Tools for *“Pi of the Sky”* data exploration

Małgorzata Sawitus[a], Jerzy Tyszkiewicz[a], Lech Mankiewicz[b], Grzegorz Wrochna[c]

[a] Institute of Informatics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
[b] Center for Theoretical Physics, Polish Academy of Sciences, Lotników 32-46,
02-668 Warsaw, Poland
[c] A. Soltan Institute for Nuclear Studies, Hoża 69, 00-681 Warsaw, Poland

ABSTRACT

The *Pi of the Sky* experiment searches for fast optical transients in the sky. Its data is stored in huge databases, which contain mainly astrophysical objects' data and measurement of their brightness over time. This paper describes an application that was designed to make analysis of this data more effective. User requirements and solution based on them will be presented. The main idea of this solution is based on creating forms describing database queries in XML files. Such approach guarantees that changing existing forms and creating new ones is a simple task.

Keywords: astronomical data, SQL, query, XML, database.

1. INTRODUCTION

The *Pi of the Sky* is a project that monitors the sky to search for flashes having extragalactic origin and explores astronomical objects with fast variability. Special Charge Coupling Device (CCD) cameras take pictures of the sky, which afterwards are being automatically analyzed. Flashes are recognized from a huge amount of data (about 25 GB per night) using photometry and astrometry methods. The results are stored in database. Once the data is stored, other people can analyze it further. For example, variable stars can be recognized by the light curve analysis. Presented application was made specially for that purpose. It's important for everyone interested in astronomical data to have easy and comfortable access to data. The more people can easily analyze this data, the more interesting findings can be made.

Next section contains requirements for this application. Section 3 contains list of features. Section 4 describes the architecture. Section 5 is about technical part of application. Section 6 presents user interface.

2. REQUIREMENTS

A typical user of the presented application is an astronomer or physicist rather than a computer science or information technology professional. In particular, many of the users have very limited, if any, knowledge about SQL and its features. At the same time, the way the databases of *Pi of the Sky* are used resembles an OLAP application, where the users create and evaluate queries *ad hoc*. This excludes the possibility of using only a small set of predefined queries, hard-encoded into the database’s interface.

Therefore the need emerged to provide the users with an intuitive, comfortable and easy to use tool to create, modify and share queries.

Using the application we describe, advanced users should be able to write arbitrarily complex SQL queries, while those who do not know much about SQL should be offered with templates, in which they only choose or alter predefined parameters. Moreover, there should be a method to share the queries among the users. Apart from that, the application should provide a way to specify the method of result visualization, and allow the users to save the results of queries in external formats.

Since application allows for access to the data for external users, it must be secure.

3. FEATURES

The main features of the application are as follows:

- Forms and queries
  - The basic form of access to queries are user-friendly forms. They include the code of queries. Forms are stored as XML files and displayed by the application, which is independent of the database.
  - For each of the users, the available forms are organized in a tree-like structure, so it's easier to navigate through them.
  - If a user wants to save the parameters of the query for some reason (for example the query gives an interesting result set), he can name the parameters' set and store it to get back to it later or to share it with others.
  - If an appropriate form does not exist or the user just wants to type in directly an SQL query, it is also possible via a built-in generic query executor.
- Connections with databases and execution of queries
  - Application features asynchronous access to data. Users can execute as many queries at the same time as they want to. This feature is very useful when, for instance, some very time-consuming query is executed on one database, while user wants to execute a small query on another database. Each database connection is configured through single XML description.
  - After preparing the query and starting its execution, this task is added to task list. User can trace its state and is notified when it is ready.
  - If query can give very big result and the user does not want to wait for its completion too long or is satisfied with a partial result, he/she can set a limit execution time and/or maximal size of data set.
- Result analysis
  - The result of a query can be visualized directly by the application. So far only an HTML viewer is provided, but other formats are planned to be supported in the future.
  - The result of the query can be written to a file, for subsequent visualization or analysis by an external tool. Many data formats can be chosen for that purpose.
- Plugin system.
  - The application has a built-in plugin system which allows for extending its functionality. Java plugins can be used to add: new connection types, new types of input fields in forms (at present the application supports: integer, text, real, date and boolean types), new query result visualization methods, and new formats for saving query results.

4. ARCHITECTURE

The program was written in Java 1.4. It works under both Windows and Linux operating systems. Formats of all XML documents are described by XML schema. In the internal structure of the application, a few modules can be separated:

- Interface

  The main goal of the user interface design was usage convenience. It will be presented in more details in section 5 below.

- Logic

  Logic performs all key tasks of the application. It also operates on query results. Logic is fully independent from the user interface.

