Approximate Query Answering System Architecture

Francesco DI TRIA, Ezio LEFONS, and Filippo TANGORRA
Dipartimento di Informatica
Università degli Studi di Bari Aldo Moro
Via Orabona 4, 70125 Bari
ITALY
{francescoditria, lefons, tangorra}@di.uniba.it

Abstract: Business Intelligence is an activity that aims to extract information and knowledge from a central repository, the so-called data warehouse, in order to improve the business processes of an information system. Typical applications are based on reporting, on-line analytical processing, data mining, and approximate query processing. Business Intelligence platforms are software tools that allow to develop such applications. In general, these applications are composed of complex dashboards that provide a synthetic frame to be used in decision making. According to the criteria proposed by the Gartner Group to evaluate Business Intelligence platforms, one of the most important feature is the information delivery strategy, whereby final users can share the environment and access the same resources in real-time. For this reason, more and more traditional vendors include web services along their software packages. At the present time, Web-based platforms focus mainly on the issues related to the applications' deployment on a server. However, platforms that support approximate query processing require a more complex architecture since they generally need to perform a preliminary data reduction process and, then, they require ad hoc metadata. In this paper, the architecture of such a system is presented along with a proposal of standardization of metadata to be used in approximate query processing.

Keywords: Decision support system, Web service, Approximate Query Processing, Data warehouse, ADAP system, OL2AP metamodel.

1 Introduction

Business Intelligence applications consist of multi-dimensional analyses (such as the popular OLAP) of the data stored in a data warehouse. The results are usually represented with dashboards [1], which are graphical frameworks that summarize the main key performance indicators. A key performance indicator is a metric defined to measure the results with reference to business goals identified by decisional makers. As an example, a dashboard is able to effectively describe the current set-up of the enterprise and to produce estimates of future trends. The Business Intelligence Platform Capability Matrix [2] has been defined by the Gartner Group in order to establish standard criteria to evaluate systems used by business companies to develop applications supporting decision makers. A set of criteria is related to the information delivery issue. According to these criteria, a platform must provide the ability to publish dashboards to a web-based interface and each user must be allowed to build personal indicators.

Business Intelligence applications can be developed also using approximate query answering systems [3], that are software tools able to provide fast responses during analytical processing. These systems are based on the assumption that it is more suitable to obtain a fast response affected with a tolerable quantity of error rather than the exact response, if the query is very time-consuming. In fact, in decision making approximate answers often suffice for strategic choices. As a counterpart, these systems need to perform a data reduction [4], that is a process that computes a synopsis of (part of) the data stored in the data warehouse (DW). Therefore, the analysis environment of an approximate query answering system needs to access a set of ad-hoc metadata describing which data contained in the data warehouse have been reduced and, thus, effectively available for approximate query processing.

The aim of the paper is to illustrate the architecture of an approximate query answering system that (a) includes a web service, and (b) supports the analytical processing on the basis of an XML representation of the needed metadata. This representation can be thought as a proposal of standardization in the scope of approximate query processing.

The paper is organized as follows. The architecture of the approximate query answering system is described in Section 2, along with its detailed features. Some issues about the representation of the metadata used by these systems are reported in Sec-
tion 3. Finally, Section 4 concludes the paper reporting our remarks.

2 Approximate query answering system

Nowadays, approximate query answering systems are always used in decision support systems, since they allow business users to obtain fast responses to queries whenever they do not need total precision or exact values. In Figure 1, there is depicted the high-level architecture of an approximate query answering system which processes datasets stored in the data source (i.e., the data warehouse DW) in order to compute synopses of the data (DS) by a reduction process.

Once the data reduction has been performed, it is important to trace which fields and relations have been actually reduced, as some data stored in the database can be ignored for decision making based on approximate answers. Therefore, tracing which data have been reduced provides decision makers with the knowledge about which data are effectively available for approximate query processing. In fact, the data synopses are then used in On-Line Approximate Analytical Processing (hereinafter called OL2AP), a process aiming to perform traditional OLAP on the basis of approximate query processing. Such a process is able to provide fast answers to complex (and usually aggregate) queries that would require a high computational time.

