Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Integration of standard datasources with interactive data visualization solutions

Bachelor's Thesis

Author
Márton Orova

Advisor
dr. Zoltán Szatmári
Attila Simon

November 20, 2019

Contents

HALLGATÓI NYILATKOZAT

Alulírott *Orova Márton*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2019. november 20.

Orova Márton
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon LaTeX alapú, a *TeXLive* TeX-implementációval és a PDF-LaTeX fordítóval működőképes.

Abstract

This document is a LaTeX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* TeX implementation, and it requires the PDF-LaTeX compiler.

Chapter 1

Introduction

A bevezető tartalmazza a diplomaterv-kiírás elemzését, történelmi előzményeit, a feladat indokoltságát (a motiváció leírását), az eddigi megoldásokat, és ennek tükrében a hallgató megoldásának összefoglalását. A bevezető szokás szerint a diplomaterv felépítésével záródik, azaz annak rövid leírásával, hogy melyik fejezet mivel foglalkozik.

Nowadays, the question of storing, processing and displaying the data is becoming more and more important throughout every industry. The time, when the collected data was only useful for computers and specialists, passed. Today, the need for showing the data in an easily understandable form is significant. It is no wonder, people through the whole hierarchy of a company would like to be well-informed about the results and the ongoing processes. In addition, it is getting highly valuable to be able to display vast amount of data in a way that even outsiders can comprehend.

Because of this trend, many technologies attempting to solve these problems have appeared, creating a wide variety of tools which organizations can use.

In enterprise-grade environments, the use of complex systems - so called data-pipelines - are becoming increasingly common. These tools provide an integrated solution for transforming and querying data coming from data sources built with different technologies. With the help of these data-pipelines, it is possible to collect many types of data, no matter the format or the frequency. All these things for one reason, to prepare the data for machine or human decision-making.

1.1 Problem definition

- many types of datasources, many ways of customizing them -> integration challenges
- standards made by the industry (data formats, accessing data, visualization) -> permeability is not easy

1.2 Motivation

- one visualization tool (Grafana) for multiple datasources in on place (one consistent way of visualizing data)
- open-source development

- integration task -> get familiar with many new technologies
- Grafana: de facto open-source visualization tool

1.3 Goals

- creating a data-gateway for accessing and visualizing data
- using different datasources (different technologies, data formats)
- connecting the gateway to Grafana (industry standard for opensource data visualization)
- presenting the main datasources and their features
  - relational databases
  - time-series databases
  - key-value stores
- discovering available options for interactivity in Grafana
- design a data-gateway for connecting (two-way, duplex) different types of datasources to grafana
- implement a POC data-gateway for connecting a Python based and a RapidMiner based datasource
- present the advantages and disadvantages of the created gateway

Chapter 2

Background

2.1 Datasource

- data formats
  - JSON
  - XML
  - other??
- data sources and their features, use-cases, pros/cons?
  - relational db
  - timeseries
  - key-value
- with examples!

2.2 Grafana

Grafana is one of the most popular open source analytics and monitoring solution that can be connected to the majority of the main data sources out-of-the-box. It allows its users to query, visualize and alert on the collected metrics.

Altough Grafana has got plenty of useful feature, only those relevant to the scope of the thesis project will be briefly explained here

2.2.1 Data Source

Grafana supports many different storage backends (Data Source). Each Data Source has a specific Query Editor (see later) that is customized for the features and capabilities that the particular Data Source exposes. Of course, this leads to the fact that the query language and capacity of each Data Source are obviously very different.

Grafana mainly favors time series data (e.g. from Prometheus or InfluxDB), but it can work with other types of data source (e.g. relational databases (MySQL, PostgreSQL, MSSQL), logging and document databases (Loki, Elasticsearch)).

2.2.2 Panel

The Panel is the basic visualization building block in Grafana. Each Panel provides a so called Query Editor (dependent on the Data Source selected in the panel) that allows the user to create data source specific queries (e.g. an SQL query for a MySQL data source) in order to extract the desired metrics as precisely as possible.

There are multiple built-in Panel types available in Grafana, however, custom panels made by the open source community are also accessible. Probably the most widely used Panel types are the Graph, Table, Singlestat and Gauge.

