coffee and is interested in visiting new food shops. The social context includes information about Sal's kids and neighbors. The application context includes information about her email account in order to inform her about received emails, their subject, sender and whether it has an attachment or not. Finally, the system context records information concerning systems used by Sal, such as her garage opener and heating system. Sal trusted Mobility as it is one of the companies providing a very high level of security and privacy for their customers.

Mobility at 6:30 am sends Sal a Context-Aware MMS (CA-MMS) that inform Sal that it is a gray day at the area of Sal's work (silicon valley) and there is chance of afternoon showers. Sal decides to take her waterproof jacket. This CA-MMS is a response to one of Sal's pre-registered continuous queries. On the way to work Sal received a CA-MMS, which informs her that there is a traffic jam after 5 km. Sal decides to get result for query number 7, which is one of her pre-registered queries at Mobility. In query 7, Sal requested a list of food shops close to the next exit. She immediately received a CA-MMS by a new shop at the coming exit. Sal decided to have a short visit to the new shop. After the visit, while Sal was one km away from the parking of her work, she received a CA-MMS guiding her to free parking spots.

4. Query Representation

In mobile information systems, applications generate queries and send them to the server. Therefore, there is no need to support descriptive query languages, such as SQL. Queries are to be represented in a useful way for storage and retrieval. The relational algebra representation [13] is an efficient way to represent queries over data stored in relational database. However, one can always translate SQL-queries into such expressions.

The query notation used in this paper is the notation of the relational algebra operators [13], such as selection (σ), θ-join ($\bowtie_\theta$), and projection (π). The θ-join is represented in our work as Cartesian product ($\times$) and selection (σ). As discussed by Höpfner in [14], it is possible to optimize the transformation into query trees in order to improve the support for query indexing by reducing the number of alternatives. In general and formally, a database query q can have the recursive structure [14] shown in Figure 1.

$$
\begin{array}{rcl}
q & : & \{\pi|\pi^a\}([\sigma]([\rho](R))) \\
q & : & \{\pi|\pi^a\}([\sigma](\rho(q))) \\
q & : & \{\pi|\pi^a\}([\sigma](cp)) \\
cp & : & \{[\rho](R)|\rho(q)\} \times \{[\rho](S)|\rho(q)|cp\} \\
q & : & \{\pi|\pi^a\}(q\{\cup|-|\cap\}q)
\end{array}
$$

Figure 1. A recursive relational algebra structure

Sal's query 7 retrieves name, street and telephone of shops that are around Sal's current location. It is assumed that Sal was in the area, whose postal_code is 76646, when she asked *Mobility* to send the result of query 7. In relational algebra query 7 could be $\pi_{ShopName,tele,street}(\sigma_{status='NEW' \wedge postal\_code=76646}$ $(shop \bowtie_{PID=ID} location))$. However, there are several semantically equivalent relational algebra expressions for each query, each corresponding to a query tree.

$$
\begin{array}{lll}
QS & \leftarrow & \pi_{ShopName,tele,PID}(\sigma_{status='NEW'}(shop)) \\
QL & \leftarrow & \pi_{street,ID}(\sigma_{postal\_code=76646}(location)) \\
Q & \leftarrow & \pi_{ShopName,tele,street}(\sigma_{PID=ID}(QS \times QL))
\end{array}
$$

Figure 2. A recursive structure algebra of query 7

We push selection and join operations inside a join using the algebraic properties for query optimization [13]. Then, we convert the θ-join to a selection (σ) over the Cartesian product ($\times$). Figure 2 depicts a unified algebra of query 7. Besides reducing the number of alternative trees, this way of representation groups the operations with a relation, to which these operations are applied, as a sub-query. The original projected attributes are kept.

5. The XREAL Model

XREAL (**X**ML-Based **Re**lational **Al**gebra) is an XML-based model for the contextual information related to mobile clients and queries issued by these clients. The XREAL model consists of two main components, *mobile client* and *query*.

