



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

Rozbudowa i uaktualnienie systemu GGSS detektora ATLAS TRT

Arkadiusz Kasprzak
Jarosław Cierpich

Opiekun pracy: dr hab. inż. Bartosz Mindur, prof. AGH



- 1 Wprowadzenie do tematyki pracy**
- 2 Modyfikacja kodu źródłowego projektu**
- 3 Modyfikacja systemu budowania projektu**
- 4 Podsumowanie**



- System Stabilizacji Wzmocnienia Gazowego (GGSS - *Gas Gain Stabilization System*)
- projekt zintegrowany z systemem kontroli detektora ATLAS w CERN
- umożliwia poprawne działanie detektora TRT (*Transition Radiation Tracker*) będącego częścią ATLAS
- składa się z warstwy oprogramowania (aplikacje oraz infrastruktura) oraz sprzętu (zasilacz wysokiego napięcia, multiplekser analogowy, wielokanałowy analizator amplitudy MCA)
- warstwa oprogramowania koordynuje działanie urządzeń i umożliwia sterowanie nimi za pomocą specjalnych komend
- aplikacje i biblioteki napisane z wykorzystaniem języka C++, infrastruktura: CMake, Bash oraz Python

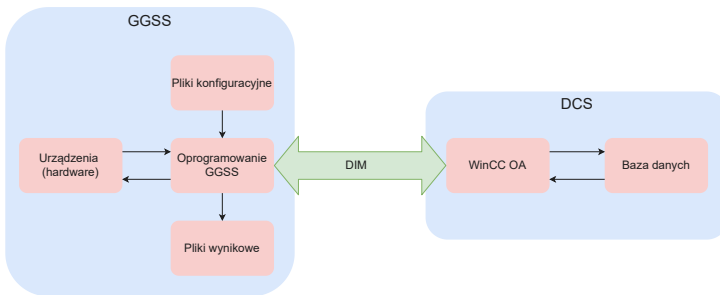


Figure: Wysokopoziomowa architektura systemu GGSS

- WinCC OA - system system typu SCADA pozwalający na sterowanie i kontrolę działania podsystemów detektora
- DIM - protokół komunikacyjny dla aplikacji rozproszonych



AGH Cele pracy

- kontynuacja pracy inżynierskiej, skupiającej się na infrastrukturze projektu
- rozbudowa przygotowanych w ramach wspomnianej pracy rozwiązań
- główny nacisk na aplikację *ggss-runner*, stanowiącą trzon systemu
- poprawa jakości kodu źródłowego oraz wprowadzenie nowych funkcjonalności
- rozbudowa infrastruktury pozwalającej na testowanie projektu
- udokumentowanie projektu (dokumentacja kodu źródłowego oraz pliki instruktażowe)
- migracja projektu - nowy komputer docelowy (początkowo planowany wyjazd do CERN, ostatecznie zrealizowana zdalnie)
- konieczność zachowania wysokiej niezawodności systemu - stosowanie praktyk takich jak *code review* i testy automatyczne



- projekt ściśle związany z infrastrukturą CERN
- aplikacje wchodzące w skład systemu działają na dedykowanym komputerze z dostępem do wymaganych urządzeń
- ograniczenia dotyczące wersji kompilatora języka C++ oraz narzędzia CMake
- ograniczone uprawnienia w środowisku docelowym
- konieczność zachowania kompatybilności wstecznej



AGH Aplikacja ggss-runner

- wielowątkowa aplikacja napisana z wykorzystaniem C++ i pakietu Boost
- zadanie: cykliczne gromadzenie danych w postaci widma poprzez komunikację z wielokanałowym analizatorem amplitudy oraz wyznaczanie na ich podstawie wartości napięcia, komunikacja z systemem WinCC OA

