MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Module for Easy and Interoperable Data Analysis in Electronic Medical Records

Bruno Cadonna[a], Peter Beck[a], Ivo Rakovac[a], Thomas Truskaller[a], Thomas R. Pieber[a, b]

[a] *Institute of Medical Technologies and Health Management, JOANNEUM RESEARCH Forschungsgesellschaft mbH, Austria*
[b] *Department of Internal Medicine, Diabetes and Metabolism, Medical University Graz, Austria*

Abstract

Data analysis in electronic medical records (EMRs) is crucial for patient treatment, quality management, administration and research. The widely used SQL queries quickly become complex in the medical domain resulting in difficult and time-consuming development and maintenance. The aim of this work is to develop a data analysis module using Java, XML and SQL, which eases development of analyses by utilising recurring patterns in SQL queries, minimises dependence of analysis on databases, makes analyses modifiable at runtime and visualises results as customisable charts and tables. The data analysis module developed is a first step toward easy-developing and interoperable data analysis in EMRs.

Keywords:

data analysis, medical records system, databases

Introduction

Data analysis in electronic medical records (EMR) and visualisation of its results is crucial for patient treatment, quality management, administration and research. Widely used, often complex, plain SQL queries are tightly bound to database schemas (DBSs) and database management systems (DBMSs) resulting in difficult and time-consuming development and maintenance. Current reporting tools avoid direct use of SQL, but they do not decrease complexity of queries nor do they reduce dependencies of queries on DBSs.

SQL queries in the medical domain often have recurring requirements with similar or identical patterns – e.g. patient classification regarding their habits (e.g. non-smoker, moderate smoker, chain-smoker) and average calculation of laboratory values (e.g. blood pressure) – which could be reused among queries. Greater flexibility can be obtained through parameterisation of SQL queries – e.g. selecting only female, only male or all patients (see Figure 1).

The aim of this work is to develop a data analysis module for integration in information systems, which

a) eases analysis development by providing abstractions for recurring patterns in SQL queries,
b) minimises dependence on DBSs and DBMSs,
c) makes analyses modifiable at runtime and
d) executes analyses and visualises results as customisable charts and tables.

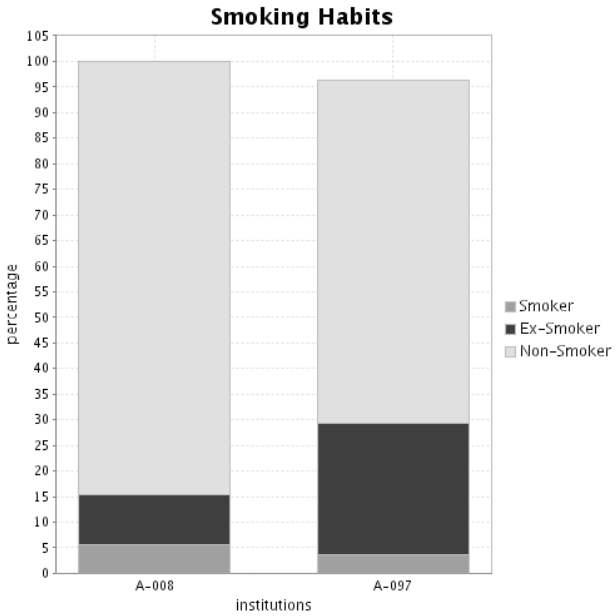

Figure 1 - Comparison of two institutions regarding the smoking habits of their female patients (100% are not reached for A-097 due to missing data). For this analysis the classification pattern was used classifying patients regarding their smoking habits. Additionally the analysis was parameterised in order to select only female patients

Methods

The module was implemented in Java using open source libraries. XML Schema was used to define valid mark-up for analysis and configuration instances. An analysis XML instance contains mark-up for

a) generalised SQL query patterns,
b) a parameter-based definition of the data to analyse referencing reusable sets of data in the configuration XML and
c) default visualisation properties.

A configuration XML instance has to be defined once per database and contains metadata about

a) DBMS,
b) DBS
c) reusable sets of data to analyse and
d) available parameters for parameterisation of the data to analyse.

It is used for runtime generation of database-specific SQL queries. A second configuration XML instance specifies which chart types are permitted for visualisation of analysis results depending on attributes of analyses. This configuration XML instance avoids meaningless combinations of chart types and analyses such as longitudinal trends of patients' blood pressures visualised as a pie chart.

Results

The data analysis module developed generates SQL queries from database-independent analysis XML instances and database-specific configuration XML instance, executes the SQL queries and visualises results with default visualisation properties (see Figure 2). Visualisation properties such as chart type, colour schemas etc. can be further modified. A web interface for query selection, execution and visualisation of results has been developed. Internationalisation was achieved by means of Java resource bundles.

Discussion and conclusion

The work presented is a first step toward faster and easier development and maintenance of interoperable analysis of data in EMRs. The analysis structure developed allows easier development, maintenance and reuse of analyses on a higher level of abstraction. Modularity of analysis structure simplifies parameter-based modifications at runtime. Visualisation of results can be easily customised. Analysis XML instances unify query and result visualisation in one place.

The module developed is currently integrated in quality management web applications for hepatitis and cardiovascular disease hosted at www.healthgate.at and in a module for generation of PDF reports.

Future work will focus on improving visualisation capabilities (sorting and filtering of obtained results), user interface integration using metadata (e.g. dynamically generated input elements for parameter values), adding dynamic analysis descriptions (i.e. descriptions which adapt to results) to analyses XML instances and further separation of database-independent components from database-specific ones.

Moreover to foster further development of the data analysis module practicability of an open source release will be explored.

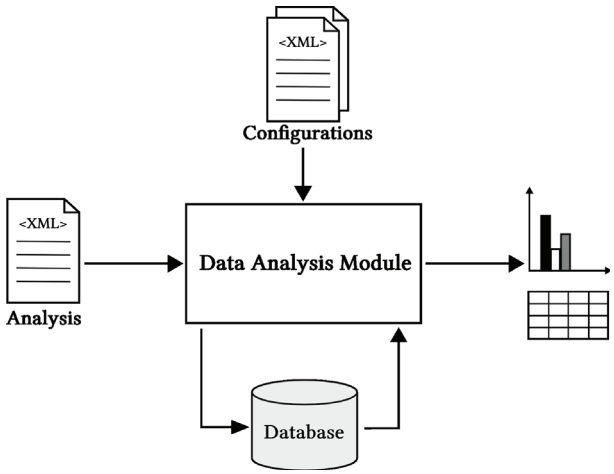

Figure 2 - Black box model of data analysis module