5.1. The XREAL Model of a Mobile Client

The *mobile client* specification represents a particular mobile client and its contextual information. In this paper, it is assumed that mobile service providers detect the contextual information and formalize it using the XREAL model. The *mobile client* component consists of an identification attribute, called *MCID*, and a sequence of elements (*physical context, environmental context, informational context, personal context, social context, application context,* and *system context*). Figure 5 shows the XML schema of *mobile client* at an abstract level. Any *mobile client* is assigned a *MCID* number, to be recognized by the system. *Physical context* provides information related to location and time. The location is a position, elevation, and direction. The position could be represented using a geographical coordinates and/or relative coordinates, such as a street, area and city. The time represents time zone, which could be inferred from the location information. The time zone determines the absolute time, day, week, month, quarter, and year. *Physical context* might help to infer information at a generic level related to *environmental context*, such as weather, light and sound level. However, other methods are needed to determine an accurate environmental information.

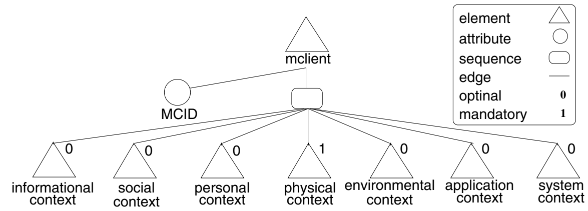

Figure 3. The XML Schema of the *mobile client*

Informational context formalizes information of interest to the mobile client, such as currency rates, stock quotes and sports scores. *Personal context* specifies information such as health, mood, biographical information, habit and activities. *Social context* formalizes information concerning group activity and social relationships. *Application context* models information, such as email received and websites visited. The *system context* represents information related to systems used by the client and specs of her mobile, such as processor, and memory capacity.

The user of a mobile client might provide personal and social information to be recorded as contextual information related to her mobile client. It is assumed that the minimum level of information is the information of *physical context*. So, the *physical context* element is a mandatory element. The other elements are optional. Furthermore, it is assumed that there is a repository of contextual information related to the environment, in which mobile clients are moving, such as parking spots or food shops.

5.2. The XREAL Model of a Query

The XREAL model formalizes a relational algebra query as a *query* element that consists of two attributes, *QID* and *MCID*, and a sequence of elements, *relations*, *projection* and *join*. Figure 4 shows the XML schema of XREAL *query*. The *QID* attribute represents a query identification. The *MCID* attribute represents the identification number of a mobile client that issued the query. A query might access only one relation. Therefore, a *query* element contains at least a *relations* element and *projection* element, and might have a *join* element. The query provides a formalization for queries represented as discussed in Section 4.

The *relations* element is composed of a sequence of at least one *relation* element. The *relation* element consists of an identification attribute, called *RID*, and a sequence of elements, *name, rename, selections* and *rprojection*. The *name* element represents the relation name. The *rename* element denotes the temporally name used to refer to the relation in the query. The

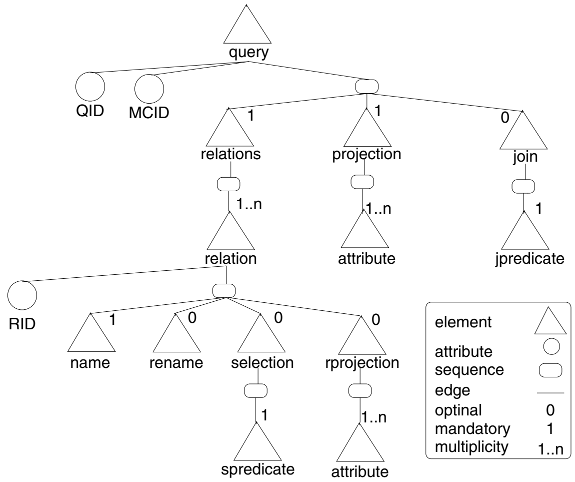

Figure 4. The XML Schema of the *query*

selection element is composed of a sequence of a *spredicate* element of type *predicateUDT*. The *rprojection* element consists of a sequence of at least one *attribute* element of type *attributeUDT*.

