

# Transfer ekranów synoptycznych pakietu SCADA na urządzenia przenośne

Dorian Chrzanowski\*, Krzysztof Kołek\*\*, Jakub Sotwin\*\*\*

\*Nokia Siemens Networks

\*\*AGH Akademia Górniczo-Hutnicza, Wydział EAIiIB, Katedra Automatyki i Inżynierii Biomedycznej

\*\*\*Motorola Solutions Systems Polska

**Streszczenie:** W pracy zaprezentowano metodę umożliwiającą automatyczny transfer ekranów synoptycznych systemu SCADA do urządzeń pracujących pod kontrolą systemu operacyjnego Android. Omówiono metody transferu danych czasu rzeczywistego z oprogramowania SCADA do docelowej platformy programowej oraz metodę transferu danych z serwerów OPC DA. Opiszano zasadę przeniesienia informacji o obiektach i ich właściwościach, a w szczególności parametrach dotyczących animacji, z pakietu Proficy iFIX do oprogramowania w systemie Android realizowanego w języku Java. Przedstawiono przykład ekranu synoptycznego przeniesionego ze środowiska komputera PC do telefonu komórkowego.

**Słowa kluczowe:** SCADA, ekran synoptyczny, Android, urządzenia mobilne

## 1. Wprowadzenie

System operacyjny Android, dedykowany urządzeniom przenośnym, notuje ponad 250 % rocznego przyrostu aktywnych urządzeń. Szacuje się, iż pracuje już 300 milionów urządzeń, dziennie aktywowanych jest 850 000 nowych, a liczba dostępnych, w znacznym stopniu bezpłatnie, aplikacji wynosi około 450 000 [1]. Wśród takiego bogactwa sprzętu oraz oprogramowania w stosunkowo niewielkim stopniu poruszana jest tematyka możliwości zastosowania urządzeń z systemem Android w systemach sterowania.

Istniejących rozwiązań w zakresie monitorowania procesów urządzenia mobilne są stosowane w roli przeglądarki internetowej lub wymagają określonego typu urządzenia docelowego. Pierwsze rozwiązanie jest wykorzystywane w pakiecie iFIX WebSpace. Przykładem drugiego podejścia jest aplikacja ProficySCADA dla tabletu iPad [2].

Warto również wspomnieć o próbach zastosowania urządzeń z systemem Android do bezpośredniego sterowania. W pracy [3] przedstawiono aplikację, w której telefon komórkowy steruje w czasie rzeczywistym laboratoryjnym modelem testującym algorytm hamowania w trybie zapobiegającym zakleszczeniu (ABS Antilock Breaking System). Interesującym może okazać się opracowanie metody zapewniającej automatyczny transfer ekranów synoptycznych pakietu SCADA na dowolne urządze-

nie pracujące pod kontrolą systemu operacyjnego Android, w sposób umożliwiający ich lokalne przetwarzanie.

Automatyczny transfer ekranów synoptycznych wymaga rozwiązania następujących zagadnień:

- opracowania metody transferu danych czasu rzeczywistego z pakietu SCADA do urządzenia z systemem Android,
- automatycznego transferu obiektów graficznych wraz z ich właściwościami z ekranów synoptycznych SCADA do urządzenia mobilnego.

Jako pakiet SCADA wybrano Proficy HMI/SCADA iFIX. W środowisku tym tworzone wyjściowe ekrany synoptyczne. Również pakiet iFIX był źródłem danych czasu rzeczywistego dla animowanych, na docelowym urządzeniu mobilnym ekranów synoptycznych.

## 2. Transfer danych czasu rzeczywistego

Jako przykładowe rozwiązania, ilustrujące zagadnienie transferu danych czasu rzeczywistego, zaimplementowano dwa scenariusze. Pierwszy opiera się na wykorzystaniu połączenia z zastosowaniem komunikacji z protokołem TCP/IP. Rozwiązanie drugie zostało przystosowane do pracy zgodnej ze standardem OPC.

