

Paweł DRZYMAŁA, Henryk WELFLE, Sławomir WIAK, Jakub RUTKOWSKI  
Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych

## **SYSTEM PRZETWARZANIA I ANALIZY DANYCH PROCESOWYCH Z WYKORZYSTANIEM BAZ DANYCH**

**Streszczenie.** Opracowanie przedstawia możliwość tworzenia aplikacji sieciowej do przeglądania i analizy danych procesowych, gromadzonych przez system klasy SCADA. W przedstawionym rozwiązaniu wykorzystany został system Firmy Siemens – WinCC jako platforma gromadzenia danych. Główne funkcje opracowywanego systemu dostępu obejmują zbieranie aktualnych danych (pomiarów), ich analizę, wizualizację, w tym monitorowanie alarmów produkcyjnych, oraz archiwizację gromadzonych danych. Przedstawiono przykład praktycznego zastosowania integracji nowoczesnych technologii w celu zbudowania wydajnego, przemysłowego rozwiązania.

**Słowa kluczowe:** Systemy Zarządzania Bazami Danych, SCADA, Siemens, WinCC, MS SQL Server 2005, Aplikacje WWW, Google Web Toolkit

## **THE INDUSTRY DATA ANALYSIS SYSTEM BY DATABASE**

**Summary.** The paper presents the possibility of creating of network application to preview and the analysis of data process, accumulated by system of SCADA class (Supervisory Control And Data Acquisition). The Siemens – WinCC system was used as accumulating data platform. The main functions of presented system is the access to the current data (the measurements), their analysis, visualization, the monitoring of the productive alarms as well as the filing of accumulated data. The paper presents the example of practical use of modern technologies integration to built the effective, industrial solution..

**Keywords:** Database Management System, SCADA, Siemens, WinCC, MS SQL Server 2005, WEB Application, Google Web Toolkit

## 1. Platforma WinCC firmy Siemens

WinCC jest systemem SCADA dla platformy Windows[1], przeznaczonym do wizualizacji i sterowania procesami przemysłowymi. Znajduje zastosowanie we wszystkich gałęziach przemysłu, począwszy od pojedynczych jednostek roboczych do rozproszonych systemów z redundantnymi serwerami i krzyżowymi połączeniami sieciowymi z klientem sieciowym. WinCC jest systemem wymiany informacji dla pionowej integracji firmy bazującej i używającej koncepcji Plant Intelligence w celu zapewnienia większej przezroczystości w produkcji.

Do podstawowej konfiguracji systemu włączone są standardowe funkcje, takie jak: sygnalizacja, potwierdzanie wystąpienia zdarzeń, archiwizacja komunikatów oraz mierzonych wartości, wyświetlanie logów, konfiguracja danych, administracja i wizualizacja.

Platforma WinCC jest zaprojektowana dla technologicznych i typowo przemysłowych procesów. Ma budowę modułową i pozwala na łatwą rozbudowę. Rozwiązanie znajduje zastosowanie zarówno w technicznych aplikacjach jednostanowiskowych, jak i w kompleksowych wielostanowiskowych systemach rozproszonych, zawierających wiele serwerów i wielu klientów (również opartych na sieci Internet), na przykład wizualizacja procesów w przedsiębiorstwach produkcyjnych.

Główną cechą charakterystyczną platformy jest połączenie automatyzacji produkcji i procesów. System WinCC zyskał miano standardu przemysłowego w dziedzinie wizualizacji procesów zarówno jako autonomiczny system SCADA, jak i komponent kontroli procesów systemowych SIMATIC PCS7 lub Spectrum Power CC dla zarządzania energią. Duża ilość opcji WinCC (WinCC Options) i dodatków (Add-ons) pozwala na dostosowanie systemu do specyficznych wymagań, np. dla przemysłu farmaceutycznego lub stacji uzdatniania wody.

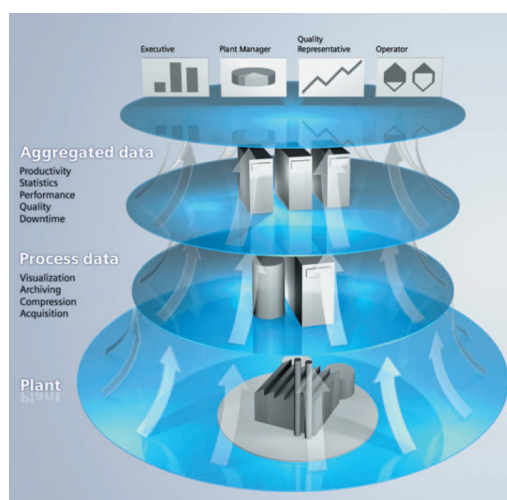
Zakres zastosowań WinCC obejmuje najważniejsze kanały komunikacyjne do połączeń ze sterownikami SIMATIC (rodzina S5, S7), a także kanały niededykowane, takie jak PROFIBUS-DP/FMS (Standard w komunikacji za pomocą magistrali używany do kontroli wszemkich elementów automatyki), DDE (Dynamic Data Exchange – technologia komunikacji między aplikacjami w środowisku Windows lub OS/2) i OPC (OLE for Process Control – standard komunikacyjny stosowany w automatyce przemysłowej, wykorzystywany w warstwie biznesowej oraz zarządzania). Możliwe jest również skonfigurowanie komunikacji za pomocą innych kanałów przez odpowiednie dodatki do systemu (Add-ons). Odkąd wszyscy producenci sterowników zaczęli stosować serwery OPC dla swoich produktów, możliwości połączeń z WinCC stały się praktycznie nieograniczone.

Koncepcja SIMATIC Plant Intelligence zintegrowana z aplikacjami na każdym poziomie przedsiębiorstwa oferuje kompletne i skalowalne rozwiązanie pochodzące z jednego źródła. Plant Intelligence jest oparte na przetwarzaniu i przepływie danych w taki sposób, aby

w czasie rzeczywistym wygenerować informacje wspomagające proces decyzyjny w przedsiębiorstwie. Taka procedura prowadzi do redukcji kosztów, lepszego wykorzystania zasobów oraz zapobieganiu awariom i zwiększenia jakości wyrobów.

### 1.1. Technologia Plant Intelligence

Plant Intelligence łączy poziom systemu SCADA z poziomem MES (Manufacturing Execution System – określenie typu oprogramowania używanego do monitorowania i sterowania procesami w przemyśle), dzięki czemu uzyskuje się pełny obraz procesów produkcyjnych w dowolnych przedziałach czasowych przez wydajne zbieranie, archiwizowanie, kompresję, analizę i dystrybucję danych produkcyjnych. Na podstawie ogromnej liczby składowanych danych, pochodzących z systemu możliwe jest zbudowanie efektywnej hurtowni danych. Dzięki zastosowaniu dodatkowych narzędzi drążenia danych (Data Mining) możliwe jest znajdowanie ukrytych zależności w procesie produkcji. Wiedza ta pozwala na optymalizację procesu oraz częściowe prognozowanie ewentualnych problemów.



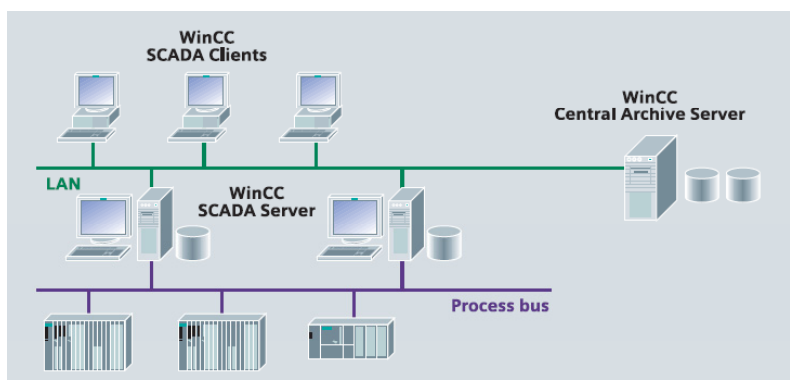
Rys. 1. Przepływ danych Plant Intelligence [1]

Fig. 1. Plant Intelligence – workflow of data [1]

### 1.2. Moduł Historian

SIMATIC WinCC zawiera potężny [1], skalowalny moduł Historian oparty na środowisku bazodanowym Microsoft SQL Server 2005, który w podstawowej wersji systemu używany jest jako centralny serwer wymiany danych. Możliwość oceny danych na stacjach klienckich, otwarte interfejsy (baza danych i interfejsy do programowania) i wiele możliwości dla integracji IT (Information Technology) i biznesu są podstawą

funkcjonalności Plant Intelligence. Dzięki temu możliwe jest połączenie poziomego produkcji i poziomego zarządzania – systemy MES i ERP<sup>1</sup>.

