

---

# AKADEMIA GÓRNICZO-HUTNICZA

Wydział Elektrotechniki, Automatyki, Elektroniki i Informatyki



KATEDRA INFORMATYKI

## Przebieg projektu WEB Dispatch Rider

*Opis przebiegu prac nad projektem semestralnym  
z Inżynierii Oprogramowania*

Wersja **1.0** z dnia **28 maja 2012**

*Kierunek, rok studiów*

Informatyka, rok III

*Przedmiot*

Inżynieria Oprogramowania

*Prowadzący przedmiot*

dr inż. Małgorzata Żabińska-Rakoczy

dr inż. Jarosław Koźlak

*rok akademicki:* 2011/2012

*semestr:* letni

Zespół autorski:

Szymon Kasiński

Dariusz Mydlarz

# Spis treści

<b>1. Wstęp</b>	<b>3</b>
<b>2. Historia prac</b>	<b>4</b>
2.1. Marzec 2012	4
2.1.1. 12 marca	4
2.1.2. 19 marca	4
2.1.3. 26 marca	6
2.2. Kwiecień 2012	6
2.2.1. 2 kwietnia	6
2.2.2. 16 kwietnia	6
2.2.3. 23 kwietnia	6
2.3. Maj 2012	7
2.3.1. 7 maja	7
2.3.2. 14 maja	7
2.3.3. 21 maja	7
2.3.4. 28 maja	7
<b>3. Harmonogram dalszej pracy</b>	<b>8</b>
3.1. Czerwiec 2012	8
3.1.1. 4 czerwca	8
3.1.2. 11 czerwca	8
3.1.3. 18 czerwca	8

Niniejsze opracowanie powstało w trakcie i jako rezultat zajęć dydaktycznych z przedmiotu wymienionego na stronie tytułowej, prowadzonych w Akademii Górniczo-Hutniczej w Krakowie (AGH) przez osobę (osoby) wymienioną (wymienione) po słowach "Prowadzący zajęcia" i nie może być wykorzystywane w jakikolwiek sposób i do jakichkolwiek celów, w całości lub części, w szczególności publikowane w jakikolwiek sposób i w jakiegokolwiek formie, bez uzyskania uprzedniej, pisemnej zgody tej osoby (tych osób) lub odpowiednich władz AGH.

Copyright ©2012 Akademia Górniczo-Hutnicza (AGH) w Krakowie

# 1. Wstęp

Niniejszy dokument jest raportem dotyczącym prac wykonanych w ramach projektu semestralnego z przedmiotu Inżynieria Oprogramowania, prowadzonego na 6 semestrze stacjonarnych studiów na kierunku Informatyka na Wydziale Elektrotechniki, Automatyki, Informatyki i Elektroniki w Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. Opiekunami projektu są Państwo dr inż. Małgorzata Żabińska-Rakoczy oraz dr inż. Jarosław Koźlak. Wykonawcami projektu są brali studenci Szymon Kasiński i Dariusz Mydlarz w roku akademickim 2011/2012 (semestr letni).

Tematem projektu jest stworzenie GUI Webowego dla istniejącego systemu planowania transportu Dispatch Rider.

## 2. Historia prac

### 2.1. Marzec 2012

#### 2.1.1. 12 marca

Po wcześniejszym zapoznaniu się z zaproponowanymi przed prowadzącymi tematami ostatecznie dokonaliśmy wyboru tematu: "GUI Webow dla systemu do planowania transportu (Dispatch Rider)". Na następne zajęcia zobowiązaliśmy się zapoznać z zasadami działania oraz architekturą dotychczas istniejącego systemem.

#### 2.1.2. 19 marca

Na te zajęcia przygotowaliśmy raport dotyczący istniejącego systemu. Jego treść prezentujemy poniżej.

#### Ustalenia

1. Otrzymany projekt nie zachowuje się deterministycznie pod różnymi platformami. Pod Linuxem (Ubuntu) uruchomione zadanie nie oblicza się, natomiast pod Windowsem (7) GUI uruchamia się niedeterministycznie. Nie jesteśmy w stanie określić co powoduje owe problemy, jednakże mogą one uniemożliwić rozszerzenie funkcjonalności GUI. Faktem istotnym jest również to, że apłęt zamiast uruchamiać się w przeglądarce, raportuje błąd i nie uruchamia się.
2. W czasie analizy szczególną uwagę przyłożyliśmy do opisu problemów z którymi spotkały się zespoły wcześniej realizujące projekty. Z ich uwag wynika iż DispatchRider posiada błędy, które w szczególności mogą skutkować brakiem pliku wynikowego, nieuwzględnieniem sieci transportowej, nieuwzględnieniem czasu utworzenia holonu.
3. Dane wykorzystywane do wizualizacji wyników oraz tworzenia statystyk, są przekazywane za pośrednictwem pliku wyjściowego generowanego przez DR. Otwiera to możliwość zmiany koncepcji tworzenia GUI, ponieważ moduł nie jest w bezpośredni sposób powiązany z kodem DispatchRidera. Jednocześnie powoduje to konieczność przemysłowego zaprojektowania metody przekazywania owych danych do GUI.