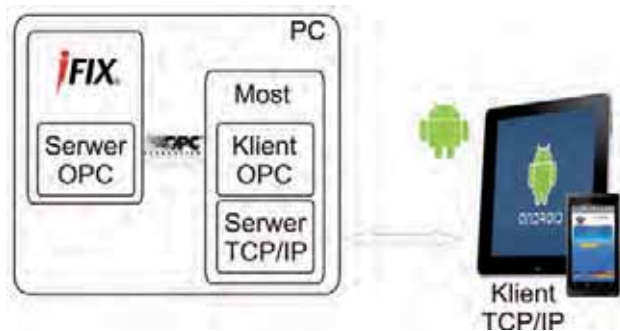
Protokół TCP/IP w opisywanym rozwiązaniu realizuje wymianę danych w architekturze klient-serwer. Zgodnie z tym podziałem można klarownie wyróżnić dwie odrębne i niezależne od siebie części projektu. Rolę serwera TCP/IP pełni może pakiet SCADA, zaś klienta – urządzenie z systemem Android (rys. 1).



Rys. 1. Konfiguracja z dedykowanym protokołem  
Fig. 1. Dedicated protocol configuration

W celu wymiany informacji opracowano dedykowane skrypty komunikacyjne. Po stronie serwera są realizowane skrypty typu „picture script”, przygotowane w języku Visual Basic for Application. Zastosowano tu kontrolkę ActiveX realizującą funkcje silnika TCP/IP, pracującą w roli serwera. Realizacja klienta TCP/IP działającego pod Androidem możliwa jest przez wykorzystanie wbudowanych klas języka Java [4].

Drugi prezentowany sposób komunikacji bazuje na standardzie OPC DA. W kontekście urządzeń automatyki jest to rozwiązanie zdecydowanie najbardziej uniwersalne. Jego zastosowanie umożliwia komunikację nie tylko z pakietami SCADA, ale również z większością obecnie dostępnych urządzeń automatyki, typowo oferujących funkcjonalność serwera OPC DA. Niestety, system Android nie ma oprogramowania wspierającego standard OPC. Niedogodność ta została rozwiązana dzięki opracowaniu programowego mostu między serwerem OPC i komunikacją realizowaną za pomocą protokołu TCP/IP (rys. 2).



**Rys. 2.** Przepływ danych w konfiguracji z mostem OPC / TCP/IP  
**Fig. 2.** OPC / TCP/IP bridge data flow

Z racji braku możliwości uruchomienia bezpośrednio na telefonie z systemem Android natywnego klienta OPC DA (tj. klienta dedykowanego dla tego systemu), umożliwiającego programowy dostęp do danych czasu rzeczywistego, komunikacja między urządzeniem mobilnym a komputerem osobistym odbywa się za pomocą protokołu TCP/IP. Ponieważ zarówno klient, jak i serwer napisane są w języku Java i całość procesu komunikacji odbywa się z wykorzystaniem wbudowanych klas, do obsługi protokołu TCP/IP nie są potrzebne zewnętrzne biblioteki. Rolą mostu OPC↔TCP/IP jest tłumaczenie zapytań otrzymywanych w postaci ramek TCP/IP na inne zapytania, które tym razem przekazywane są do serwera OPC. Most pracuje jednocześnie w roli serwera TCP/IP oraz klienta OPC, tunelując ruch między urządzeniem mobilnym i serwerem OPC.

Warto wspomnieć o kolejnej możliwości transferu danych między pakietami SCADA i programami systemu Android. Podejście wykorzystuje specyfikację OPC Unified Architecture (OPC UA), w której binarny format komunikatów specyfikacji OPC DA został zastąpiony tekstowym transferem realizowanym za pomocą protokołu SOAP. Specyfikacja OPC UA jest stosunkowo nowa i większość pakietów SCADA nie realizuje funkcji serwerów OPC UA. W tym przypadku transfer danych wymaga opracowania programowego mostu SCADA↔OPC UA. Most może pracować analogicznie do konfiguracji przedstawionej na rys. 2, gdy zamiast serwera TCP/IP zastosowany zostanie serwer OPC UA, a po stronie

systemu Android uruchomione zostanie odpowiednie oprogramowanie klienta. Zaletą tego podejścia jest wykorzystywanie wyłącznie standaryzowanych elementów. Istotnymi wadami, które przesądziły o zaniechaniu tej ścieżki, są: trudności implementacyjne serwera oraz klienta OPC UA oraz zwiększenie transferowanych danych, związane z przesyłaniem tekstowych komunikatów XML za pomocą protokołu SOAP. Zwiększenie transferu danych może okazać się czynnikiem szczególnie istotnym w przypadku urządzeń mobilnych zasilanych bateryjnie, ponieważ znacząco rzutuje na czas pracy.

