

Piotr Augustyniak

Oprogramowanie autoadaptacyjne w aplikacjach wbudowanych

Streszczenie

Aplikacje wbudowane zwykle kojarzone są z dedykowanymi urządzeniami (telefon, pilot zdalnego sterowania, nawigator GPS, kamera video, odtwarzacz plików) pozbawionymi uniwersalności typowej dla komputera. Tymczasem atrakcyjna i coraz częściej oczekiwana przez użytkowników jest personalizacja tych urządzeń zarówno w zakresie drobnych ustawień, jak i zasadniczej zmiany ich funkcjonalności. Zadanie to można zrealizować stosunkowo łatwo za pomocą oprogramowania adaptacyjnego. Prezentowany artykuł stanowi przegląd metod adaptacji oprogramowania w aplikacjach wbudowanych. Szczególną uwagę zwrócono na metody automatycznego zarządzania kodem wykorzystujące analizę funkcjonalności urządzenia.

Wstęp

Dzięki szybkiemu postępowi współczesnej elektroniki miniaturyzacja systemów mikroprocesorowych, a nawet całych mikrokomputerów pozwala na stosowanie ich w urządzeniach powszechnego użytku. Oprogramowanie tych komputerów, w przeciwieństwie do komputerów ogólnego przeznaczenia, jest ściśle dostosowane do docelowej architektury sprzętowej, a ponieważ przewidywane jest wyłącznie uruchamianie programów producenta, stosowanie systemu operacyjnego nie jest konieczne. Aplikacje wbudowane są zwykle projektowane jako jednokrotnie programowalne, choć obecnie większość producentów udostępnia uaktualnienia (ang. update) oprogramowania systemowego (ang. firmware). Uzyskują oni możliwość szybkiego prototypowania urządzeń, których funkcjonalność oparta jest na oprogramowaniu, a następnie poprawiania usterek u użytkownika. Urządzenia przechowują oprogramowanie systemowe (przez analogię do płyt głównych komputerów zwane niekiedy bios'em) w pamięci wielokrotnie programowalnej typu flash, a zatem możliwa jest nie tylko nadanie określonej funkcjonalności urządzenia podczas jego produkcji, ale także modyfikacja jej u klienta, często dokonywana przez niego samodzielnie przy wykorzystaniu Internetu. Uaktualnianie oprogramowania jest możliwe w wielu urządzeniach powszechnego użytku: cyfrowych kamerach i aparatach fotograficznych, telefonach, nawigatorach GPS, pilotach zdalnego sterowania, odtwarzaczach muzyki i filmów.

Rozszerzeniem tej koncepcji jest modyfikacja oprogramowania dokonywana zdalnie w sposób automatyczny na podstawie analizy potrzeb użytkownika. Stwarza ona interesującą marketingowo możliwość masowej,

a więc taniej produkcji urządzeń przenośnych ogólnego przeznaczenia (np. telefono-kamero-piloto-odtwarzaczy), a następnie definiowania ich wyglądu i przeznaczenia za pomocą oprogramowania stosowanie do potrzeb i upodobań konkretnego użytkownika. Rozważaniom na temat technicznej możliwości automatycznej personalizacji aplikacji wbudowanych poświęcony jest niniejszy artykuł. Bez utraty ogólności wniosków, temat zostanie przedstawiony na przykładzie przenośnego aparatu EKG [1-3] pobierającego sygnał z powierzchni ciała za pomocą elektrod, wykonującego automatyczną analizę sygnału i transmitującego rezultaty diagnostyczne cyfrowym łączem bezprzewodowym GPRS. To samo łącze zostało wykorzystane do kontroli poprawności działania aparatu oraz zdalnego uaktualniania jego oprogramowania [4-5].