Currently, the standard representation of the metadata in data warehousing is the Common Warehouse Metamodel (CWM) [5]. These metadata are used by the Business Intelligence platforms to represent the multidimensional model of the data warehouse and to generate, in automatic way, SQL instructions without writing lines of code [6]. The CWM is based on XMI [7], as XML-based layer that provides interoperability among Business Intelligence tools.

However, the CWM does not include a metamodel that covers approximate query processing. So, there is no standard representation of metadata useful for approximate query answering systems. As already stated, these systems need to compute a synopsis of the data stored in a data warehouse. Then, the synopsis is used for the computation of the analytical queries. Therefore, this kind of systems need to know which data have been effectively reduced and, consequently, usable in the approximate query processing. Of course, this information must be provided by metadata. For this reason, a central research topic related to approximate query processing is strictly related to the representation of metadata, which is not yet been standardized.

In our research, the methodology used to address this issue is based on an XML-based representation of metadata and a web service able to export such metadata into a web-based analysis environment. In the next sub-section the architecture of a web-based approximate query answering system is presented.

2.1 System architecture

According to the general architecture shown in Figure 1, in this sub-section the architecture of our system, called Analytical Data Profiling (ADAP) is presented. In detail, ADAP is an OLAP tool, whose features are to collect, to organize, and to process large volumes of data stored in a data warehouse, in order to obtain statistical data profiles to be used in approximate query processing.

The main macro-activities supported by ADAP are:

1. **Data reduction.** In this first step, the system calculates the synopses of data. All the calculated data are stored in the system database, that represents the main repository accessed in the next analytical processing. ADAP performs read-only accesses to the data warehouse, whenever it is necessary to (re)calculate the data synopses DS.

2. **Approximate query processing.** In this step, the system performs the computations of aggregate functions in approximate way, by using only the data synopses. The output of the processing are scalar values that represent the approximation of aggregated values. However, the system provides also a method to execute queries directly on the data warehouse, when the user deals with
critical factors or when the maximum precision is needed. In ADAP system, the data reduction process uses the consolidated methodology of orthonormal series based on Legendre polynomials [8]. Using this method, data synopses are represented by sets of coefficients, the so-called Canonical Coefficients. On the other hand, using histogram-based methodologies for example, the data synopses are represented by buckets, each of them containing the frequency of values falling within predetermined intervals.

Regardless of the specific methodology, the data synopses carry synthetic information to approximate the data distribution of a multidimensional relation. Therefore, the main aggregate functions (such as count, sum, and average) can be computed without accessing the millions-of-records relations of the data warehouse. However, the response may be affected with a small quantity of error.

ADAP has been designed according to a four-levels architecture (see, Figure 2) and developed according to a modular design, in order to allow the add-in of features not yet implemented. The system manages both the data (i.e., the computed coefficients) and metadata (i.e., the information about which data in the data warehouse have been reduced).

In the Presentation layer, Administration is the input component that allows users to select a data warehouse. It receives the metadata of the selected data warehouse (in particular, names of fields and tables) from the DW Manager and presents them to the user. Using the Administration component, users define the attributes to be involved in the data reduction. When the user ends the selection process, this component asks the DS manager to start the computation of the data synopsis on the basis of the selected attributes. Analysis Environment is the web-based application that allows users to define analytical queries. First, it loads ADAP metadata via the web service. These metadata define which data are available for approximate query processing. The approximate analysis is executed by accessing only the Data Synopsis repository. This kind of analysis returns very fast query answers and the results are visualized in the Report Browser, which is the dashboard container. In detail, it gets from the Analysis Environment component the user query and starts the analytical processing by calling a public method provided by the web service. At the end of the computation, it reports the results, that can be approximate or real values, and also shows the response time (in milliseconds).

In the Application layer, DW Manager is the only component deputed to access the data warehouse using the DB Bridge component. It extracts data and metadata from the data warehouse and distributes them to others components. For the Administration component, it extracts the metadata of the selected data warehouse. For the DS Manager, it performs a read-access to the data warehouse and passes it the dataset containing the data to be used for data reduction. For the Query Engine component, it supports real analyses by executing SQL instructions on the data warehouse.

