Lehrstuhl für Systeme und Informationsverwaltung

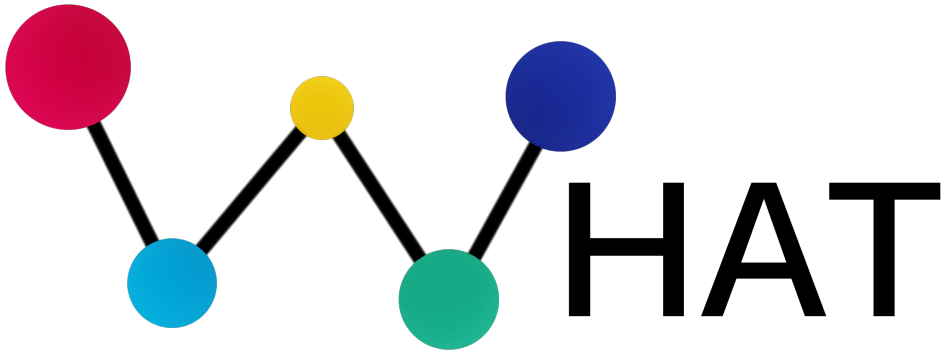
Prof.Dr.Ing. Klemens Böhm
Hoang Vu Nguyen
Marco Neumann

Karlsruher Institut für Technologie

Praxis der Softwareentwicklung (PSE)
WS 2012/2013

Visualizing and Statistically Analyzing Access Behavior to Scientific Databases

Design



WareHouse Analyzing Tool

December 10, 2012

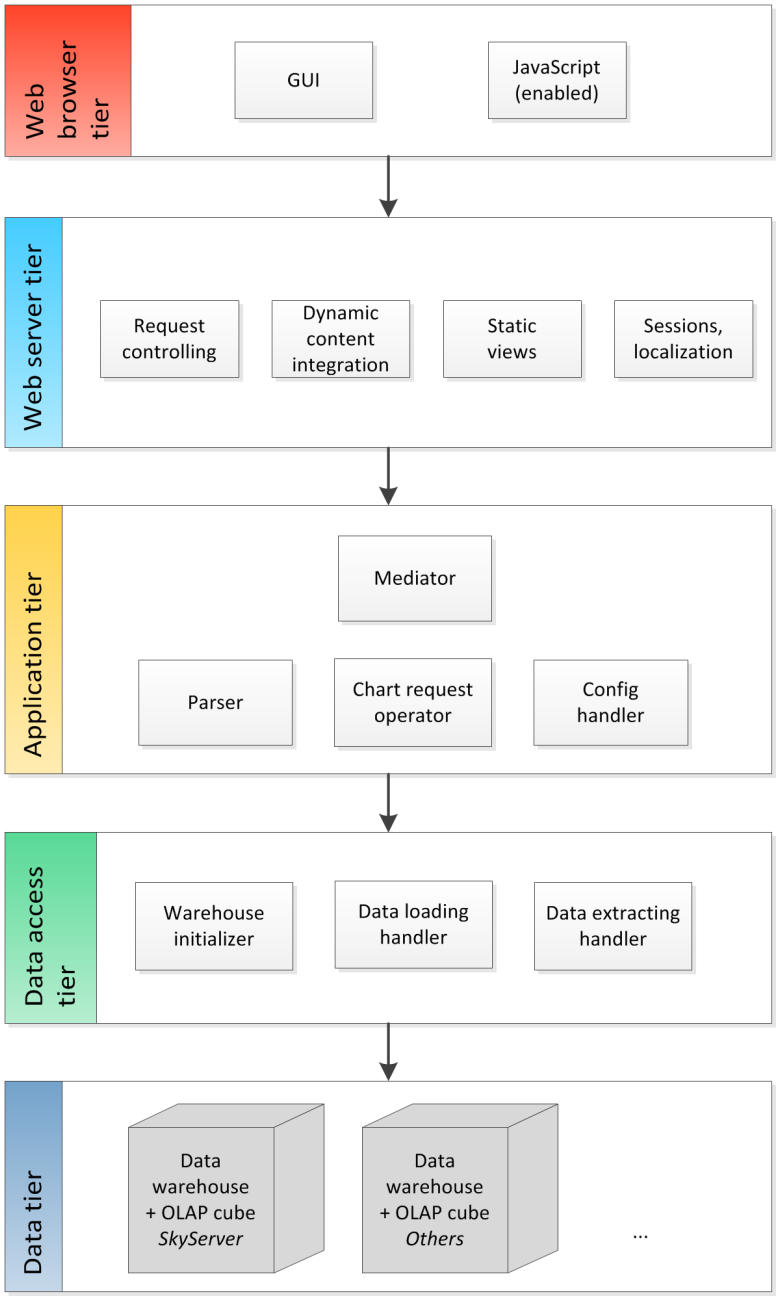
Functional Specification **Alexander Noe**
Design **Jonathan Klawitter**
Implementation **Anas Saber**
QA / Testing **Nikolaos Alexandros Kurt Moraitakis**
Final **Lukas Ehnle**