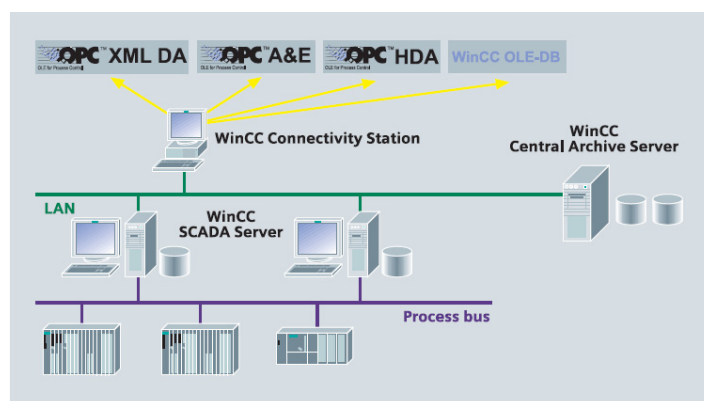


Rys. 2. Architektura systemu składowania i archiwizowania danych procesowych w systemie Siemens WinCC [1].

Fig. 2. The system architecture of storing and archives of process data in Siemens WinCC system [1]

### 1.3. WinCC Connectivity Station

Rolę WinCC Connectivity Station w architekturze stanowi stacja z zainstalowanym dodatkiem WinCC Connectivity Pack. Użycie takiego rozwiązania powoduje, że możliwy staje się dostęp do danych WinCC z komputerów, które nie mają zainstalowanego oprogramowania Siemens WinCC przez OLE-DB (Object Linking and Embedding, Database – Interfejs programistyczny wspierany przez Microsoft służący do uzyskania dostępu do danych) i OPC.



Rys. 3. Użycie Connectivity Station w celu zapewnienia dostępu do danych klientom bez zainstalowanego WinCC [1]

Fig. 3. Connectivity Station use to access to data for clients without WinCC installation [1]

<sup>1</sup> Enterprise Resource Planning – Określenie typu systemu informatycznego mającego na celu wspomaganie procesu planowania produkcji)

Interfejsy Connectivity Station pozwalają również na dostęp poprzez klienta OPC:

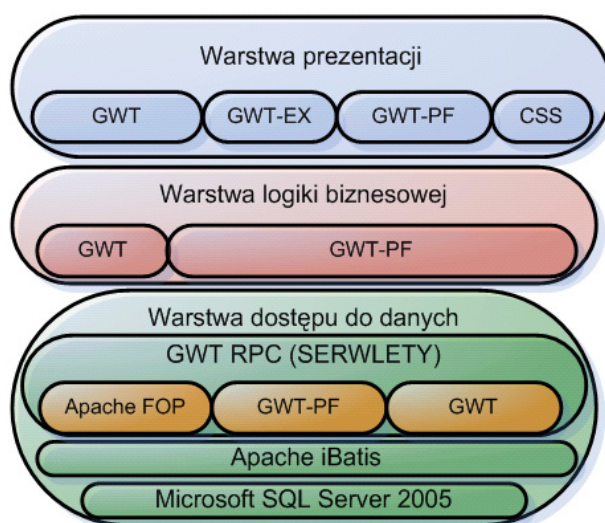
- OPC DA server: Tagi – Dane procesowe.
- OPC HDA server: Archiwalne dane procesowe.
- OPC-A&E server: Dane o alarmach.

## 2. Aplikacja WinCCReporter

Zadanie zbudowania internetowej aplikacji umożliwiającej przeglądanie i raportowanie danych procesowych wymaga skomplikowanego procesu integracji z systemami źródłowymi. Siemens WinCC jest oparty na systemie zarządzania bazą danych SQL Server 2005. Posiada dość rozbudowane i skomplikowane struktury, a poszczególne elementy są porozrzucone pomiędzy odrębne bazy danych. Wydobycie danych z systemu WinCC nie jest sprawą prostą ze względu na to, że Siemens kompresuje dane archiwalne i w tej postaci przetrzymuje je w bazie danych. W opisywanej aplikacji dane są dekompresowane w locie, co nie stanowi problemu przy rozsądnych przedziałach czasowych pobieranych danych i przechowywanych w bazie.

### 2.1. Architektura trójwarstwowa

Architektura trójwarstwowa stanowi inną rozszerzoną formę podejścia klient-serwer. Istotą tej struktury jest rozdzielenie elementów prezentacji, elementów odpowiedzialnych za logikę biznesową oraz najniższych warstw dostępu do danych. Dzięki takiemu podejściu możliwe staje się wydzielenie modułów, a zmiany dokonywane w jednej z warstw nie powinny mieć wpływu na inne. Warstwy komunikują się między sobą za pomocą zdefiniowanych interfejsów i tylko dzięki nim mogą na siebie oddziaływać. Takie rozwiązanie pomaga utrzymać kontrolę nad kodem, a dodatkowo utrzymuje całą logikę w należyтым porządku przez zapewnienie, że tylko elementy warstw sąsiednich mogą na siebie oddziaływać. Na rys. 4. przedstawiono architekturę aplikacji z zaznaczonymi poszczególnymi technologiami używanymi w danej warstwie.



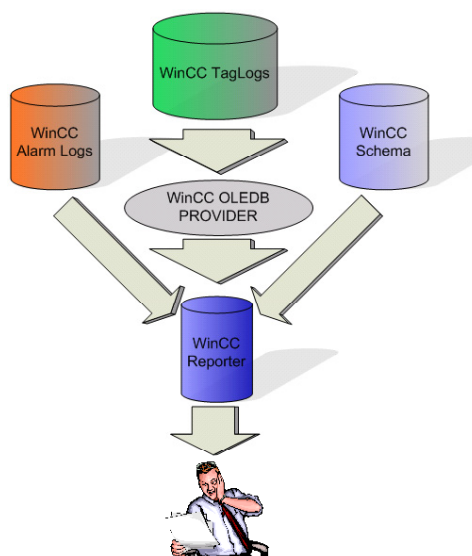
Rys. 4. Trójwarstwowa architektura aplikacji WinCC Reporter  
 Fig. 4. WinCC Reporter – three-layer architecture

## 2.2. Integracja aplikacji z systemem WinCC

Dane wykorzystywane do raportowania i prezentacji są pobierane z trzech odrębnych baz danych wchodzących w skład systemu Siemens WinCC. Ogólny schemat integracji przedstawiony jest na rys. 5. Najważniejszą, a zarazem zawierającą najwięcej danych, jest archiwum Tagów, którymi nazywane są wartości wskazań czujników i wszelkich mierników, jakie są podłączone w przemyśle do systemu SCADA. Ze względu na możliwości częstego próbkowania tego rodzaju wartości, a tym samym dużej liczbie danych, Siemens kompresuje dane historyczne, które następnie są trzymane w tabelach w postaci binarnej. Dane kompresowane są cyklicznie według zaprogramowanego w systemie harmonogramu. Po procesie kompresji i zapisania w archiwum dane w postaci nieskompresowanej są usuwane. Aby uzyskać dostęp do danych archiwalnych, trzeba skorzystać z biblioteki dll zawierającej sterownik OLEDB dołączany do dodatku WinCC ConectivityPack.

