

**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 do największych zagrożeń zaliczyć można było:

- Dużą ilość zadań w stosunku do posiadanych zasobów.
- Brak doświadczenia członków zespołu w zakresie implementacji webserwisów.
- Możliwe trudności w zaimplementowaniu parsera formatu AQ21

W rzeczywistości największym zagrożeniem okazał się brak pełnego wyobrażenia zespołu co do tego, jak bardzo złożone jest zaimplementowanie optymalizacji diagramu GLD.

2. Zakres funkcjonalności

Użytkownikiem produktu jest osoba wykonująca eksperymenty przy pomocy programu AQ21. Użytkownik używa naszego produktu jako medium komunikacji z tym programem, które ma zastąpić edytor tekstowy i konsolę. Dodatkowo użytkownik używa produktu do wizualizacji danych wejściowych i wyjściowych eksperymentu. Zewnętrznym systemem jest tutaj program AQ21. Do komunikacji pomiędzy nim a naszym systemem wymagane jest użycie specyficznego dla niego formatu danych. System powinien być dostępny na komputerze stacjonarnym wyposażonym w przeglądarkę internetową.

Najważniejsze wymagania funkcjonalne 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. Do persystencji danych użyto dokumentowej bazy danych OrientDB, ponieważ nie wymaga ona

utrzymywania mapowań obiektowo-relacyjnych w przeciwieństwie do relacyjnych baz danych. Do wygenerowania parsera formatu AQ21 użyto generatora ANTLR ponieważ jest najlepszym wyborem dla rodziny technologii Java.

Do komunikacji pomiędzy Frontendem a Backendem użyto formatu JSON ze względu na łatwość kodowania/dekodowania po stronie zarówno Frontendu, jak i Backendu.

4. Organizacja pracy

Nasz zespół składał się z dwóch osób. Podział prac zespołu był następujący. Zadania Michała:

- Rozwój Frontendu i Dokumentacja techniczna Frontendu
- Kontakty z Klientem
- Dokumentacja użytkownika z wyjątkiem Opisu API REST
- Szkic interfejsu użytkownika, opis iteracji i Podsumowanie pracy inżynierskiej.

Zadania Marcina:

- Rozwój Backendu i Dokumentacja techniczna Backendu
- Kontakty z Menadżerem
- Dokumentacja procesowa z wyjątkiem Szkicu interfejsu użytkownika, opisu iteracji i podsumowania
- API REST (API usług serwisu webowego)

Nasza praca została wykonana w dwóch etapach. W pierwszym etapie trwającym do września 2013 roku, spotykaliśmy się z klientem raz na 5 tygodni. W trakcie trwania tego etapu głównie były analizowane wymagania oraz tworzone prototypy. Drugi etap nastąpił od momentu gdy posiadaliśmy już prototyp interfejsu użytkownika. Pracowaliśmy wtedy iteracyjnie, spotykając się na przemian z Menadżerem i Klientem. Do zarządzania częścią ważniejszych zadań posłużyła JIRA, pozostałymi zadaniami każdy członek zespołu zarządzał we własnym zakresie. Wszystkie zadania do wykonania w trakcie danej iteracji były zapisywane w notatkach po spotkaniu z Klientem bądź Menadżerem.

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 jednostkowymi oraz zostało zatwierdzone przez Klienta. Dokumentacja projektu składa się z trzech części:

- Dokumentacji Procesowej. Opisuje wyniki i przebieg procesu wytwarzania systemu

- Dokumentacji Technicznej. Dokładnie opisuje rozwiązania zastosowane w projekcie i przybliża zasadę jego działania.
- Dokumentacji Użytkownika. Przedstawia sposób użytkowania systemu.

Aplikacja umożliwia wczytanie pliku AQ21, import danych z CSV. Umożliwia wykonanie eksperymentu przy użyciu programu AQ21. Istnieje możliwość zapisu i odczytu eksperymentu z bazy danych. W zakresie wizualizacji danych, aplikacja umożliwia wyświetlanie danych wejściowych w formie Scatterplota oraz tabeli. Aplikacja umożliwia także wypisanie reguł logicznych w formie wzoru i oczywiście wyświetlenie danych wyjściowych w postaci zoptymalizowanego diagramu GLD. System może zostać uruchomiony lokalnie bądź na zdalnym serwerze. Uzyskana dokumentacja projektu przedstawia wiele do życzenia i mogła by być lepszej jakości, gdyby była uzupełniana i aktualizowana na bieżąco już od samego początku istnienia projektu.

Materialy 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/>.