## Wnioski

1. W świetle wyżej zarysowanych problemów, chcielibyśmy zaproponować pewną zmianę koncepcji tworzenia graficznego interfejsu użytkownika.
2. Chcielibyśmy rozwiązać problemy z uruchamianiem GUI na różnych systemach. Mimo iż aplety Java powinny uruchamiać się bezproblemowo, empirycznie sprawdziliśmy że GUI nie działa prawidłowo na poszczególnych systemach operacyjnych. Aby zapobiec tym problemom, chcielibyśmy stworzyć nową aplikację webową i wykorzystać przy tym takie technologie jak: HTML5, CSS, JavaScript. Z naszego doświadczenia wynika iż dobrze zaprojektowane z użyciem w.w. GUI działać będzie na każdym systemie i każdej przeglądarce. Powodem dla którego preferujemy akurat te technologie, a nie aplety Javy, jest ich nowoczesność - pragniemy wykorzystać HTML5, zapewniający olbrzymie możliwości, choćby w dziedzinie modelowania i prezentacji grafów. Z przykrością musimy stwierdzić, iż dotychczasowe sposoby wizualizacji grafów w interfejsie DispatchRidera charakteryzowały się pewnym brakiem czytelności, oraz interaktywności.
3. Budowana przez nas aplikacja podobnie jak dotychczasowe projekty GUI, korzystałaby z plików wyjściowych DispatchRidera. Oddzielenie interfejsu użytkownika od głównego ciała programu pozwala na komfortową pracę, bez konieczności zagłębiania się w kod głównego programu i jego modyfikacji, co mogłoby nawet uniemożliwić jakiegokolwiek widoczne postępy w rozwijaniu przez nas dotychczasowego GUI.
4. Projektując GUI z użyciem naszych technologii, chcielibyśmy odtworzyć zaimplementowane w wersji z apletami Javy funkcjonalności, mając jednak stuprocentową pewność że w przeciwieństwie do niej, nasz interfejs będzie w pełni użytkowalny, i będzie możliwe jego uruchomienie na każdym systemie i przeglądarce obsługującej HTML5.
5. Chcielibyśmy skupić się zwłaszcza na problemie prezentacji tras, zwracając dużą uwagę na ergonomię działania programu. Użycie w.w. technologii umożliwia tworzenie interaktywnych prezentacji grafu na które kładziemy dużą uwagę, oraz umożliwia właściwie nieograniczone możliwości modyfikowania ich, zmiany perspektywy oraz ilości przekazywanych informacji. W chwili obecnej, interfejs graficzny jest wyjątkowo nieergonomiczny, co wynika choćby z faktu iż jego uruchomienie jest czynnością niedeterministyczną. Mamy nadzieję że nasz pomysł warstwy prezentacji spowoduje jakościową zmianę w sposobie obsługi systemu.
6. Naszym zdaniem, kontynuacja istniejących implementacji graficznego interfejsu może przerodzić się w klasyczny marszu ku klęsce. Aby temu zapobiec, postulujemy zmianę koncepcji tworzenia GUI i zbudowanie podwalin dobrze zaimplementowanego, udokumentowanego, łatwego w dalszym rozwoju, a przede wszystkim działającego GUI.

## Rezultat spotkania z prowadzącymi

Prowadzący zaakceptowali nasz pomysł stworzenia GUI Webowego jako aplikacji webowej z wykorzystaniem typowych technik wykorzystywanych w tego typu projektach takich jak: PHP, HTML/CSS i JavaScript.

### 2.1.3. 26 marca

Na te zajęcia podjęliśmy się próby oceny dostępnych narzędzi JavaScriptowych umożliwiających wizualizację grafów na stronie www. W naszym teście najlepsze wyniki uzyskał JavaScript InfoVis Toolkit. Więcej na ten temat w dokumentacji projektu.