Historia alarmów jest przechowywana w zupełnie odmiennej bazie danych, co jest związane z przyjętą modułową budową WinCC. Tabele zawierają informacje o występujących w przeszłości zdarzeniach, dodatkowo dostarczają podpowiedzi rozwiązania problemu, jeśli została taka zdefiniowana w systemie. Dane historyczne dotyczące alarmów ze względu na małą przewidywaną ilość wystąpień nie są kompresowane.

Definicje wartości mierzonych (Tagów) są zawarte w odrębnej bazie danych, przetrzymującej całą konfigurację WinCC, oznaczonej na schemacie jako WinCC Schema.



Rys. 5. Integracja systemu Siemens WinCC z budowaną aplikacją  
 Fig. 5. Integration of WinCC Siemens system with built application

### 2.3. Struktura tabel dla modułu zarządzania użytkownikami, profilami i autoryzacją

Budowana aplikacja nie posiada funkcjonalności umożliwiających modyfikację danych procesowych składowanych w systemie, z wyjątkiem modułu związanego z użytkownikami i autoryzacją treści. Moduł powinien umożliwiać przechowywanie podstawowych danych dotyczących pracownika, jego login i hasło w postaci zaszyfrowanej. Dane opisujące pracownika przechowywane są w tabeli `USERS_T`.

Aplikacja wspiera proste profilowanie użytkowników. W systemie zostały zdefiniowane następujące profile odzwierciedlające hierarchię i funkcje w typowym przedsiębiorstwie produkcyjnym:

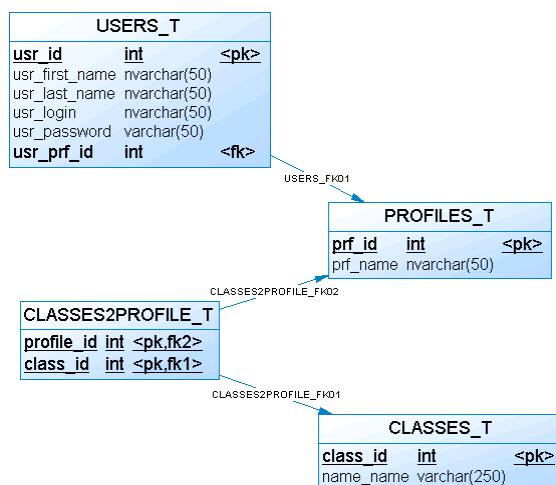
- Produkcja.
- Technicy.
- Management.
- Administracja.

Profile zdefiniowane są w tabeli `PROFILES_T`.

Z przypisanym profilem użytkownika związana jest zawartość, do jakiej ma on dostęp, po zalogowaniu się do aplikacji. Tabela `CLASSES_T` zawiera listę klas Java w aplikacji wymagających autoryzacji. Powiązanie pomiędzy profilami a uprawnieniami zrealizowane jest za pomocą wpisów w tabeli `CLASSES2PROFILE_T`.

Na rys. 6. przedstawione są podstawowe zależności pomiędzy tabelami modułu zarządzania użytkownikami i autoryzacją. Pracownik zawarty w tabeli `USERS_T` posiada pole `usr_prf_id`, które połączone jest przez klucz obcy z tabelą profili `PROFILES_T`. Profile

posiadają odwołanie przez klucz obcy do tabeli intersekcyjnej CLASSES2PROFILE\_T wiążącej profile z klasami Java przez klucz odwołujący się do pola CLASSES\_T.class\_id.



Rys. 6. Struktura tabel bazodanowych odpowiedzialna za zarządzanie użytkownikami, profilami oraz autoryzacją

Fig. 6. Database tables structure to management users, profiles and authorization

#### 2.4. Struktura tabel odpowiedzialnych za przechowywanie danych procesowych

Dane procesowe to informacje spływające z wszelkich sterowników podłączonych do linii produkcyjnej. W zależności od rodzaju mogą być próbkowane z różną częstotliwością. WinCC stosuje rozwiązanie polegające na kompresowaniu danych historycznych. Kompresja jest wywoływana cyklicznie, w momencie określonym w harmonogramie. Po skompresowaniu dane są zapisywane w formie binarnej, a tabele źródłowe są czyszczone. Rysunek 7. przedstawia strukturę tabeli zawierającej aktualne dane przed kompresją.

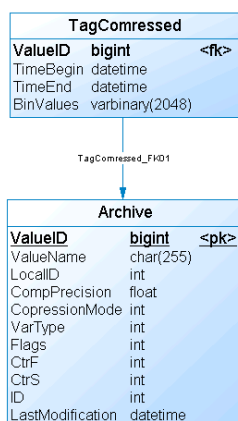
TagUncompressed	
ValueID	bigint
TimeStamp	datetime
MS	smallint
RealValue	float
Quality	int
Flags	int

Rys. 7. Tabela odpowiedzialna za przechowywanie danych procesowych w postaci nieskompresowanej

Fig. 7. Table structure to store of uncompressed process data

Rysunek 8. prezentuje strukturę tabel, do których odwołuje się aplikacja raportująca w celu pozyskania danych procesowych. Tabela TagCompressed zawiera identyfikator wartości, której szczegóły można znaleźć w tabeli powiązanej Archive. Pola TimeBegin i TimeEnd określają przedział czasowy, z którego wartości wchodzi w skład skompresowanego archiwum zapisanego binarnie w polu BinValues.





Rys. 8. Struktura tabel bazodanowych odpowiedzialna za przechowywanie danych procesowych w postaci skompresowanej

Fig. 8. Database tables structure to store of compressed process data

## 2.5. Struktura tabeli związanej z przechowywaniem alarmów

Tabela alarmów zawarta jest z w strukturze WinCC, zawiera w większości dane związane z konfiguracją systemu. Dodatkowo, obejmuje informacje o wystąpieniach sytuacji alarmowych, ich prawdopodobnych przyczynach, a także podpowiedziach dotyczących rozwiązania problemu.

W aplikacji wykorzystywane są następujące pola:

- MsgNr – Identyfikator alarmu.
- DateTime – Data wystąpienia alarmu.
- Classname – Nazwa klasy alarmu.
- Typename – Nazwa typu alarmu.
- Text – Opis zdarzenia.
- InfoText – Podpowiedź dotycząca rozwiązania problemu.

Rysunek 9. prezentuje pełną specyfikację tabeli alarmów. WinCC umożliwi konfigurację, na podstawie której generowane są automatyczne powiadomienia o wystąpieniu sytuacji alarmowej.

AlgCSDatENU			
NR	int		<pk>
AG_NR	smallint		
CPU_NR	smallint		
Priority	int		
LocaleID	int		
Class	int	AP_type	int
Type	int	AP_name	varchar(255)
Text1	varchar(255)	AP_par	varchar(255)
Text2	varchar(255)	InfoText	varchar(255)
Text3	varchar(255)	iText1	int
Text4	varchar(255)	iText2	int
Text5	varchar(255)	iText3	int
Text6	varchar(255)	iText4	int
Text7	varchar(255)	iText5	int
Text8	varchar(255)	iText6	int
Text9	varchar(255)	iText7	int
Text10	varchar(255)	iText8	int
Classname	varchar(255)	iText9	int
TxCame	varchar(255)	iText10	int
TxtWent	varchar(255)	iClass	int
TxCameNWent	varchar(255)	iCame	int
TxAck	varchar(255)	iCameNWent	int
Typename	varchar(255)	iWent	int
CrComeFore	int	iAck	int
CrComeBack	int	iType	int
CrGoFore	int	Alarm Tag	int
CrGoBack	int	AckType	smallint
CrAckFore	int	Params	int
CrAckBack	int	ServerID	bigint <pk>

Rys. 9. Tabela odpowiedzialna za przechowywanie historii alarmów

Fig. 9. Table structure to store of history alerts

## 2.6. Opis i charakterystyka zdefiniowanych tabel

Tabele [1, 2, 3, 4, 5, 6] przedstawiają zestawienie specyfikacji i opisów pól tabel wykorzystywanych w aplikacji.