DS Manager is the basic component with a twofold role: to generate the data synopses (that, in our case, sets of Legendre polynomials’ coefficients) and to extract them during the approximate query processing. When storing the coefficients in the Data Synopsis database, it also stores further metadata (i.e., which data of the data warehouse have been selected by the administrator for data reduction). Query Engine is the basic component that executes OLAP queries. In approximate query processing, its own sub-component, the so-called Approximate Query Engine, uses the data synopses and returns the approximate answers. Otherwise, the Query Engine translates the query into an SQL instruction to be executed by the DW manager.

Finally, the DW Manager and the DS Manager interact with the DB Bridge that is the component that manages the communication with both the DW and DS databases via an ODBC connection.

3 Metadata representation

As widely explained in [9, 10], the importance of metadata derives from the fact that they represent...
the only way to provide further knowledge about a business process or a component of an information system. In our context, the metadata are used to describe a data reduction process in order to support approximate query answering systems in multi-dimensional analyses.
So, the ADAP system is able to create the needed metadata according to the novel OL2AP metamodel, that constitutes our proposal for the extension of CWM.
We included the OL2AP metamodel in the Analysis layer of [5] that groups all the metamodels that must be implemented by Business Intelligence tools performing analytical processing.
The aim of this metamodel consists of tracing:

- which fact tables, among all the tables of a given data warehouse, have been chosen for data reduction;
- which dimension tables have been involved in the data reduction;
- which attributes of the selected tables have been reduced; and
- eventual parameters of the data reduction.

The dimensions and measures of cubes that have been reduced are effectively available for approximate query processing. Therefore, a decision maker is able to express an analytical query involving the reduced attributes in order to obtain a fast response.
In Figure 3, there are depicted the main classes and associations that allow to create standard metadata that can be effectively used by approximate query answering system in order to trace the data reduction process and to generate analytical queries based on approximate answers. Notice that this metamodel depends on: (a) the Relational metamodel, that provides an approximate query answering system with the metadata describing the data warehouse logical mode (usually, ROLAP), and (b) the Core metamodel, that allows to attach a descriptor to any model element (i.e., the element of the database being modelled). A detailed description of these metamodels can be found in [5].
The main classes of the OL2AP metamodel are summarized in Table 1. Given a class, an its instance is called metaobject and it represents a construct of the system to be modelled.
As an example, if we want to create the metadata of a relational database, then we have to use the Relational metamodel, which establishes to create a metaobject of the Table class for each table of the database.
The steps for the creation of the metaobjects of the OL2AP metamodel are the following:

- a metaobject of the class Redcution is created to represent the physical database, and the name of this object is the Data Source Name (DSN) used for the physical connection to the database;
- for each relational schema chosen for the data reduction, a metaobject of the class RSchema is created, having the name of the schema (i.e., the name of the relational database) and a reference to the physical database it belongs to;
- for each table of a relational schema chosen for the data reduction, a metaobject of the class RTable is created, having the name of the table and a reference to the schema it belongs to; and
- if a table or a column must be tagged, then a metaobject of the class Descriptor is created, having an arbitrary name and a reference to the model element it belongs to.

In the modelling process, in order to obtain identifiers of the created metaobjects and to ensure correct referencing among these objects, techniques derived from object-oriented database management must be used (cf. [11] for instance).

3.1 XML layer
The OMG has defined XMI format as the physical layer used to store the metadata that are obtained through a serialization process of the created metaobjects. Since it is XML-based, XMI allows high interoperability among Business Intelligence tools.
Here, we show how the metadata can be represented using an XML file, that can be suitably transferred via a web service.
In fact, the OL2AP metamodel can represented by the following Document Type Definition (DTD):

```
<!ELEMENT reduction (schema)>
<!ELEMENT schema (table+)>
<!ELEMENT table (column+)>
<!ELEMENT table (descriptor*)>
<!ELEMENT column (descriptor*)>
<!ELEMENT descriptor (name, value)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT value (#PCDATA)>
<!ATTLIST reduction name CDATA #REQUIRED>
<!ATTLIST schema name CDATA #REQUIRED>
<!ATTLIST table name CDATA #REQUIRED>
<!ATTLIST column name CDATA #REQUIRED>
```
Fig. 3.OL2AP metamodel.