## 2.2. Kwiecień 2012

### 2.2.1. 2 kwietnia

Na pierwsze kwietniowe zajęcia dokonaliśmy analizy struktury pliku wyjściowego Dispatch Ridera, na podstawie którego będziemy chcieli przedstawić użytkownikowi wyniki działania całego systemu. Podjęliśmy decyzję, iż wyjściowy plik w formacie XML będziemy parsować do formatu JSON akceptowanego przez wybraną przez nas bibliotekę graficzną za pomocą jQuery. Więcej na ten temat w dokumentacji projektu.

### 2.2.2. 16 kwietnia

Kolejne zajęcia z powodu przerwy świątecznej odbyły się po 2 tygodniach. Tym razem przygotowaliśmy spis wymagań funkcjonalnych i niefunkcjonalnych dotyczących projektu oraz stworzyliśmy wstępny projekt wyglądu aplikacji internetowej w formie plików graficznych. Wymagania oraz projekt znajdują się w dokumentacji projektu.

### 2.2.3. 23 kwietnia

23 kwietnia przedstawiliśmy prowadzącym wstępny szkielet aplikacji webowej (działający kod). Dodatkowo zobowiązaliśmy się do rozpoczęcia prac nad **schedulerem** – skryptem w Javie mającym za zadanie zarządzać ładowanymi do systemu danymi. **Scheduler** ma za zadanie przekierować pliki wrzucane przez użytkownika w odpowiedniej kolejności do Dispatch Ridera, oraz udostępniać obliczone wyniki aplikacji internetowej, która to wyświetli je zarówno w formie tekstowej jak i graficznej (w postaci grafu).

## 2.3. Maj 2012

### 2.3.1. 7 maja

Tym razem, ponownie po 2 tygodniach przerwy (z okazji świąt majowych) pokazaliśmy część aplikacji internetowej odpowiedzialnej za upload plików przez użytkownika do systemu. Jednocześnie w tym momencie posiadaliśmy pierwszą wersję działającego schedulera, monitorującego zdarzenia dziejące się w systemie.

### 2.3.2. 14 maja

W tym momencie aplikacja internetowa pozwala na edycję danych wprowadzonych do systemu przez użytkownika. Niestety, napotkaliśmy problemy związane z uruchomieniem DR z linii poleceń, przy pomocy **Schedulera**, w sytuacji gdy plik `configuration.xml` znajduje się w innej lokalizacji niż katalog główny programu. Problem ten rozwiązaliśmy poprzez implementację mechanizmu kopiowania pliku konfiguracyjnego przed uruchomieniem.

### 2.3.3. 21 maja

Aplikacja internetowa została wzbogacona w możliwość wyświetlania wyników Dispatch Ridera w formacie tekstowym. Zostało to zrealizowane przy wykorzystaniu **jQuery** i **AJAX-a**. Dane są parsowane i wyświetlane w formie tabel z podziałem dla każdego zlecenia. Kolejnym napotkanym problemem związanym z działaniem DR jest brak plików wynikowych podczas uruchamiania programu za pomocą **schedulera**. Uruchomienie programu powoduje włączenie się DR, wyświetlenie wykresu i listy jednostek obliczeniowych, lecz po zakończeniu działania nie są generowane pliki wynikowe. Ten sam plik `configuration.xml` wybrany w GUI po przetworzeniu w programie powoduje stworzenie plików wynikowych.

### 2.3.4. 28 maja

Na te zajęcia przygotowaliśmy dokumentację przebiegu projektu. Uznaliśmy, iż powinniśmy nieco istniejącą do tej pory dokumentację, w której przeplatały się informacje dotyczące projektu oraz przebiegu prac nad nim. Dodatkowo skontaktowaliśmy z poprzednimi autorami systemu celem wyjaśnienia problemów które napotkaliśmy podczas prac Schedulerem.

## **3. Harmonogram dalszej pracy**

### **3.1. Czerwiec 2012**

#### **3.1.1. 4 czerwca**

Na ten tydzień planujemy przeznaczyć czas buforowy związany z problemami, które mogą wyniknąć w trakcie tworzenia projektu, głównie mamy na myśli jakieś błędy w **schedulerze** bądź chęć dobrego poznania biblioteki graficznej i podpięcia do niej naszych danych.

#### **3.1.2. 11 czerwca**

Tego dnia chcielibyśmy przedstawić wersję kandydującą do ostatecznej (z działającym schedulerem i poprawnym wyświetlaniem grafu). Przeprowadzimy wówczas testy dotyczące działania aplikacji, a także doszlifujemy ostatnie braki.

#### **3.1.3. 18 czerwca**

Na ostatnich zajęciach planujemy oddać działający projekt wraz z dokończoną dokumentacją, a w zamian otrzymać satysfakcjonującą obie strony ocenę z zajęć.