Tabela 1

Struktura tabeli USERS\_T


	NAZWA	TYP	OPIS
	USR_ID	INT	KLUCZ PODSTAWOWY.
	USR_FIRST_NAME	NVARCHAR(50)	IMIĘ PRACOWNIKA
	USR_LAST_NAME	NVARCHAR(50)	NAZWISKO PRACOWNIKA
	USR_LOGIN	NVARCHAR(50)	LOGIN PRACOWNIKA
	USR_PASSWORD	VARCHAR(50)	HASŁO DOSTĘPU DO APLIKACJI ZAPISANE W POSTACI ZASZYFOWANEJ
	USR_PRF_ID	INT	ZEWNĘTRZNY IDENTYFIKATOR PROFILU. KLUCZ OBCY ODWOŁUJĄCY SIĘ DO PROFILES_T.PRF_ID

Tabela 2

Struktura tabeli PROFILES\_T


	NAZWA	TYP	OPIS
	PRF_ID	INT	KLUCZ PODSTAWOWY.
	PRF_NAME	NVARCHAR(50)	NAZWA PROFILU

Tabela 3

Struktura tabeli CLASSES\_T


	NAZWA	TYP	OPIS
	CLASS_ID	INT	KLUCZ PODSTAWOWY.
	CLASS_NAME	VARCHAR(250)	NAZWA KLASY ZAWIERAJĄCA PEŁNĄ ŚCIEŻKĘ

Tabela 4

Struktura tabeli CLASSES2PROFILES\_T



	NAZWA	TYP	OPIS
	PROFILE_ID	INT	ZEWNĘTRZNY IDENTYFIKATOR PROFILU. KLUCZ OBCY ODWOŁUJĄCY SIĘ DO PROFILES_T.PRF_ID
	CLASS_ID	INT	ZEWNĘTRZNY IDENTYFIKATOR KLASY. KLUCZ OBCY ODWOŁUJĄCY SIĘ DO CLASSES_T.CLASS_ID

Tabela 5

Struktura tabeli TagCompressed

	<b>NAZWA</b>	<b>TYP</b>	<b>OPIS</b>
	VALUEID	BIGINT	ZEWNĘTRZNY IDENTYFIKATOR WARTOŚCI PROCESOWEJ.
	TIMEBEGIN	DATETIME	DATA POCZĄTKU ZAKRESU DANYCH PRZECHOWYWANYCH W ARCHIWUM BINARNYM
	TIMEEND	DATETIME (50)	DATA KOŃCA ZAKRESU DANYCH PRZECHOWYWANYCH W ARCHIWUM BINARNYM
	BINVALUES	VARBINARY(2048)	BINARNA POSTAĆ DANYCH SKOMPRESOWANYCH

Tabela 6

Struktura tabeli Archive

	<b>NAZWA</b>	<b>TYP</b>	<b>OPIS</b>
	VALUEID	BIGINT	KLUCZ PODSTAWOWY.
	VALUENAME	CHAR(255)	NAZWA WARTOŚCI
	LOCALID	INT	IDENTYFIKATOR JĘZYKA
	COMPPRECISON	FLOAT	STOPIEŃ PRECYZJI LICZB ZMIENNOPRZECINKOWYCH ZAPISYWANYCH W SKOMPRESOWANYM ARCHIWUM
	COMPRESSIONMODE	INT	TRYB KOMPRESJI
	VARTYPE	INT	TYP ZMIENNEJ
	FLAGS	INT	FLAGI DOTYCZĄCE BIBLIOTEKI KOMPRESUJĄCEJ
	CTRF	INT	PARAMETRY DLA BIBLIOTEKI KOMPRESUJĄCEJ
	CTRS	INT	PARAMETRY DLA BIBLIOTEKI KOMPRESUJĄCEJ
	ID	INT	IDENTYFIKATOR GLOBALNY
	LASTMODIFICATION	DATE TIME	DATA OSTATNIEJ MODYFIKACJI WARTOŚCI

### 3. Logika aplikacji (procedury składowane)

#### 3.1. Dekompresja danych

W celu bezpośredniej dekompresji danych wykorzystana została rozszerzona procedura składowana, zaimplementowana z wykorzystaniem wbudowanej procedury `sp_addextendedproc`. Algorytm dekompresji jest zawarty w bibliotece `ccxp_decompress.dll` dostarczonej przez firmę Siemens.

```
EXEC dbo.sp_addextendedproc N'sp_ctlg_decompress', 'ccxp_decompress.dll'
```

#### 3.2. Pobieranie danych

Wszelkie odwołania do bazy danych realizowane są przez procedury składowane, dzięki czemu uzyskuje się większą elastyczność i bezpieczeństwo aplikacji. Plany wykonania procedur są buforowane na serwerze, dzięki czemu możliwe jest uniknięcie zbędnej analizy kodu. Podobnie jest w przypadku zapytań zawartych wewnątrz. Silnik bazodanowy bada zapytanie, zmieniając parametry na zmienne wiązane nawet w przypadku użycia literału, dzięki czemu eliminuje się potrzebę twardego procesu analizy kodu. Takie rozwiązanie zmniejsza obciążenie bazy danych, a co za tym idzie zwiększa jej wydajność.

Użycie procedur składowanych powoduje wzrost bezpieczeństwa aplikacji dzięki dodatkowej ochronie przeciw wstrzykiwaniu poleceń SQL.

Zmiana ciała procedury powoduje natychmiastową zmianę we wszystkich wynikach zwracanych w aplikacji. W ten sposób można uniknąć wielu błędów oraz przykrego poprawiania zapytania we wszystkich miejscach w kodzie, w którym jest ono wykorzystywane.

Umieszczenie wyrażeń DML w ciele procedury pozwala na lepszą kontrolę jakości danych, dzięki możliwości dodania dodatkowych walidacji czy mechanizmu uzupełniania danych.

- Procedura `sp_getTags` zwraca wynik dynamicznie budowanego zapytania. Wykorzystywane jest w funkcji tabelarycznej pobierającej dane do przeglądu tagów. Dynamiczna budowa zapytania była niezbędna ze względu na możliwości operowania na zwracanych danych. Funkcja obsługuje stronicowanie danych w formatkach oraz możliwość sortowania po wielu kolumnach z zachowaniem bieżącej strony.
- Procedura `sp_getTagsCount` zwraca ilość wszystkich rekordów, jakie spełniają podane kryteria. Wartość ta jest niezbędna do oszacowania ilości możliwych stron w kontrolce wyświetlającej wartości tagów. Procedura przyjmuje w parametrach identyfikator wartości oraz zakres dat.

### 3.3. Mechanizm raportowania

Na potrzeby raportowania dane muszą być zdekompresowane, aby możliwe było przeprowadzenie agregacji.

- Procedura `sp_getDailyReportData` zwraca kursor zawierający uśrednione dane na poziomie dni. Ostateczna wersja jest ograniczona do maksymalnie trzech różnych wartości. Testy wykazały znaczne problemy wydajnościowe przy dekompresowaniu i agregowaniu danych w „locie” przy większej liczbie tagów. Została zaimplementowana analogiczna procedura dla miesięcznego poziomu agregacji.
- Procedura `sp_getFormattedReportData` zwraca sformatowany zestaw danych na potrzeby wykresów i kontrolek Grid. Dane pobrane z kursora zwracanego z procedur agregujących są wstawione do zmiennej tablicowej `@ReportDataTable`. Dzięki temu rozwiązanie późniejsze manipulacje na danych mające na celu sformatowanie wyników do oczekiwanej struktury odbywają się w pamięci, bez odwołania do kursora.

### 3.4. Mechanizm eksportu

Mechanizm eksportu został oparty na podstawie plików formatu XML. Pozyskanie danych można zrealizować zarówno po stronie bazy danych, jak i po stronie aplikacji. Ze

względu na prostotę i czytelność rozwiązania została zaimplementowana procedura zwracająca dokument XML.

Procedura `sp_geXMLTagData` zwraca dokument XML, zawierający rekordy wartości w postaci XML. Procedura przyjmuje parametry określające Id wartości oraz zakres dat.