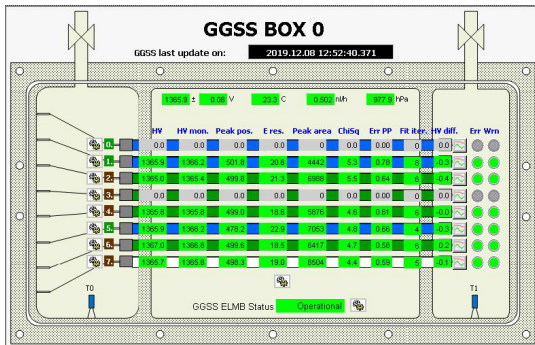


Figure: Panel dostępny w ramach systemu WinCC OA

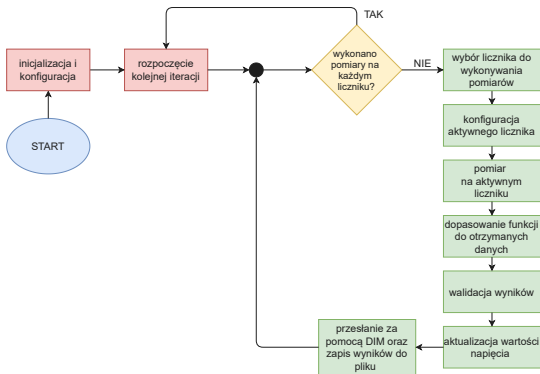


Figure: Uproszczony schemat działania aplikacji



AGH Modernizacja i poprawa jakości kodu źródłowego

- pierwotnie kod wykorzystywał standard C++03 oraz elementy C++11
- przeprowadzono migrację do standardu C++11 (pętle zakresowe, silne typy wyliczeniowe itd.)
- migracja niepełna z uwagi na ograniczenia kompilatora
- ujednolicono konwencje stosowane w kodzie (formatowanie, nazewnictwo)
- zlikwidowano niewykorzystywane lub wykomentowane fragmenty kodu
- usunięto nieliczne błędy (biblioteki *xml-lib* oraz *log-lib*)
- znaczne zmiany w 12 z 14 bibliotek wchodzących w skład systemu



- dodanie obsługi zaawansowanych komend dla zasilaczy wysokiego napięcia
- rozbudowa biblioteki odpowiedzialnej za dopasowanie krzywej do zebranych danych
- dodanie możliwości aktualizacji parametrów i zebranych danych na żądanie
- dodanie zabezpieczenia przed przepełnieniem bufora urządzenia MCA
- dodanie możliwości przywracania domyślnej kolejności liczników słomkowych



AGH

Przykład: nowe komendy dla zasilaczy



AGH Testy automatyczne

- w celu zapewnienia niezawodności i szybkiego wykrywania błędów wykorzystano testy automatyczne i metodykę TDD (*Test Driven Development*)
- testy przygotowane z wykorzystaniem biblioteki Boost.Test
- uruchamiane automatycznie po wdrożeniu zmian do repozytoriów projektu

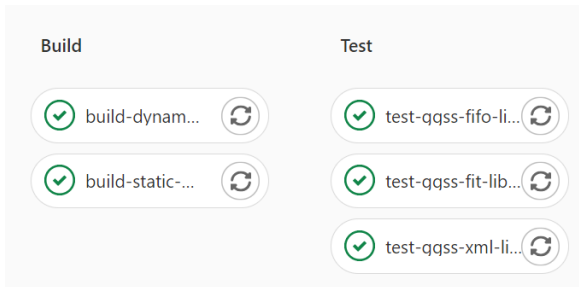


Figure: Pipeline CI/CD wykonujący testy automatyczne po wdrożeniu zmian



AGH Modyfikacja systemu budowania projektu

- system oparty o narzędzie CMake, przygotowany w ramach pracy inżynierskiej
- jego zadaniem jest obsługa projektu o hierarchicznej strukturze
- wsparcie dla testów automatycznych oraz generowania dokumentacji za pomocą narzędzia Doxygen
- system oparty o tzw. szablony - pliki *.cmake* zawierające często wykorzystywane fragmenty kodu
- zmiana sposobu implementacji szablonów - wykorzystanie funkcji i makr

```
ggss_build_static_library(  
    TARGET_NAME "thread"  
    DEPENDENCY_PREFIX "${CMAKE_CURRENT_SOURCE_DIR}/.."   
    DEPENDENCIES "sigslot" "ggss-util-libs/log"  
)
```



- wszystkie podstawione wymagania zostały spełnione
- stosowanie nowoczesnych praktyk takich jak testy automatyczne pozytywnie wpłynęło na proces rozwoju oprogramowania
- liczne testy w środowisku docelowym potwierdzają poprawność wprowadzonych zmian
- możliwy dalszy rozwój projektu - modernizacja kodu, zmiany na poziomie architektonicznym



- 5 Modyfikacja architektury projektu**
- 6 Modyfikacja infrastruktury projektu**
- 7 Aplikacje do testów urządzeń**
- 8 Testy projektu**
- 9 Podsumowanie**



AGH

Modyfikacja architektury projektu



AGH

Mechanizm ciągłej integracji i dostarczania



AGH

Automatyzacja i centralizacja wersjonowania projektu



AGH

Automatyzacja pracy z submodułami



AGH

Pozostałe zmiany



AGH

Aplikacje do testów urządzeń



AGH

Testy projektu



AGH

Podsumowanie