The *predicateUDT* type is a complex type that is able to represent simple predicate or composite predicate. The *attributeUDT* type is a complex type composed of attribute, called *ofRelation*, and a sequence of elements, *name* and *rename*. The *ofRelation* attribute represents a relation ID, to which the attribute belongs. The *name* element denotes the name of the attribute. The *rename* element represents the new name assigned to the attribute in the query. The *projection* element is similar to the *rprojection* element, but *projection* represents the original projected attributes used in the query. The *join* element specifies the join predicates used to join together the relations (sub-queries).

```
<mclient MCID="MC101">
  +<physical>
  +<system>
  +<application>
  +<environmental>
  +<personal>
  +<social>
  +<informational>
</mclient>
```

Figure 5. The XREAL document for a mobile client

6. A Formalization for the Scenario

The XREAL model provides an XML representation for the contextual information of the mobile clients and their queries. Figure 5 shows the contextual information document specified using XREAL that is generated by *Mobility* for Sal as discussed in Section 3. Sal's document is assigned *MC101* as an ID.

A	B
<relative_position> <country>Germany</country> <city>Bruchsal</city> <area>south</city> <street>Durlacher<street> <postal_code>76646</postal_code> </relative_position>	<quote> +<value> +<newspaper> +<section> +<date> +<description> </quote>

Figure 6. A) Part of the physical context, B) Part of the informational context

```
<query QID="QID1" MCID="MC101">
 <relations>
  +<relation RID="RID01">
  +<relation RID="RID02">
 </relations>
 +<projection>
 <join>
  <jpredicate>
   <simplePredicate>
    <attribute ofRelation="RID01">
     <name>PID</name>
    </attribute>                 ↪
<operator>eq</operator>
<operand>
 <attribute
 ofRelation="RID02">
  <name>ID</name>
 </attribute>
</operand>
</simplePredicate>
</jpredicate>
</join>
</query>
```

Figure 7. An XREAL specification for query 7

```
<relation RID="RID01">
 <name>shop</name>
 <selection>
  <spredicate>
   <simplePredicate>
    <attribute>
     <name>status</name>
    </attribute>
    <operator>eq</operator>
    <operand>
     <value>'NEW'</value>
    </operand>
   </simplePredicate>
  </spredicate>
 </selection>                 ↪
<projection>
 <attribute>
  <name>ShopName</name>
 </attribute>
 <attribute>
  <name>tele</name>
 </attribute>
 <attribute>
  <name>PID</name>
 </attribute>
</projection>
</relation>
```

Figure 8. The specification of the relation *shop*

Figure 6 depicts part of the *physical* and *informational* contexts of the document. Figure 7 illustrates an overview of the XREAL specification for the query 7 shown in Figure 2. This specification consists of a *query* element. The query ID is *QID1* and is issued by a mobile client, whose ID is *MC101*. There are two relations (*shop* and *location*), which are joined together using one join predicate. Figure 8 illustrates the XREAL specification for the relation (*shop*), whose ID is *RID01*, used in the query 7. This specification consists of a *relation* element, whose name is *shop*. There is a selection operation over the relation. This selection predicate is to select the shops, whose status is *NEW*. There is also a projection operation that picks the attributes (*ShopName,tele* and *PID*).

7. A Prototype System

We have utilized DB2 Express-C 9.5 and the Sun Java 1.6 language to implement XREAL, and built-in functions within DB2 for update notification and the context-aware query processing. Concerning our method for update notification based on XREAL, the reader is referred to [3].

7.1. An XML repository for XREAL

Most of the modern DBSs extend their relational DBS features to support XML storage and retrieval, such as recent versions of IBM DB2 or Oracle. These DBSs provide an XML data type. A well-formed XML document could be stored in an attribute of XML type. This XML document could be validated against an XML Schema, which should be registered in the DBS. These modern DBSs provide support for retrieving XML data using both SQL and XQuery languages.