### 3.5. Zarządzanie użytkownikami

Mechanizm zarządzania użytkownikami obejmuje podstawowe operacje, takie jak:

- Rejestrowanie użytkownika.
- Usuwanie użytkownika.
- Modyfikowanie danych użytkownika.
- Logowanie użytkownika.
- Pobranie autoryzowanych klas na podstawie profilu.

### 3.6. Funkcje Transact-SQL (Funkcje tabelaryczne)

Funkcje tabelaryczne zostały zastosowane głównie ze względu na możliwość łatwego manipulowania zwracanymi wynikami. Kontrolki Grid wymuszają konieczność zmiany kierunku sortowania oraz zakresu danych w zwracanych zestawieniach.

- Funkcja `tf_getAlarms` zwraca w wyniku dane dotyczące alarmów z podanego w parametrach okresu czasu.
- Funkcja `tf_getArchiveTags` zwraca w wyniku listę wszystkich tagów zdefiniowanych w systemie, dla których istnieją dane historyczne.
- Funkcja `tf_getTags` zwraca w wyniku zestawienie zawierające dane procesowe. Funkcja przyjmuje w parametrach identyfikator wartości oraz zakres dat.
- Funkcja `tf_getTags` zwraca w wyniku zestawienie zawierające dane procesowe. Funkcja przyjmuje w parametrach identyfikator wartości oraz zakres dat.
- Funkcja `tf_getUsers` zwraca zestawienie zawierające dane wszystkich użytkowników wykorzystywane do wypełniania kontrolek grid w module zarządzania użytkownikami.
- Funkcja `tf_CCTLG_GETVALUESBYID` jest używana do dekompresowania danych procesowych. Najważniejszymi elementami jej struktury są wywołania wykorzystujące rozszerzone procedury składowane `SP_CCTLG_*` oparte na bibliotekach `dll OLEDB PROVIDER`.

## 4. Aplikacja GWT

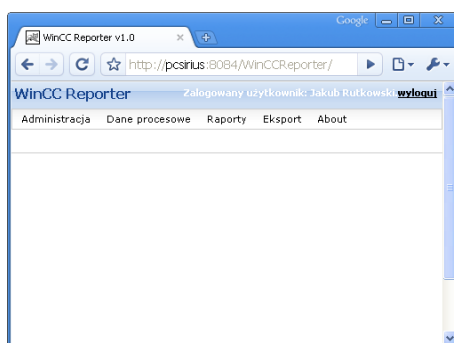
Aplikacja posiada wbudowany moduł zarządzający uwierzytelnianiem i autoryzacją. Uruchomienie aplikacji odbywa się przez wybranie odpowiedniego adresu w przeglądarce internetowej:

`http://localhost:8084/WinCCReporter/`

gdzie:

- Localhost – nazwa domenowa serwera aplikacji.
- 8084 – port nasłuchu serwera aplikacji.
- WinCCReporter – moduł aplikacji.

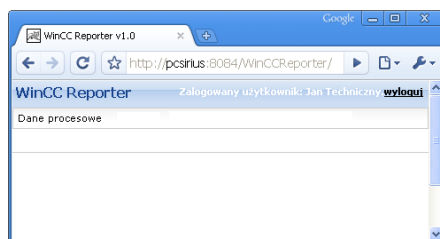
Po prawidłowym uwierzytelnieniu użytkownika odbywającym się przez podanie loginu i hasła, następuje sprawdzenie przywilejów użytkownika.



Rys. 10. Główne okno aplikacji po poprawnym uwierzytelnieniu

Fig. 10. Main application window after authorization

W przypadku gdy uwierzytelnianie zakończy się błędem, na formatce zostanie wyświetlony odpowiedni komunikat. Na rys. 11. przedstawiono ekran pracownika, który ma przydzielony profil „Pracownik produkcji”. Definicja tego profilu przewiduje jedynie udostępnienie przeglądu danych procesowych.



Rys. 11. Ekran aplikacji dla pracownika z profilem „Pracownik produkcji”

Fig. 11. Application window for employee with „Production employee” profile

Mechanizm autoryzacji został zaimplementowany jako oddzielny moduł GWT Auth. Po kliknięciu przycisku Login na formatce logowania zostaje wywołana metoda `loginUser` za pomocą zdalnego wywoływania procedur.

Definicja metody znajduje się w interfejsie

```
pl.sirsoft.gwt.winccreporter.auth.client.AuthService:  
UserInfo loginUser(LoginInfo user) throws AuthenticationException;
```

Implementacja metody znajduje się po stronie serwletu w klasie:

```
pl.sirsoft.gwt.winccreporter.auth.server.DefaultAuthServiceImpl.
```

Jeżeli uwierzytelnianie zakończy się poprawianie, to metoda loginUser zwraca zainicjowany obiekt UserInfo zawierający następujące pola:

```
private int usrId = 0;  
private String firstName = new String();  
private String lastName = new String();  
private String login = new String();  
private String[] authClasses.
```

W przeciwnym przypadku metoda podnosi wyjątek, co jest reprezentowane w postaci komunikatu błędu na formacie logowania.

Metoda onSuccess w wywołaniu RPC zawiera przepisanie autoryzowanych klas do HashMapy. Klasa pl.sirsoft.gwt.winccreporter.auth.client.AuthController zawiera metody sprawujące kontrolę nad autoryzacją. W momencie renderowania menu wywoływana jest metoda :

```
public boolean isObjectAuthorized(IAuth authorizingObject).
```

Metoda sprawdza, czy klasa jest zawarta HashMapie zwracając odpowiednią wartość logiczną. Definicja metody zawiera się w parametrze IAuth, którego implementowanie zapewnia udostępnienie klasie możliwości kontroli autoryzacji.

#### 4.1. Profile użytkowników

W systemie zostały zdefiniowane następujące profile użytkowników :

- Pracownik produkcji.
- Technik.
- Kierownik.
- Administrator.

Dostępne opcje i pozycje menu są zależne od przypisanego profilu użytkownika. Zależności zostały przedstawione w tab.7.

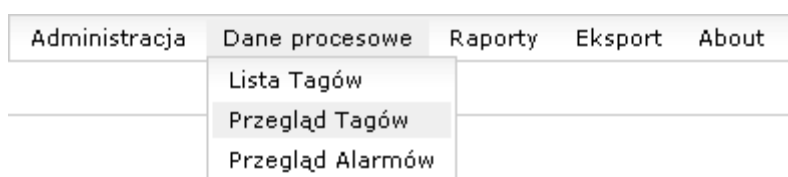
Tabela 7

## Uprawnienia profili

	ADMINISTRACJA	DANE PROCESOWE			RAPORTY	EKSPORT	ABOUT
	ZARZĄDZANIE UŻYTKOWNIKAMI	LISTA TAGÓW	PRZEGLĄD TAGÓW	PRZEGLĄD ALARMÓW	RAPORT WARTOŚCI ŚREDNICH	EKSPORT DO PLIKU	ABOUT
PRACOWIK PRODUKCJI		✓	✓	✓			
TECHNIK		✓	✓	✓		✓	
KIEROWNIK		✓	✓	✓	✓	✓	
ADMINISTRATOR	✓	✓	✓	✓	✓	✓	✓

#### 4.2. Warstwa prezentacji

Menu główne jest oparte na mechanizmie wbudowanym w szkielet GWT-PF. Klasa Menu zawarta w szkielecie charakteryzuje się dość rozbudowaną strukturą opartą na interfejsach. Mimo pozornego utrudnienia daje to niesamowitą swobodę w definiowaniu składników menu oraz zachowania się aplikacji przy wybraniu konkretnej pozycji. Pod pozycję menu można wstawić dowolną klasę, a jedynym wymaganym warunkiem jest implementowanie przez nią interfejsu `IActionProvider`. Sprowadza się to zatem do tego, aby klasa udostępniała metodę `String getActionCaption()` zwracającą wyświetlany w zakładce napis.



Rys. 12. Wygląd menu.

