



**AGH**

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Fizyki i Informatyki Stosowanej

---

## **Praca inżynierska**

**Jarosław Cierpich  
Arkadiusz Kasprzak**

kierunek studiów: **informatyka stosowana**

# **Rozbudowa i uaktualnienie oprogramowania systemu GGSS detektora ATLAS TRT**

Opiekun: **dr hab. inż. Bartosz Mindur**

**Kraków, styczeń 2020**

### Oświadczenie studenta

Uprzedzony(-a) o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2018 r. poz. 1191 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony(-a) o odpowiedzialności dyscyplinarnej na podstawie art. 307 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.) „Student podlega odpowiedzialności dyscyplinarnej za naruszenie przepisów obowiązujących w uczelni oraz za czyn uchybiający godności studenta.”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Jednocześnie Uczelnia informuje, że zgodnie z art. 15a ww. ustawy o prawie autorskim i prawach pokrewnych Uczelnia przysługuje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli Uczelnia nie opublikowała pracy dyplomowej w terminie 6 miesięcy od dnia jej obrony, autor może ją opublikować, chyba że praca jest częścią utworu zbiorowego. Ponadto Uczelnia jako podmiot, o którym mowa w art. 7 ust. 1 pkt 1 ustawy z dnia 20 lipca 2018 r. — Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.), może korzystać bez wynagrodzenia i bez konieczności uzyskania zgody autora z utworu stworzonego przez studenta w wyniku wykonywania obowiązków związanych z odbywaniem studiów, udostępniać utwór ministrowi właściwemu do spraw szkolnictwa wyższego i nauki oraz korzystać z utworów znajdujących się w prowadzonych przez niego bazach danych, w celu sprawdzania z wykorzystaniem systemu antyplagiatowego. Minister właściwy do spraw szkolnictwa wyższego i nauki może korzystać z prac dyplomowych znajdujących się w prowadzonych przez niego bazach danych w zakresie niezbędnym do zapewnienia prawidłowego utrzymania i rozwoju tych baz oraz współpracujących z nimi systemów informatycznych.

.....  
(czytelny podpis)

## Spis treści

<b>1. Wstęp</b>	5
1.1. Wprowadzenie do systemu GGSS	5
1.2. Cel pracy	5
<b>2. Zastosowane technologie</b>	7
2.1. Język C++	7
2.2. Biblioteki statyczne i dynamiczne	7
2.3. Narzędzie CMake	7
2.4. Język Python	7
2.5. Język powłoki bash	7
2.6. System kontroli wersji Git i portal Gitlab	7
2.7. Manager pakietów - RPM	7
2.8. Technologie wirtualizacji i konteneryzacji	7
<b>3. Stan początkowy projektu</b>	9
3.1. Architektura	9
3.2. Budowanie	9
3.3. Dostarczanie i uruchamianie	9
3.4. Kontrola wersji	9
<b>4. Stan docelowy projektu</b>	11
<b>5. Ograniczenia dostępnej infrastruktury</b>	13
5.1. Ograniczone uprawnienia w środowisku docelowym	13
5.2. Wersje kompilatorów i interpreterów	13
5.3. Wersje narzędzia budującego CMake	13
5.4. Związek projektu z wersją jądra systemu	13
<b>6. Wykonane prace</b>	15
6.1. Wykorzystanie funkcjonalności portalu Gitlab wspierających zarządzanie projektem	15
6.2. Migracja projektu do systemu kontroli wersji Git i zmiany w architekturze	15
6.3. Zastosowanie podejścia CI/CD	15

---

6.4.	Zmiana sposobu budowania aplikacji.....	15
6.5.	Budowanie i dystrybucja sterownika oraz aplikacji testującej .....	15
6.6.	Maszyna wirtualna oraz konteneryzacja - Docker .....	15
6.7.	Pomniejsze prace.....	15
6.7.1.	Integracja bibliotek napisanych w języku C z aplikacją w C++ .....	15
6.7.2.	Integracja zewnętrznej biblioteki dynamicznej z użyciem narzędzia CMake .....	15
6.8.	Dokumentacja projektu.....	15
<b>7.</b>	<b>Dalsza ścieżka rozwoju projektu .....</b>	<b>17</b>
7.1.	Wprowadzenie zautomatyzowanego systemu testowania projektu .....	17
7.2.	Migracja do nowego standardu języka C++ .....	17
7.3.	Automatyzacja procesu publikowania produktu .....	17
<b>8.</b>	<b>Podsumowanie oraz wnioski .....</b>	<b>19</b>
<b>A.</b>	<b>Dodatki/Appendixes .....</b>	<b>21</b>
A.1.	Adding new modules to the project using existing CMake templates.....	21
A.2.	Preparing virtual machine to work as a runner .....	21

# 1. Wstęp

## 1.1. Wprowadzenie do systemu GGSS

## 1.2. Cel pracy



## 2. Zastosowane technologie

2.1. Język C++

2.2. Biblioteki statyczne i dynamiczne

2.3. Narzędzie CMake

2.4. Język Python

2.5. Język powłoki bash

2.6. System kontroli wersji Git i portal Gitlab

2.7. Manager pakietów - RPM

2.8. Technologie wirtualizacji i konteneryzacji





### **3. Stan początkowy projektu**

#### **3.1. Architektura**

#### **3.2. Budowanie**

#### **3.3. Dostarczanie i uruchamianie**

#### **3.4. Kontrola wersji**



#### 4. Stan docelowy projektu



## 5. Ograniczenia dostępnej infrastruktury

5.1. Ograniczone uprawnienia w środowisku docelowym

5.2. Wersje kompilatorów i interpreterów

5.3. Wersje narzędzia budującego CMake

5.4. Związek projektu z wersją jądra systemu



## 6. Wykonane prace

6.1. Wykorzystanie funkcjonalności portalu Gitlab wspierających zarządzanie projektem

6.2. Migracja projektu do systemu kontroli wersji Git i zmiany w architekturze

6.3. Zastosowanie podejścia CI/CD

6.4. Zmiana sposobu budowania aplikacji

6.5. Budowanie i dystrybucja sterownika oraz aplikacji testującej

6.6. Maszyna wirtualna oraz konteneryzacja - Docker

6.7. Pomniejsze prace

6.7.1. Integracja bibliotek napisanych w języku C z aplikacją w C++

6.7.2. Integracja zewnętrznej biblioteki dynamicznej z użyciem narzędzia CMake

6.8. Dokumentacja projektu





## 7. Dalsza ścieżka rozwoju projektu

7.1. Wprowadzenie zautomatyzowanego systemu testowania projektu

7.2. Migracja do nowego standardu języka C++

7.3. Automatyzacja procesu publikowania produktu



## 8. Podsumowanie oraz wnioski



## A. Dodatki/Appendixes

A.1. Adding new modules to the project using existing CMake templates

A.2. Preparing virtual machine to work as a runner