— TODO insert example picture —

2.2.3 Query Editor

The Query Editor exposes the capabilities of the Data Source and allows the user to query the metrics that it contains. The queries created in the Query Editor of a panel determine what data will be displayed on the panel.

— TODO insert example picture

2.2.4 Dashboard

The Dashboard is where all the previously mentioned building blocks come together. Dashboards can be thought of as an organized set of Panels.

We can use the Dashboard to visualize different metrics in one, easily manageable place. This is quite useful, when many aspects must be taken into account in order to be able to thoroughly understand our currently inspected data set.

— TODO insert example picture

2.2.5 Interactivity in general

When working with and visualizing massive amount of data is a major part of one's profession, it can be quite convenient if the tools can provide the ability to allow some user-interaction. Meaning that the user do not have to work with static diagrams or rewrite complex configurations so that he or she is capable of further analyzing the collected data.

Commonly available interactivity options can be the followings:

- *statistic indicators* - easily accessible statistic information (e.g. average value, spread-diagram) about individual objects shown on the diagram (this is usually achieved by mouse-hovering)
- *local interactions* - changing the outlook of a diagram without affecting other displayed objects (e.g. reordering the bars on a bar-chart, rescaling the axis on a graph or other diagram-specific modifications)
- *selection and linked highlighting* - when we have multiple diagrams displaying different views of the same data set, selecting a subset of the data on one diagram (e.g. a bar on a bar-chart) highlights that particular part of the data on another diagrama (e.g. a set of points on a scatter-plot)

- *linked analysis*- for example, selecting a subset of the data triggers a reactive analysis, creating a statistic model (regression, scatter-plot, spread-diagram) using that specific data set

2.2.6 Interactivity in Grafana

While it is hard to find a tool, which possesses all the before-mentioned interactivity capabilities, Grafana provides certain features to enable efficient user-interaction for the good of solid understanding of the data.

2.2.6.1 Time Range Controls

Grafana provides numerous ways to interactively manage the time ranges of the data being visualized, both at the Dashboard-level and the Panel-level.

- what is Grafana?
  - dashboard
  - panel
  - datasource
  - queries
  - query editor (because of extended ui - dynamic parameter listing))
- connects with many main datasources
- customizable, can wtite own plugins (panel, datasource, app)
- interactivity features
  - what can be meant as interactivity in this context (retelab2 lab segedlet)
  - variables, templating
  - setting the time-range
  - customizing builtin Graph Panel
  - data-links in Graph Panel
  - drill-down links

2.3 JSON

- JavaScript Object Notation
- lightweight format for storing and transporting data (w3school)
- often used when data is sent from a server to a web page (w3school)
- ”self-describing” and easy to understand
- syntax rules (need good source, currently w3school)
  - data is in name/value pairs

  - data is separated by commas
  - curly braces hold objects
  - square brackets hold arrays
- example

2.4 REST API

Chapter 3

Design

3.1 Architecture

insert architecture diagram

- why do we need a gateway
- how can we access data from RapidMiner WebService
- why is it good to have a python compoment between Grafana and MySQL
  - we can customize it better, what we see from the database
  - can implement business logic, only see business-relevant projections, granularity of the data
  - can aggregate data from different backends
  - can aggregate data with outsider APIs (POC implementation for this use-case)

3.2 Components

- Responsibilities
- Interfaces?

- Grafana
- proxy/gateway
- python-datasource
  - python-datasource
  - mysql
  - weather-api
- RapidMiner stack
  - rapidminer-server
  - job-agent

  - database
- Grafana datasource plugin (extended API - parameters)
  - extended API for asking for available parameters
  - extended GUI, that dynamically lists available parameters
- Grafana panel plugin

Chapter 4

Implementation

4.1 Gateway

4.2 Pros

4.3 Cons

Chapter 5

Evaluation

Chapter 6

Future work

- Integrate proxy and Grafana RapidMiner datasource into one

Chapter 7

Related works

interactive-piechart-panel (github/eastcirclek) see notebook

Chapter 8

Summary

Acknowledgements

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Bibliography