### 3. Transfer obiektów graficznych

Metoda przenoszenia ekranów synoptycznych z systemu SCADA do urządzeń mobilnych jest przedstawiona na rys. 3. Proces składa się z dwóch etapów:

- pobranie danych dotyczących ekranu synoptycznego, zapisanie ich do pliku konfiguracyjnego i odtworzeniu ekranu na urządzeniu z systemem operacyjnym Android,
- animowanie obiektów odtworzonego ekranu zgodnie z danymi czasu rzeczywistego pobieranymi na bieżąco z systemu SCADA.



**Rys. 3.** Metoda transferu ekranów synoptycznych  
**Fig. 3.** SCADA screen transfer method

Po pierwsze, niezbędne jest pobranie danych dotyczących wszystkich wyświetlanych obiektów ekranu synoptycznego (właściwości obiektów). Obiekty ekranów synoptycznych niektórych pakietów SCADA, w tym wykorzystywanego pakietu iFIX, tworzą hierarchię drzewa – korzeniem ekranu synoptycznego jest obiekt o nazwie „Picture”, który zawiera właściwości ogólne ekranu synoptycznego (np. kolor tła, wymiary ekranu). Gałęziami dla tego obiektu są wszystkie inne wyświetlane elementy ekranu synoptycznego (np. teksty, figury geometryczne, przyciski, kontrolki, wykresy). Każdy z elementów zawiera pewien zbiór parametrów, które definiują sposób wyświetlania obiektów (rozmiar, kolor, spo-

sób wypełnienia). Dodatkowo, jeśli obiekt jest animowany, w hierarchii drzewiastej obiekt ma także „dzieci”. Obiekty te, nazywane dalej animacjami, zawierają strukturę danych opisującą zachowanie się animowanych obiektów w zależności od wartości zmiennej animującej obiekt. Przykładowe animacje to: zmiana koloru obiektu, przemieszczanie się poziome/pionowe, skalowanie obiektu wzdłuż/wszereż, wypełnianie obiektu kolorem oraz tekst reprezentujący wartość zmiennej. Aby możliwe było odtworzenie ekranu synoptycznego na urządzeniu z systemem operacyjnym Android, niezbędne jest zebranie pewnego podzbioru właściwości wszystkich wymienionych obiektów. W tym celu napisano skrypt w języku Visual Basic for Applications, wykorzystując fakt, że obiekty ekranu synoptycznego tworzą strukturę drzewa. Począwszy od „korzenia”, skrypt przegląda właściwości obiektu i zapisuje je do pliku. Następnie sprawdzane jest, czy dany obiekt ma „dzieci”. Jeśli tak, procedura pobierania niezbędnych właściwości jest w sposób rekurencyjny powtarzana dla każdego z „dzieci”.

Wszystkie pobrane dane zapisywane są do pliku ze znacznikami XML. Wybór takiego formatu pliku podyktowany był dwoma powodami [5]:

- Hierarchię drzewa można opisać w pliku XML (format XML sam zachowuje hierarchię drzewa).
- Istnieje wiele parserów plików XML dla języka Java, co znacznie ułatwia czytanie ich na urządzeniu z systemem Android. Parsery realizują analizę znakowych plików XML skutkującą uzyskaniem struktur z danymi opisanymi w tych plikach. Zagadnienie to jest istotne, gdyż ekrany synoptyczne mogą zawierać wiele elementów na jednym ekranie synoptycznym, każde opisane przez kilkadziesiąt właściwości.

Plik XML, ze wszystkimi obiektami ekranu synoptycznego oraz ich właściwościami, zostaje przesłany do urządzenia z systemem operacyjnym Android. Warto podkreślić, iż transfer ten wykonywany jest jednorazowo. Przy pomocy aplikacji napisanej w języku Java odtwarzamy ekran synoptyczny na ekranie urządzenia mobilnego. Aby było to możliwe, niezbędne jest odczytanie wszystkich danych zapisanych w pliku konfiguracyjnym ze znacznikami XML. W tym celu użyto parsera „DOM Parser” [6]. Nazwę pliku konfiguracyjnego oraz inne ustawienia (częstotliwość odświeżania danych, rodzaj komunikacji między pakietem SCADA a urządzeniem mobilnym) definiuje użytkownik z poziomu aplikacji. Na podstawie właściwości obiektów zapisanych w pliku konfiguracyjnym, na urządzeniu mobilnym odtwarzany jest graficzny odpowiednik ekranu synoptycznego pakietu SCADA.