Fig. 12. Appearance of menu

#### 4.3. System zakładek

System zakładek pozwala na zwiększenie ergonomii korzystania z aplikacji. Mechanizm został zintegrowany z aplikacją dzięki zastosowaniu szkieletu GWT-PF. Wybranie pozycji menu powoduje otwarcie zawartości w nowej zakładce. Możliwe jest otwarcie wiele razy jednej pozycji menu ze względu na to, że wyświetlana zawartość jest tak naprawdę interfejsem użytkownika dla wybranej klasy. Zakładka reprezentuje zatem instancje klasy. Na rys. 13 przedstawiono wygląd systemu zakładek.





Rys. 13. Wygląd systemu zakładek

Fig. 13. Appearance of tab system

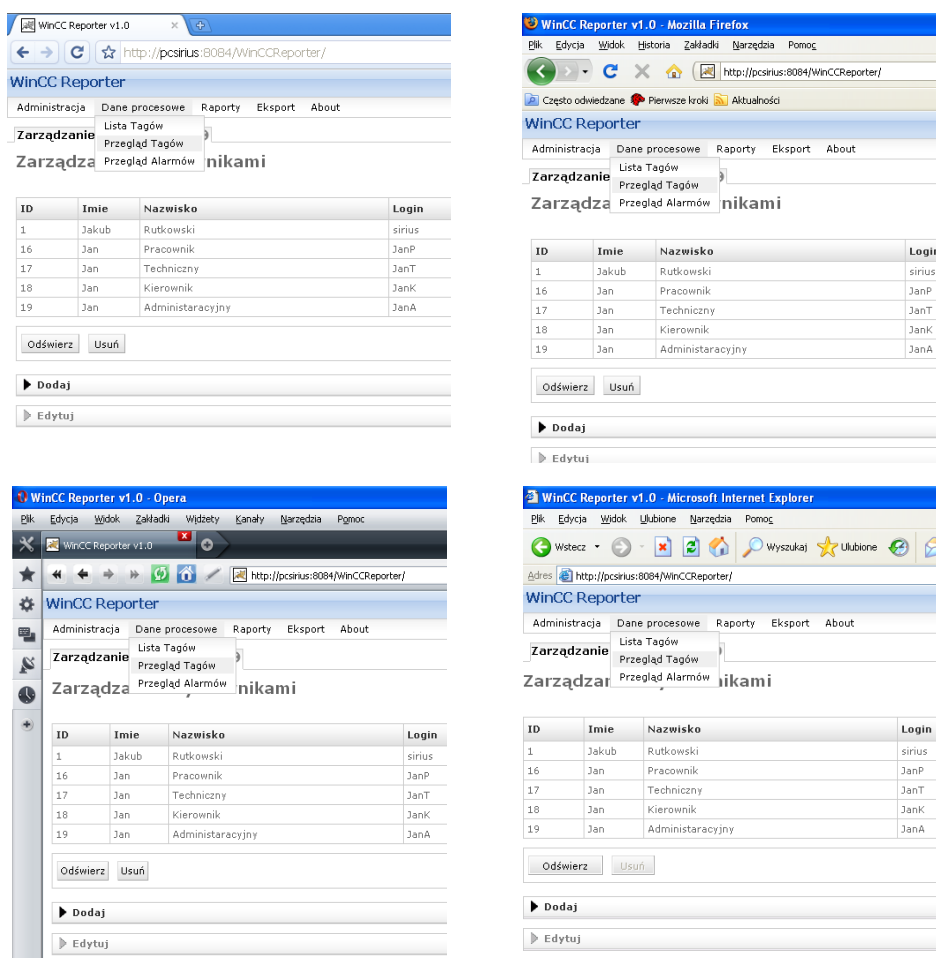
Każda zakładka zawiera opis oraz przycisk do zamknięcia (usunięcia) zakładki. Ustawienia użytkownika oraz wyświetlane dane pozostają na zakładce niezmiennie po ponownym jej otwarciu. Takie rozwiązanie umożliwia otwarcie kilku tego samego rodzaju formatek wypełnionych różnymi danymi oraz przełączanie widoku między nimi.

#### 4.4. Niezależność prezentacji od przeglądarki

Niezależność od platformy i przeglądarki jest wymieniana przez Google jako jedna z głównych zalet szkieletu GWT. Przeprowadzone testy wykazały, że aplikacja działa w różnych przeglądarkach identycznie. Biorąc pod uwagę, że językiem kodu wynikowego jest JavaScript, poziom zgodności jest zaskakująco wysoki. Ewentualne drobne różnice w wyglądzie są efektem nałożenia stylu kaskadowych CSS, których interpretacja może się różnić w zależności od przeglądarki. Znacznie trudniejszym zadaniem jest zachowanie zgodności dla elementów wykorzystujących kontrolki GWT-EXT, gdzie wygląd i działanie może się różnić w zależności od przeglądarki.

Testowane przeglądarki:

- Google Chrome 0.2.149.29 Beta.
- FireFox 3.
- Opera 9.51.
- Internet Expolorer 6.



Rys. 14. Wygląd kontrolek GWT w różnych przeglądarkach  
 Fig. 14. The appearance of the GWT controls in different web browsers

#### 4.5. Warstwa logiki biznesowej

Warstwa logiki biznesowej zawiera głównie elementy związane z frameworkiem GWT-PF. Filozofia pracy z tą biblioteką polega na definiowaniu logiki biznesowej, zachowań oraz charakterystyki obiektu, a na podstawie tych danych zostają automatycznie wygenerowane formatki. Aby dodać nowy element, należy napisać szereg klas implementujących odpowiednie interfejsy, dzięki którym zachowana jest spójność projektu. Interfejsy wymuszają implementację wymaganych metod, dzięki którym możliwa jest integracja nowego obiektu z istniejącym szkieletem.

#### 4.6. Warstwa dostępu do danych

Wszystkie serwlety zdefiniowane w aplikacji znajdują się w ścieżce pakietów `pl.sirsoft-gwt.winccreporter.server.*`. Większość zaimplementowanych metod dotyczy pobierania danych z bazy danych. Należy pamiętać o umieszczeniu odpowiedniego wpisu konfiguracji

serwletu w pliku web.xml. Fragment pliku web.xml odpowiedzialny za konfigurację serwletu AuthService wygląda następująco:

```
<servlet>
<servlet-name>
AuthService
</servlet-name>
<servlet-class>
pl.sirsoft.gwt.winccreporter.server.auth.WinccReporterAuthServiceImpl
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AuthService</servlet-name>
<url-pattern>
/pl.sirsoft.gwt.winccreporter.Main/auth
</url-pattern>
</servlet-mapping>
```

#### 4.7. Pobieranie danych

Poniżej przedstawiony jest fragment prostego serwletu odpowiedzialnego za implementację metod RPC dla przeglądu alarmów. Najważniejszą rolę pełnią metody select oraz selectCount odpowiedzialne za wypełnienie kontrolki grid. Należy zwrócić uwagę na sposób wywołania metod na obiekcie SqlMapClient związanym z biblioteką Apache iBatis.

Wywołanie:

```
getSqlMapClient().queryForList("selectAlarms", params);
```

powoduje zwrócenie wyników zdefiniowanych w SqlMap dla identyfikatora selectAlarms. Rezultat przypisany jest do listy, która jest zwracana w metodzie wcześniejszym zrzutowaniu na tablice obiektów AlarmOverview.

#### 4.8. Transformaty Apache FOP

Poniżej został przedstawiony prosty przykład zaimplementowania procesora Apache FOP w serwlecie. Dokument xml jest pobrany za pomocą biblioteki iBatis przez wywołanie metody queryForObject("getTagXMLDataFromDB"). Kolejnym krokiem jest zainicjowanie procesora FOP oraz przygotowanie plików biorących udział w transformacji. Wywołanie polecenia transformaty odbywa się przez wywołanie metody transform(src, res). Ostatecznie plik xml oraz pdf zostają zapisane na dysku.