E-M@il: pse10-group14-ws12@ira.uni-karlsruhe.de

Contents

1 Architecture

WHAT is based upon a intransparend multitier architecture. The GUI, presentation logic, application processing, data accessing and stroring data are logically and also partly locally separated. Intransparent means that communication between tiers just happens between adjacent ones. The different tiers are described below.



1.1 Web browser tier

The web browser tier represents the web browser of the client on which the GUI, a web page, will be displayed. As the web page uses JavaScript, this has to be enabled in the browser. Besides this, Google Chrome and Firefox have to be supported.

1.2 Web server tier

The web server provides the presentation logic. This includes static html-pages and integration of dynamic conent. Other tasks are sessioning, localization (languages) and most important request controlling. Which means it handles the actions triggered on the web page and decides whether it can handle them itself or pass them to application tier. Last one happens for example when a chart is requested. The Java Play framework is used in this tier.

1.3 Application tier

In the application tier the parsing process, managing the configuration files and above all the chart computing are taking place.

1.4 Data access tier

This tier manages all requests of loading and extracting data from the data tier. So his main task is to build a bridge from the application in Java to the SQL language of the Oracle warehouses and OLAP-Cubes. If there is enough time to implement this optional function, it will also handle the automatic initialization of new data warehouses.

1.5 Data tier

In the data tier the data warehouses and their OLAP-Cubes are stored. This will be done with the Oracle software.

2 Web page design

3 Classes

3.1 Class diagram

3.2 Fassade + Configurations

3.2.1 Fassade

3.3 Chart request operator

Visitor pattern blabla classes

3.3.1 Fassade

3.4 Chart parameters

The chart parameters describe

3.5

3.6 Parser

ParserMediator

Parser is the 'main'-Class of the Parser. It creates and administrates a threadpool of ParsingThreads, contains the entryBuffer for finished DataEntries, the stringBuffer for strings, whcih were extracted from the logfile, a method to add them to the data warehouse and allows the user to select a new logfile and configuration file.

LogFile

The gateway between parser and logfile - it contains the path of the logfile and an integer, which saves how many lines have been read from this file. It can read single lines from the logfile and return them to the Parser.

DataEntry

The dataEntry which will be written in the warehouse. It contains hour, day, month, year of the request, rows which were read from the logfile, the elapsed and busy time on the server, the ip, from which the request came, the type of request, the database and server which handle the request and an object-array of additional information, which depends on the actual database entry.

Java Files

The Parser uses java.io.File and a ThreadPool from java.util.

ParseTools

The ParseTools are a set of static tools to parse the logline.

NormalFormTool

The NormalFormTool is used first. It converts the logline in a standardized form, which can be used by the other tools.

VerificationTool

The VerificationTool checks the logline and looks if it is a correct logline - if the logline got a mistake, it will be deleted and the VerificationTool sends an error.

ParsingThread

A parsing thread is one of the threads which are created from Parser. It gets a line from the logfile and uses the ParseTools to create a dataEntry, which represents the line from the log.

4 Data warehouse design

4.1 Overview

4.2 Dimension descriptions

4.3 Measure descriptions

4.4 Type description

As Type depends which data base is operated, it may get it’s own subsection.

5 Sequences

6 Other data

6.1 Static data

6.2 Dynamic data

6.3 Extern data

7 Libraries