Proces tworzenia pliku konfiguracyjnego oraz jego parsowanie następuje jednokrotnie i deklaruje wszystkie obiekty ekranu synoptycznego. Jednakże, w celu przedstawienia zmian stanu obiektów w pakiecie SCADA, w aplikacji uruchamiany jest wątek odpowiedzialny za odświeżanie danych animujących wybrane obiekty. W wątku tym, z zadaną przez użytkownika częstotliwością, uruchamiana jest procedura pobrania z pakietu SCADA aktualnych wartości zmiennych niezbędnych do animowania obiektów. Dostęp do danych pakietu SCADA realizowany jest za pomocą metod opisanych w punkcie 3. Nazwy zmiennych potrzebnych do animowania obiektów są zapisane w pliku konfiguracyjnym. Po odświeżeniu wartości animujących, aplikacja generuje odpowiednik ekranu synoptycznego na ekranie urządzenia mobilnego.

## 4. Przykład

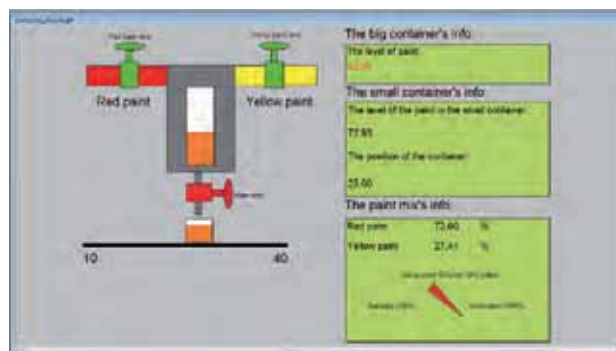
Na rys. 4 przedstawiono ekran utworzony w środowisku Proficy iFIX (rys. 4 a) oraz odpowiadający mu ekran utworzony na telefonie komórkowym (rys. 4 b).

Na podstawie zawartości ekranu synoptycznego SCADA, za pomocą skryptów VBA, utworzono plik XML. Plik ten jest podstawą pracy aplikacji Java tworzącej graficzny ekran w telefonie komórkowym. Do transferu danych wykorzystano opisany most OPC↔TCP/IP.

Na obu ekranach widoczne jest dobre, choć nie identyczne, odwzorowanie szczegółów. Mogą być dwa źródła takiej niejednoznaczności. Pierwszym jest przeskalowanie ekranu związane ze zmianą rozdzielczości. Drugą może być brak dostępu (producent nie udostępnia tych danych) do algorytmów rysowania poszczególnych obiektów graficznych pakietu SCADA. Dostępne są informacje o wartościach właściwości, ale ich interpretacji trzeba się wielokrotnie domyślać.

Warto w tym kontekście również wspomnieć o niekompletności opracowanych algorytmów przenoszenia i odtwarzania ekranów. W wyjściowym środowisku deweloperskim występują dziesiątki typów obiektów, każdy z dziesiątkami parametrów. Wierne odtworzenie każdej kombinacji byłoby pracą iście syzyfową. W opisywanym oprogramowaniu skupiono się na modularności – zaimplementowano stosowane obiekty oraz ich właściwości, pozostawiając miejsce na implementację pozostałych.

a)



b)



Rys. 4. Przykładowy ekran synoptyczny: a) na komputerze PC b) na telefonie komórkowym

Fig. 4. Example SCADA screen a) at PC screen b) at mobile phone display

## 5. Wnioski

Przedstawione podejście wykorzystuje środowisko deweloperskie pakietu SCADA do tworzenia ekranów synoptycznych, dla których docelową platformą nie jest komputer PC, lecz mobilne urządzenie pracujące pod kontrolą systemu Android. Utworzony ekran może pracować również w środowisku PC, a ekran na urządzeniu mobilnym będzie jego graficzną kopią. Jedynymi operacjami wymaganymi do przeniesienia ekranu system SCADA do środowiska Android są: uruchomienie skryptu VBA generującego plik XML opisujący obiekty ekranu oraz przeniesienie tego pliku na urządzenie docelowe. Opisane oprogramowanie dla urządzenia docelowego wygeneruje ekwiwalenty do systemu SCADA ekran oraz przeprowadzi animację bazującą na odczytywanych z serwera OPC DA danych.