<table>
<thead>
<tr>
<th>Class</th>
<th>Metamodel</th>
<th>CWM Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelElement</td>
<td>Core</td>
<td>A model element is an element that is an abstraction drawn from the system being modelled.</td>
</tr>
<tr>
<td>TaggedValue</td>
<td>Core</td>
<td>A tagged value allows information to be attached to any model element in the form of a “tagged value” pair; that is, name = value.</td>
</tr>
<tr>
<td>Catalog</td>
<td>Relational</td>
<td>A catalog is the unit of logon and identification. It also identifies the scope of SQL statements: the tables contained in a catalog can be used in a single SQL statement.</td>
</tr>
<tr>
<td>Schema</td>
<td>Relational</td>
<td>A schema is a named collection of tables and collects all the tables of the same relational schema (i.e., a logical database).</td>
</tr>
<tr>
<td>Table</td>
<td>Relational</td>
<td>Table is a data structure representing a relation.</td>
</tr>
<tr>
<td>Column</td>
<td>Relational</td>
<td>Column is a data structure representing the field of a relation.</td>
</tr>
<tr>
<td>Reduction</td>
<td>OL2AP</td>
<td>This class represents a process of reduction of the data stored in a relational database.</td>
</tr>
<tr>
<td>RSchema</td>
<td>OL2AP</td>
<td>This class represents a relational schema chosen for the data reduction.</td>
</tr>
<tr>
<td>RTable</td>
<td>OL2AP</td>
<td>This class represents a table chosen for the data reduction.</td>
</tr>
<tr>
<td>RColumn</td>
<td>OL2AP</td>
<td>This class represents a column whose data have been reduced.</td>
</tr>
<tr>
<td>Descriptor</td>
<td>OL2AP</td>
<td>A descriptor is a tag that can be attached to any element of the model.</td>
</tr>
</tbody>
</table>

Tab. 1. OL2AP metamodel classes.
This DTD expresses the concept that a data reduction process involves a relational schema, which is composed of a set of tables. On the turn, a table is composed of one or more columns. Each table and each column can be tagged by several descriptors, which are pairs (name, value) useful to associate context-dependent data. As an example, we can associate the descriptor (rows, 1000) to a given table in order to state that the cardinality of the reduced table is 1000 rows. As a further example, we can associate the descriptors (min, 1) and (max, 100) to a given column in order to trace the minimum and the maximum of its domain, respectively. An example of XML file containing OL2AP metadata follows.

```xml
<reduction name='salesDSN'>
  <schema name='sales'>
    <table name='orders'>
      <column name='employId' />
        <descriptor>
          <name>min</name>
          <value>1</value>
        </descriptor>
        <descriptor>
          <name>max</name>
          <value>100</value>
        </descriptor>
      </column>
      <column name='orderId' />
        <descriptor>
          <name>min</name>
          <value>1</value>
        </descriptor>
        <descriptor>
          <name>max</name>
          <value>100000</value>
        </descriptor>
      </column>
      <column name='quantity' />
        <descriptor>
          <name>min</name>
          <value>1</value>
        </descriptor>
        <descriptor>
          <name>max</name>
          <value>100</value>
        </descriptor>
      </column>
    </table>
  </schema>
</reduction>
```

4 Conclusions

In this paper, we present the architecture of ADAP system, a Business Intelligence tool based on approximate query processing. This tool can be used in a web environment since it includes a middleware able to export internal metadata in XML format. Therefore, the main contribution of the paper is an effort to define a standard model to represent metadata used by approximate query answering systems. Standard metadata are effectively able to trace which data have been reduced and, so, which data are available for approximate query processing. Since the physical layer used to represent such metadata is an XML-file, the standard metadata can be used in web based applications in order to define centralized and shared analysis environments for decision makers aiming at obtaining fast responses.

References