#### 4.9. Konfiguracja i zastosowanie Apache iBatis

Podstawowa konfiguracja biblioteki Apache iBatis znajduje się w pliku SqlMapConfig.xml. Poniżej znajduje się konfiguracja połączenia dla obiektu transactionManager. Wartości parametrów \${host}, \${port}, \${database}, \${login} oraz \${password} są pobierane z pliku .properties. Aby konfiguracje sqlMap były aktywne, musi być dodany odpowiedni wpis

<sqlMap resource=""/>. W przykładzie znajduje się wpis dla mapy związanej z modulem uwierzytelniania i autoryzacji, którego konfiguracja znajduje się w pliku Auth.xml.

```
<transactionManager type="JDBC">
<dataSource type="SIMPLE">
<property name="JDBC.Driver"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
<property name="JDBC.ConnectionURL"
value="jdbc:sqlserver://${host}:${port};databaseName=${database}"/>
  <property name="JDBC.Username" value="${login}"/>
  <property name="JDBC.Password" value="${password}"/>
</dataSource>
</transactionManager>
<sqlMap resource="pl/sirsoft/gwt/winc creporter/server/auth/Auth.xml"/>
```

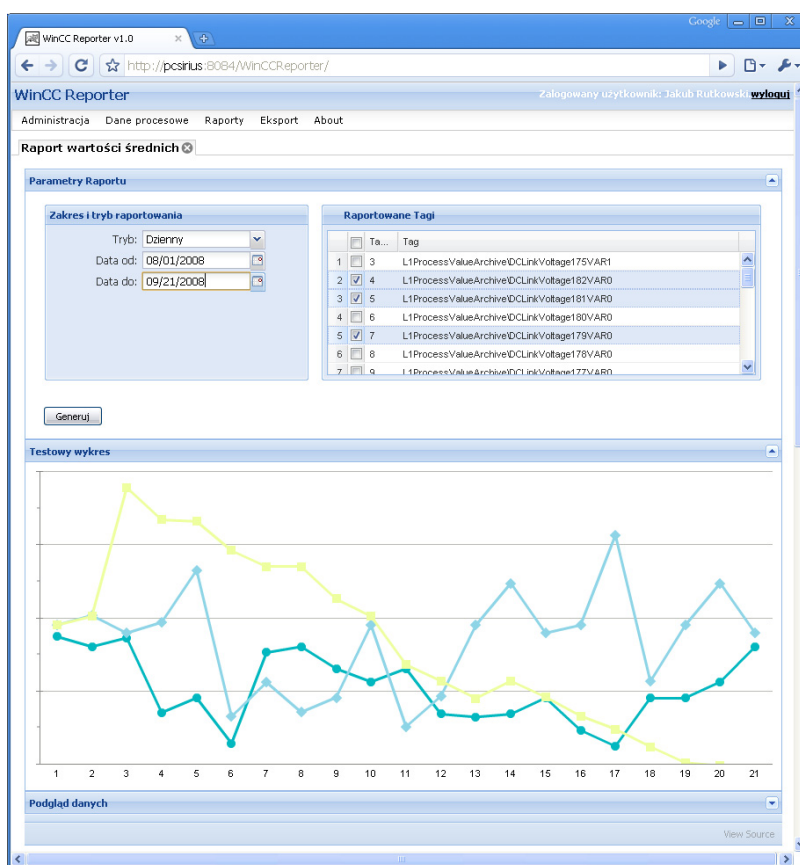
Definicje zawarte w sqlMap służą do mapowania wyników zapytań na obiekty Java. Przykład zawiera mapowanie dla klasy przeglądu alarmów. Element <typeAlias/> pozwala na zdefiniowanie aliasu, aby nie było potrzeby korzystania z pełnych ścieżek klas zawierających nazwę pakietu. Kolejnym krokiem jest zdefiniowanie <resultMap/>, dzięki któremu możliwe jest określenie mapowania pomiędzy nazwami kolumn zwracanych w zapytaniu a właściwościami docelowej klasy. Elementy <select/> pozwalają na zdefiniowanie nazwanego polecenia SQL z określeniem typów dla parametrów oraz rezultatu. Wartość argumentu id jest wykorzystywana w wywołaniu metod obiektu SqlMapClient.

#### 4.10. Moduł raportowania

Zbudowany moduł raportowania umożliwia przegląd oraz graficzną reprezentację danych zagregowanych według wybranych kryteriów. Ogólny wygląd formatki został przedstawiony na rys. 16. Implementacja przewiduje agregację danych w postaci średniej za okres wybrany za pomocą kontrolki „pola combo”. Lista rozwijana umożliwia wybranie trybu dziennej lub miesięcznej agregacji danych. Użytkownik ma możliwość zawężenia przedziału czasowego do wybranego okresu. Prawą stronę formatki wypełnia panel z dynamicznie pobieraną listą znaczników. Dzięki zaznaczeniu pól checkBox przy odpowiednim rekordzie możliwe jest wybranie raportowanych wartości. Wybranie przycisku „Generuj” spowoduje odwołanie się do serwletu w celu pobrania danych zagregowanych. Panel zawierający opcje raportowania zostanie zwinięty po odebraniu wyników.

Rys. 15. Rozwinięty panel zawierający dane sprzężone z wykresem  
Fig. 15. Data panel interconnected with graph

Obliczone dane można przeglądać za pomocą rozwijalnego panelu „Podgląd danych” przedstawionego na rys. 15. Wykres jest sprzężony z kontrolką grid, która zawarta jest na panelu, dlatego każda modyfikacja w komórkach zestawienia jest automatycznie ilustrowana na wykresie.

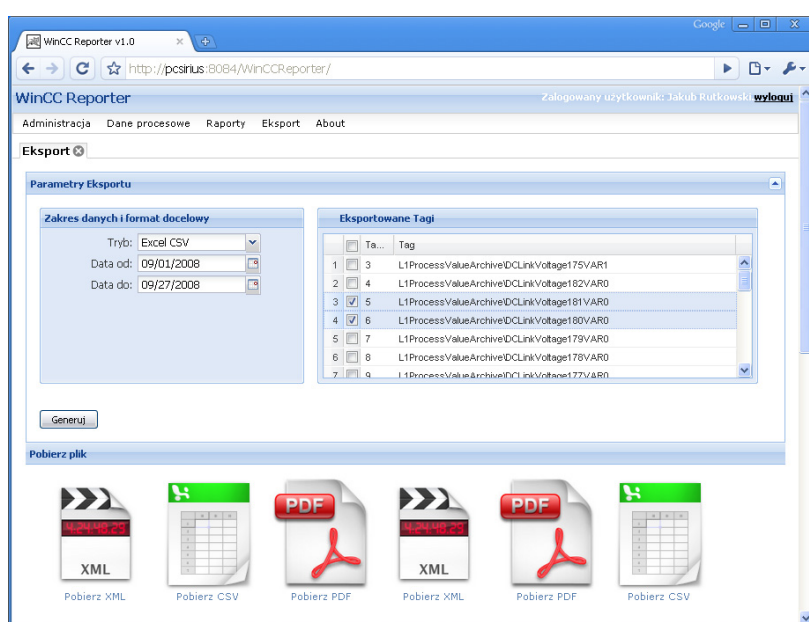


Rys. 16. Wygląd modułu raportowania  
Fig. 16. Appearance of reporting module

#### 4.11. Eksport danych

Moduł eksportu umożliwia zapisanie danych procesowych do postaci popularnych formatów plikowych. Użytkownik ma możliwość wyboru wyjściowego formatu pliku za pomocą listy rozwijanej. Dostępne opcje to CSV, XML, RTF oraz PDF. Dynamicznie

wypełniana kontrolka zestawienia pozwala wybrać eksportowane wartości za pomocą pól checkBox. Zakres dat może zostać ograniczony przez ustawienie przedziału za pomocą kontrolki daty. Wybranie przycisku generuj powoduje wysłanie zapytania do serwletu, gdzie zostaje uruchomiony procesor FOP. Zastosowana transformata XSL jest wybierana w zależności od wybranego formatu wyjściowego. Do przeglądarki jest przekazywany adres zapisanego pliku wyjściowego po jego wygenerowaniu przez FOP. Następnie na stronie zostaje umieszczony link z odpowiednią do typu ikoną. Rysunek 17. przedstawia ogólny wygląd formatki eksportu. Ponowne wybranie przycisku „Generuj” powoduje powtórzenie operacji, dzięki czemu możliwe jest wygenerowanie za pomocą jednej zakładki wielu plików eksportu bez utraty odnośników do poprzednich.