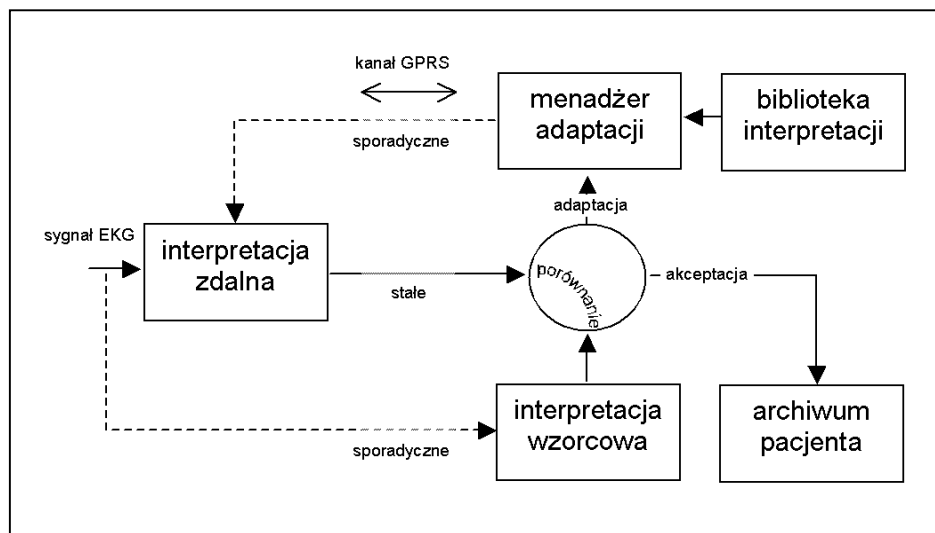
1. Uwarunkowania automatycznej adaptacji oprogramowania

A. Optymalizacja w pętli sprzężenia zwrotnego

Zasadniczym celem automatycznej modyfikacji oprogramowania wbudowanego jest optymalizacja jego funkcjonalności w określonych warunkach. Stosując klasyczne pojęcie optymalizacji należy zatem odpowiednio wybrać reprezentację błędu dostosowania urządzenia do potrzeb w postaci zestawu mierzalnych parametrów. Funkcja błędu zostanie następnie użyta jako argument wejściowy procedury zarządzania oprogramowaniem klienta i na jej podstawie zostanie wygenerowana decyzja o uaktualnieniu. Po uaktualnieniu konieczne jest sprawdzenie, czy zmodyfikowane oprogramowanie urządzenia przenośnego lepiej spełnia oczekiwania klienta, co jest reprezentowane przez zmniejszenie wartości funkcji błędu. Znaki wartości i pochodnej funkcji błędu są więc reprezentacją poprawności podejmowanych decyzji i analogicznie do typowego sprzężenia zwrotnego w wyniku modyfikacji oprogramowania zdalnego oczekiwana jest zbieżność wartości funkcji błędu do zera (rys. 1).

B. Ograniczenia sprzętowe urządzeń przenośnych

Tylko nieliczne urządzenia przenośne mogą być konstruowane jako miniaturowe kopie ich pełnowymiarowych odpowiedników stacjonarnych. Głównym ograniczeniem jest kompromis masy i czasu autonomicznej pracy.



Rysunek 1. Schemat blokowy automatycznej zdalnej adaptacji oprogramowania urządzenia przenośnego na przykładzie elektrokardiografu telemetrycznego

Pomimo znacznego postępu w zakresie regenerowalnych źródeł zasilania, z punktu widzenia użytkownika czas autonomicznej pracy będzie zawsze zbyt krótki, a urządzenie przenośne - zbyt ciężkie. Rozwiązania programowe w systemach przenośnych są zwykle kopiami podobnych rozwiązań stacjonarnych, pomimo znacznego przewartościowania wymagań i ograniczeń. Zwykle sprowadza się to do wykonywania skomplikowanych algorytmów wymagających znacznej energii, których rezultaty w większości są mało interesujące lub nawet istotne dla użytkownika. Proponowane rozwiązanie polega na wydłużeniu czasu pracy poprzez uproszczenie lub całkowitą eliminację zbędnych obliczeń wykonywanych w systemie wbudowanym. Istotne zagadnienie stanowi właściwa identyfikacja funkcjonalności urządzenia, które z punktu widzenia potrzeb użytkownika powinny być obsługiwane z większą lub mniejszą dokładnością, albo pominięte.