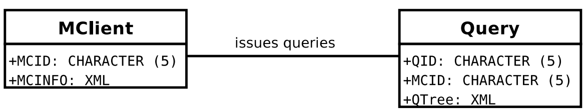

Figure 9. A DB schema for the XREAL repository

We developed a DB schema for providing an XML repository for the XREAL specification of the mobile client contextual information and queries issued by this client. Figure 9 depicts the DB schema developed under DB2 express 9.5 to store the XREAL specification. The schema consists of two tables, *mclient* and *query*. The *mclient* table is to store the XREAL specification of contextual information of a specific mobile client. The key attribute of the *mclient* table is *MCID*. The specification is to be stored for each mobile client in the *MCINFO* attribute, which is of XML type.

The *query* table consists of the attributes *QID, MCID* and *QTree*. The *QID* is the key attribute of the *query* table. The *MCID* attribute is used to link the query to the mobile client issuing it. The *QTree* attribute is of XML type and used to store the XREAL specification of the relational algebra query tree. The XML Schema of the XREAL model is registered into DB2 under these schemas names: *mclientXSD* and *queryXSD*. The XREAL specification is to be validated using these registered XML Schemas. Figure 10 shows the insert statements used to add and validate contextual information of a mobile client and a query issued by this mobile client. The inserted XML documents are validated using a function, called *XMLVALIDATE*, whose parameters are an XML document and a specific registered XML Schema.

```
INSERT into mclient values (   'MC101',
  XMLVALIDATE(? ACCORDING TO XMLSCHEMA id mclientXSD))
INSERT into query values (   'QID1','MC101',
  XMLVALIDATE(? ACCORDING TO XMLSCHEMA id queryXSD))
```

Figure 10. Insertion and validation statements

7.2. Context-Aware Query Processing

The XREAL specifications of the contextual information is the base for processing any context-aware query. The main idea is to represent the context-aware semantics using relational algebra operations. The specifications of the contextual information and a query is used to generate an instance of this query according to the current context(s) of the user, who issued this query. This instance is generated by replacing relatives attributes with its corresponding values from the context of the user. The query 7 shown in Figure 2 is an example for such process. Figures 7 and 8 shows part of the specification of the instance query. Finally, a SQL query is generated from such instance and executed using the DBS, which at the same time manages the relational data of the application.

We have implemented the context-aware query processor as built-in DBS function supported with Java-stored procedures. Now, we are supporting context-aware queries based on location specified using relative position, such as postal_code. More advanced context-aware functions, such as *close to*, *towards*, and *approaching*, are to be supported.

8. Discussion: The merits of XREAL

The XREAL specification is an XML representation of a mobile client contextual information and the queries issued by this mobile client. The XREAL model for the query has been created by mapping the productions of the relational algebra syntax directly into XML productions. The XREAL specification is not particularly convenient for humans to read and write, but it is easy for programs to parse. Because XREAL is an XML-based model, standard XML tools could be used to create, interpret, or modify queries or contextual information belong to a specific mobile client. Query management in mobile information system is application-based. Consequentially, humans are not to read or write the XREAL specification. XML tools, such as XSLT, could be used to map the XREAL specification into human readable format.

The XREAL inherited the merits of XML, such as exchange flexibility, tree nature, and DBS support. Exchange flexibility, the XREAL specification is to be shared among heterogeneous applications and systems. The tree nature of XREAL provides high compatibility in representing relational algebra query trees.

Moreover, the XREAL specification is to be stored and retrieved using DBS, which is utilized to manage the data at the server side. That means on the one hand the management of mobile queries and relevant functions, such as context-aware query processing, is to be integrated into and supported by DBS. On the other hand, the mobile query management is moved from the application layer to the database layer. Therefore, several intermediate layers are to be avoided, which leads to performance improvement as shown in [3].