Rys. 17. Wygląd modułu eksportu

Fig. 17. Appearance of export module

## 5. Proces uruchomienia i instalacji aplikacji WinCCReporter

Do działania aplikacji niezbędne jest poprawnie skonfigurowane i uruchomione środowisko Siemens WinCC. W celu zarejestrowania i udostępnienia sterownika OLEDB należy dodatkowo zainstalować dodatek WinCC Connectivity Pack. Zainstalowanie platformy WinCC wymusza korzystanie z Microsoft SQL Server 2005, co wyklucza potrzebę osobnej instalacji wymaganego środowiska bazodanowego.

Kolejnym krokiem jest uruchomienie skryptu budującego bazę danych oraz rejestrującego login wskazujący jednoznacznie użytkownika. W tym celu należy zalogować

się do bazy danych, korzystając z użytkownika posiadającego uprawnienia wystarczające do utworzenia nowej bazy danych.

Wszelkie aplikacje korzystające z języka Java wymagają instalacji środowiska i wirtualnej maszyny tego języka. W tym celu należy uruchomić instalator JDK z katalogu załączników umieszczonego na dołączonej płycie. Po poprawnej instalacji należy sprawdzić i ewentualnie skorygować docelową destynację zmiennej środowiskowej `JAVA_HOME`, która powinna wskazywać na katalog `bin` w lokalizacji podanej w instalatorze.

Ostatnim krokiem jest zainstalowanie serwera aplikacji `jBoss`. W katalogu zawierającym załączniki, umieszczonym na dołączonej płycie znajduje się archiwum zawierające serwer aplikacji `jBoss`. Instalacja ogranicza się do rozpakowania archiwum `zip`. Nowe wersje serwera są dostępne do pobrania pod adresem <http://www.jboss.org/jbossas/downloads>. Serwer powinien być uruchomiony w domyślnym trybie `autodeploy`, dzięki czemu możliwe jest wdrożenie aplikacji przez skopiowanie archiwum `WinCCReporter.war` do katalogu `JBOSS_DIST/jboss-as/server/<instance-name>/deploy`.

## 6. Podsumowanie

Przedstawiona w pracy aplikacja internetowa pozwala na dostęp do archiwalnych danych procesowych bez konieczności ponoszenia kosztów z tytułu nabycia licencji dla terminali klienckich platformy `WinCC`. Zaprezentowane rozwiązanie pozwala na zdalną kontrolę i nadzór procesu produkcji, co w rezultacie może prowadzić do zmniejszenia awaryjności systemu i zwiększenia niezawodności produkcji. System dzięki wbudowanemu modułowi raportowemu może pełnić istotną rolę w procesie decyzyjnym w przedsiębiorstwie. Dane po wykonaniu agregacji pozwalają na prowadzenie statystyk i zestawień, co umożliwia sformułowanie wniosków dotyczących wydajności procesów technologicznych (np. po dokonaniu regulacji czy zmianie parametrów sterowania). Architektura systemu umożliwia łatwą dalszą rozbudowę o kolejne moduły. Funkcjonalność eksportu pozwala na prezentację lub dalsze przetwarzanie danych procesowych z wykorzystaniem dowolnych metod i zewnętrznych narzędzi. Możliwość eksportu danych do systemu `MS Excel` pozwala na ich dalszą analizę za pomocą aplikacji biurowej. W module eksportu wykorzystywany jest uniwersalny format dokumentów `XML`. Mechanizm tego typu umożliwia swobodną integrację z systemami zewnętrznymi i klientami usług `WebServices`.

Zastosowanie nowoczesnej technologii `Google Web Toolkit` [2] umożliwiło zbudowanie uniwersalnej i funkcjonalnej aplikacji, nieograniczonej przez platformę, system operacyjny czy przeglądarkę internetową.

Dość dużą trudność przy przygotowaniu aplikacji sprawiło opracowanie sposobu wydobycia i dekompresji archiwalnych danych procesowych z systemu Siemens WinCC. Głównym źródłem problemu jest brak ogólnodostępnej dokumentacji opisującej sposób wykorzystania bibliotek WinCC OLEDB Provider. Architektura aplikacji biznesowych powinna zakładać separację warstw logiki przetwarzania (logiki biznesowej) i prezentacji. W tym celu wykorzystuje się systemy sterowania przepływem pracy, np. w opierając się na środowisku Oracle Workflow. Autorzy system taki opisali w artykule [9].

Przy budowie aplikacji korzystano z rzeczywistych danych pochodzących z linii produkcyjnej, co umożliwiło testy aplikacji w rzeczywistym środowisku, z uwzględnieniem skali problemów związanych z ilością przetwarzanych danych. Testy te wykazały problemy wydajnościowe dekompresowania danych „w locie” przy ustawionych szerokich przedziałach czasowych i jednoczesnym przetwarzaniu wielu tagów. Rozwiązaniem tego problemu mogłoby być cykliczne przeliczanie i składowanie danych zagregowanych dla potrzeb raportowania, czyli zastosowanie hurtowni danych i rozwiązań typu OLAP.

## BIBLIOGRAFIA

1. Dokumentacja i materiały firmy Siemens <http://www.siemens.com/wincc>.
2. Cooper R., Collins Ch.: GWT in Practice. Manning, 2008.
3. Watt A.: .Microsoft SQL Server™ 2005 for Dummies. Wiley Publishing Inc, Indianapolis 2006.
4. Begin C., Goodin B.: iBatis in Action. Manning, 2007.
5. Kay M.: XSLT Programmer's Reference. Wrox Press, Birmingham 2005.
6. Tickwell D.: XSLT Mastering XML Transformations. O'Reilly, 2006.
7. Specyfikacja XSLT 2.0 <http://www.w3.org/TR/xslt20/>.
8. Microsoft Developer Network <http://msdn.microsoft.com>.
9. Drzymała P., Welfle H., Wiak S.: System sterowania przepływem pracy w oparciu o środowisko Oracle Workflow. „Bazy danych. Rozwój metod i technologii. Bezpieczeństwo, wybrane technologie i zastosowania” pod redakcją: S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, WKŁ, Warszawa 2008, s. 27÷48.

Recenzent: Dr inż. Dariusz R. Augustyn

Wpłynęło do Redakcji 31 stycznia 2009 r.



## Abstract

The paper presents the possibility of creating of network application to preview and the analysis of data process, accumulated by system of SCADA class (Supervisory Control And Data Acquisition). The Siemens – WinCC system was used as accumulating data platform. The main functions of presented system is the access to the current data (the measurements), their analysis, visualization, the monitoring of the productive alarms as well as the filing of accumulated data. The paper presents the example of practical use of modern technologies integration to built the effective, industrial solution.

The subject of this work includes problem of integration of Simatic WinCC SCADA system with internet application which using the newest technologies. Microsoft SQL Server 2005 will choose as database environment because WinCC platform already use them. The application is built with the Google Web Toolkit technology use. The practical part requires building the Internet application which will let extract, present and transform compressed industry data included in the Siemens WinCC system. Finding the way of getting and decompressing data is an important element of the project. Using OLEDB interface the system invokes stored procedures returning data in desired format. Available options in the system will be dependent on the profile of the logged user. This solution will provide managing possibility depending on worker business function. Designing the right data structure will be essential in order to carry out functionality requirements of the application.

## Adres

Paweł Drzymała: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, pdrzymal@p.lodz.pl.

Henryk Welfle: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, welfle@p.lodz.pl.

Sławomir Wiak: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, wiakslaw@p.lodz.pl.

Jakub Rutkowski: Politechnika Łódzka, Instytut Mechatroniki i Systemów Informatycznych, ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, jrutkowski@o2.pl.