Przedstawione rozwiązanie ma dodatkowo dwie zalety wynikające z otwartego formatu opisu danych w postaci plików XML.

Ponieważ struktura graficznego ekranu na urządzeniu mobilnym jest całkowicie określona w pliku XML, użytkownik urządzenia może samodzielnie modyfikować wygląd graficzny edytując plik XML. W szczególności możliwe jest zaprojektowanie własnych ekranów synoptycznych nieposiadających swojego rodowodu w środowisku deweloperskim SCADA. Metoda taka nie jest obca programistom pracującym w środowisku Android – podobną strukturę ma jeden ze sposobów tworzenia interfejsu GUI aplikacji.

Również źródło danych dla animowanych ekranów może zostać zmienione. Zmieniając wpisy pliku XML oraz wykorzystując przedstawiony most OPC↔TCP/IP można uzyskać dane czasu rzeczywistego z dowolnego serwera OPC.

Wydaje się, iż zaprezentowane podejście, przynajmniej w kontekście automatyki, może z urządzeń z systemem Android uczynić urządzenia prawdziwie przenośne i użyteczne.

## Bibliografia

1. <http://thenextweb.com/mwc/2012/02/27/android-growing-250-year-on-year-with-over-300-million-total-devices-worldwide/>.
2. <https://itunes.apple.com/app/proficyscada/id525792142?mt=8&ls=1>.
3. Kołek K., *Zastosowanie systemu Android jako platformy czasu rzeczywistego*, materiały konferencyjne, Systemy Czasu Rzeczywistego, 10–12 września 2012, Kraków.
4. Ableson W.F., Sn R., King C., *Android w akcji*, Helion, Gliwice, 2011.
5. W3school. *XML Tutorial*. [Online] <http://www.w3schools.com/xml/>
6. How to read XML file in Java – (DOM Parser). [Online] <http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>. ■

## Transfer of SCADA screens to mobile devices

**Abstract:** The paper presents the method that can be applied to automatic transfer of SCADA screens to mobile devices equipped with Android operating system. The real-time data transfer methods to Android devices are given. In particular the data transfer method from OPC DA servers is shown. It is described the data flow that collects all properties, including animation setups, of the SCADA screens and creates graphically equivalent screen at a device running Android OS Java application. The example of a SCADA screen ported to the mobile phone is given.

**Keywords:** SCADA, SCADA screen, Android, mobile device

### mgr inż. Dorian Chrzanowski

Absolwent Wydziału Elektrotechniki, Automatyki, Informatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie w 2012 r. Pracę magisterską zrealizował na stypendium naukowym w Antwerpii, Belgia, we współpracy z uczelnią Karel de Grote-Hogeschool. Od września 2012 r. pracownik Nokia Siemens Networks w Krakowie. Zainteresowany systemami SCADA, systemami zarządzania i przetwarzania danych.

e-mail: [chdorian@gmail.com](mailto:chdorian@gmail.com)



### dr inż. Krzysztof Kołek

Absolwent Wydziału Elektrotechniki, Automatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Od 1989 r. zatrudniony w Katedrze Automatyki. W 1996 r. otrzymał tytuł doktora nauk technicznych. Specjalizuje się w zagadnieniach sterowania w czasie rzeczywistym oraz w zastosowaniu rekonfigurowalnych układów cyfrowych.

e-mail: [kko@agh.edu.pl](mailto:kko@agh.edu.pl)



### mgr inż. Jakub Sotwin

Ukończył studia na kierunku Automatyka i Robotyka na Wydziale Elektrotechniki, Automatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Studiował także na Uniwersytecie Papieskim Jana Pawła II w Krakowie oraz Karel de Grote Hogeschool w Antwerpii. Obecnie zatrudniony w Motorola Solutions Systems w Krakowie. Interesuje się aplikacjami mobilnymi oraz rozwiązaniami radiokomunikacyjnymi.

e-mail: [j.sotwin@gmail.com](mailto:j.sotwin@gmail.com)