9. Conclusion and Outlook

This paper has presented an approach where mobile clients could issue or pre-register queries, which depend on contextual information, on a server. Dynamically the current contextual information of mobile clients are to be considered for executing the queries issued by these clients. The main contribution of the paper is the XML-based XREAL model that formalizes the contextual information as well as the queries, which are represented as recursive relational algebra tree. A major advantage of XREAL is the direct integration into modern DBSs supporting XML. Moreover, utilizing XREAL provides context-aware management support within these DBSs, rather than requiring the development of complex and slow middleware.

We are working to enhance the XREAL model and our context-aware query processor. The main issues of our work-in-progress are: 1) the support of all relational algebra operation in the XREAL model, 2) the formalization of the correlation between different contexts, 3) the formalization of advanced context-aware predicates, such as *close to*, *towards*, and *approaching*, in the form relational algebra operations, and 4) the support of the advanced context-aware predicates in our context-aware query processor by mapping such predicates into SQL statements.

References

[1] H. Höpfner, S. Schosser, and K.-U. Sattler, "An Indexing Scheme for Update Notification in Large Mobile Information Systems," in *Current Trends in Database Technology - EDBT 2004 Workshops, Revised Papers*, ser. LNCS, vol. 3268. Berlin, Germany: Springer-Verlag, Nov. 2004, pp. 345–354.

[2] C. Bunse and H. Höpfner, "Resource substitution with components — optimizing energy consumption," in *Proceedings of the 3rd International Conference on Software and Data Technologie*, J. Cordeiro, B. Shishkov, A. K. Ranchordas, and M. Helfert, Eds., vol. SE/GSDCA/MUSE. Setúbal, Portugal: INSTICC press, Jul. 2008, pp. 28–35.

[3] E. Mansour and H. Höpfner, "An Approach for Detecting Relevant Updates to Cached Data Using XML and Active Databases," in *12th International Conference on Extending Database Technology (EDBT09). Accepted*, 2009.

[4] H. Höpfner, "Using database techniques for realizing context aware mobile information systems," in *Advanced Database – Proc. of the 2nd International Advance Database Conference (IADC 2006); June 27-29, 2006, San Diego, USA*, M. Amin, P. P. Dey, and A. Hadaegh, Eds. San Diego, CA, USA: US Education Service, LLC., 2006, pp. 34–38.

[5] M. Weiser, "The computer for the 21st century," Scientific American, Tech. Rep., 1991. [Online]. Available: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html

[6] W. N. Schilit, "A System for Context-Aware Mobile Computing," Ph.D. dissertation, Columbia University, New York, USA, 1995.

[7] K. Mitchell, "Supporting the Development of Mobile Context-Aware Computing," Ph.D. dissertation, Departement of Computing, Lancaster University, Jan. 2002.

[8] A. Schmidt, "Ubiquitous Computing – Computing in Context," Ph.D. dissertation, Computing Department, Lancaster University, Nov. 2002.

[9] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications," Ph.D. dissertation, College of Computing, Georgia Institute of Technology, Dec. 2000.

[10] Q. Ren and M. H. Dunham, "Using semantic caching to manage location dependent data in mobile computing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM Press, 2000, pp. 210–221.

[11] H. Höpfner and K.-U. Sattler, "Semantic Replication in Mobile Federated Information Systems," in *Proc. of the Fifth International Workshop on Engineering Federated Information Systems (EFIS2003), Coventry, UK 17th - 18th July, 2003*, A. James, S. Conrad, and W. Hasselbring, Eds. Amsterdam, Netherlands: Ios Press Inc, Aug. 2003, pp. 36–41.

[12] M. Korkea-Aho, "Context-aware applications survey," Department of Computer Science, Helsinki University of Technology, Tech. Rep., 2000.

[13] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*. Addison Wesley, 2007.

[14] H. Höpfner and E. Buchmann, "Towards compact operator trees for query indexing," in *Proceedings of the 3rd International Conference on Software and Data Technologie*, J. Cordeiro, B. Shishkov, A. K. Ranchordas, and M. Helfert, Eds., vol. ISDM/ABF. Setúbal, Portugal: INSTICC press, Jul. 2008, pp. 174–179.