- XML files defining the database connection language

  These XML files are interpreted by the Logic. They define a so-called "connection language" which describes both connection type and operations that can be executed on each database. All elements of the connection language are represented by XML documents and provide a method to configure the application.

- Communication module

  Since direct access to a database is often too risky, the application offers a server, which acts as a secure proxy for executing queries (see below). The user application works as an HTTP client.

  Currently there are two working implementations of the connection type plugin: direct connection with database and HTTP proxy-like connection. The first one is a typical connection used when working in a local *Pi of the Sky* network. The second one, obviously less efficient, is used for remote access through Internet and accesses the HTTP server.

- HTTP server

  This module implements a HTTP proxy server whose job is to execute the requested queries on the database and send the results back to the users. This part of he application is installed on the server side, while the users get client software. It is based on Tomcat server and Java servlets. Thanks to this additional server connection, the traffic can be filtered and so is more secure than typical connection to database.

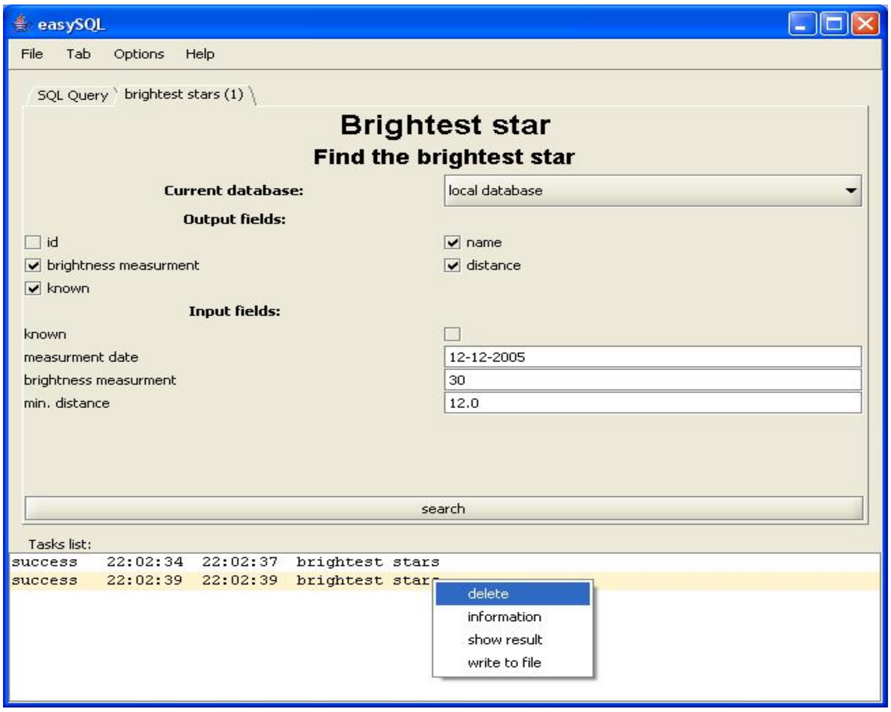
5. INTERFACE

Main elements of user interface are tabs panel, task window and menu.

All relevant options related to forms and connections are accessible from application main menu.

Task window shows the state of all queries (possible states are: success, failure and executing). Right mouse click on completed query opens up an additional menu containing options related to it (i.e. show result, write to file, delete or show execution information).

Tabs panel, placed in the central part of the application window, contains 3 kind of tabs: generic SQL query tabs, form tabs and result tabs. Each generic query tab has a text window, where the user can type in any SQL query. Each form tab contains a form, in which output and input parameter can be specified, and also text information (i.e. description of the fields and form title). All query and form tabs contain a selectable list at the top, which allows the user to switch between available databases to run the query on. Each result tab shows the results in a selected format (built-in format is HTML table).



Brightest star

Find the brightest star

Current database: local database

Output fields:

☐ id

☑ name

☑ brightness measurment

☑ distance

☑ known

Input fields:

known	☐
measurment date	12-12-2005
brightness measurment	30
min. distance	12.0

Tasks list:

```
success    22:02:34  22:02:37  brightest stars
success    22:02:39  22:02:39  brightest stars
```

Figure1. User interface

6. SUMMARY

The presented application is designed to give an access to *Pi of the Sky* database for everyone interested in it, even from other projects and individual people. It is a universal database tool for both advanced users and those new in project. The described application has been partially deployed in *Pi of the Sky* project and we hope it will help the project members in their daily work.

7. REFERENCES

1. *Pi of the Sky* web page, http://grb.fuw.edu.pl/, 13.06.06
2. M. Biskup, *Data bases for 'Pi of the Sky' experiment*, these proceedings
3. Extensible Markup Language, http://www.w3.org/XML/, 13.06.06
4. XML Schema, http://www.w3.org/XML/Schema, 13.06.06