

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Katedra Informatyki



PROJEKT INŻYNIERSKI

**INTERFEJS GRAFICZNY DLA SYSTEMU
UCZENIA MASZYNOWEGO Z
MOŻLIWOŚCIĄ WIZUALIZACJI WIEDZY**

**MICHAŁ DRZAŁ
MARCIN PAŚKO**

OPIEKUN:
dr inż. Bartłomiej Śnieżyński

Kraków 2014

OŚWIADCZENIE AUTORÓW PRACY

OŚWIADCZAMY, ŚWIADOMI ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZY PROJEKT WYKONALIŚMY OSOBIŚCIE I SAMODZIELNIE W ZAKRESIE OPISANYM W DALSZEJ CZĘŚCI DOKUMENTU I ŻE NIE KORZYSTALIŚMY ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W DALSZEJ CZĘŚCI DOKUMENTU.

.....

PODPIS

1. Cel prac i wizja produktu

Problem, jaki stał przed zespołem do zrealizowania to stworzenie interfejsu graficznego do istniejącego programu konsolowego AQ21. Problemem wymaganym przez nasz zespół do rozwiązania był też brak wizualizacji danych zwracanych w wyniku działania programu AQ21. Sposób działania naszego rozwiązania miał być wzorowany na programie Weka, jednak miał być znacznie mniej rozbudowany. Początkowo pozornie największym zagrożeniem było zaimplementowanie parsera formatu AQ21. W rzeczywistości okazał się nim brak pełnego wyobrażenia zespołu co do stopnia złożoności zaimplementowania optymalizacji diagramu GLD.

2. Zakres funkcjonalności

Użytkownikiem produktu jest osoba wykonująca eksperymenty przy pomocy programu AQ21. Użytkownik używa naszego produktu jako wygodne medium komunikacji z tym programem oraz dodatkowo używa go do wizualizacji. Zewnętrznym systemem jest tutaj program AQ21. Do komunikacji z nim wymagane jest użycie specyficznego dla niego formatu danych.

Najważniejsze wymagania to:

- Możliwość wykonywania eksperymentów
- Wizualizacja wiedzy za pomocą zoptymalizowanego diagramu GLD
- Persystencja eksperymentów

3. Wybrane aspekty realizacji

System podzielono na dwie główne części. Jedną z nich jest interfejs przeglądarkowy (Frontend) umożliwiający sterowanie systemem przez użytkownika i dokonujący wizualizacji wiedzy. Drugą częścią jest serwis webowy (Backend) opakowujący program AQ21 i udostępniający możliwość persystencji danych w bazie danych. Po stronie Frontendu użyto języka JavaScript ponieważ jest najpowszechniejszym językiem do oprogramowania strony internetowej, paradygmatu publish-subscribe ponieważ umożliwiło łatwą dystrybucję stopniowo przetwarzanych danych. Po stronie Backendu użyto języka Java z powodu subiektywnych preferencji zespołu, dokumentowej bazy danych z powodu braku konieczności utrzymywania mapowań obiektowo-relacyjnych.

4. Organizacja pracy

Nasz zespół składał się z dwóch osób, jedna osoba (Michał Drzał) była odpowiedzialna za rozwój interfejsu przeglądarkowego. Druga osoba (Marcin Paśko) była odpowiedzialna za stworzenie serwisu webowego. W pierwszym etapie prac od maja do września 2013 roku, spotykaliśmy się z klientem raz na 5 tygodni. Od momentu gdy posiadaliśmy już przejrzysty interfejs użytkownika, planowaliśmy spotykać się z klientem i menadżerem na przemian raz w tygodniu. Do

zarządzania częścią ważniejszych zadań posłużyła JIRA, pozostałymi zadaniami każdy członek zespołu zarządzał we własnym zakresie.

5. Wyniki projektu

Udało się uzyskać oprogramowanie dobrej jakości, które co prawda nie spełniało wszystkich wymagań funkcjonalnych przyjętych na początku projektu, jednak zostało bardzo dobrze przyjęte przez klienta. Poprawne działanie najbardziej złożonych aspektów funkcjonalności zostało potwierdzone dokładnymi testami. Uzyskana dokumentacja projektu mogła by być lepszej jakości, gdyby była uzupełniana i aktualizowana na bieżąco już od samego początku istnienia projektu.

Materiały źródłowe

- [1] Dr Janusz Wojtusiak. AQ21. <http://www.mli.gmu.edu/software>, 2005.
- [2] University of Waikato. Weka. <http://www.cs.waikato.ac.nz/ml/weka/>, 2013.
- [3] Janusz Wojtusiak. Machine Learning and Inference Laboratory: AQ21 User's Guide. George Mason University, 2005.
- [4] Paweł Cichosz. *Systemy uczące się*. Wydawnictwa Naukowo-Techniczne Warszawa, 2000.
- [5] Dojo Toolkit. <http://dojotoolkit.org/>.
- [6] D3. <http://d3js.org/>.
- [7] CoffeeScript. <http://coffeescript.org/>.
- [8] RequireJS. <http://requirejs.org/>.
- [9] MathJax. <http://www.mathjax.org/>.
- [10] ANTLR. <http://www.antlr.org/>.
- [11] OrientDB. <http://www.orienttechnologies.com/orientdb/>.
- [12] Jersey. <https://jersey.java.net/>.
- [13] Jetty. <http://www.eclipse.org/jetty/>.
- [14] Jackson. <http://jackson.codehaus.org/>.
- [15] Maven. <http://maven.apache.org/>.
- [16] NetBeans. <http://netbeans.org/>.