C. Zależność jakości i wymagań sprzętowych procedur

W cyklu rozwojowym oprogramowania, zwłaszcza procedur wykorzystujących dane heurystyczne, np. automatycznej interpretacji elektrokardiogramu, wczesne projekty prowadzą do uzyskania algorytmów prostych, o małych wymaganiach sprzętowych ale też niezbyt dokładnych. Dopiero w następnych etapach cyklu, gdy mankamenty prostych procedur są znane, lub gdy wymagania dokładności zostały podwyższone (zaostrenie norm lub działalność konkurencji rynkowej), zostają zaprojektowane algorytmy bardziej wyrafinowane, których rezultaty są bardziej wiarygodne.

Niestety, zwykle charakteryzują się one także wyższą złożonością obliczeniową i zwiększonym zapotrzebowaniem na zasoby. Wśród producentów oprogramowania prowadzących własne prace rozwojowe prawidłowością jest jednak posiadanie kilku wersji ważniejszych procedur, których zbiór można uporządkować w monotoniczną zależność jakości i wymagań. Zbiór taki często jest wykorzystywany do oprogramowania aplikacji wbudowanych produktów w różnych wersjach cenowych i funkcjonalnych, co wzbogaca ofertę rynkową. Umożliwia także wzbogacenie funkcjonalności w późniejszym terminie przez proste uaktualnienie oprogramowania, w przypadku gdy po początkowym okresie użytkowania okaże się ona niewystarczająca.

D. Zabezpieczenia pamięci aplikacji

Producenci urządzeń przenośnych zakładają uruchamianie aplikacji wbudowanych własnej produkcji i umożliwiają uaktualnianie pamięci programu zrealizowanej w postaci stałej, kasowanej i programowanej elektrycznie. W celu ochrony własności intelektualnej pamięć ta rzadko umożliwia odczyt zawartości przez urządzenie zewnętrzne, regułą jest także istotne ograniczanie ilości możliwych cykli zapisu (w testowanych telefonach i komputerach PDA spotykano wartości 200, 500). Celem rozdzielenia pamięci programu i pamięci operacyjnej była także wyższa stabilność systemu urządzenia wbudowanego, ponieważ nawet w przypadku błędów w oprogramowaniu powodujących próby zapisu danych do pamięci programu, zawartość pamięci programu pozostaje nienaruszona. Z kolei pamięć operacyjna w niektórych przypadkach znajdowała się poza zakresem adresowania wskaźnika programu, co nie pozwalało na pobranie z niej instrukcji kodu maszynowego. Analiza istniejących zabezpieczeń prowadzi do stwierdzenia, że w przypadku aplikacji wbudowanych regułą jest ścisły podział na pamięć programu niemożliwą do modyfikacji przez samo oprogramowanie oraz pamięć operacyjną przeznaczoną wyłącznie do przechowywania zmiennych.

2. Strategie modyfikacji oprogramowania w biegu

Opracowanie oprogramowania autoadaptacyjnego dla aplikacji wbudowanych zakłada dopasowywanie funkcjonalności urządzenia do potrzeb użytkownika na bieżąco. Wymaga zatem ciągłego monitorowania i ilościowej oceny stopnia spełnienia potrzeb poprzez wyliczenie funkcji błędu. W zakresie funkcjonalności podstawowej oprogramowanie będzie używać bibliotek procedur najwyższej jakości, w zakresie funkcjonalności dodatkowej - bibliotek procedur uproszczonych, natomiast procedury obsługujące funkcje zbędne z punktu widzenia użytkownika mogą zostać pominięte. Takie przyporządkowanie otwiera możliwość oszczędności energii bez pogorszenia

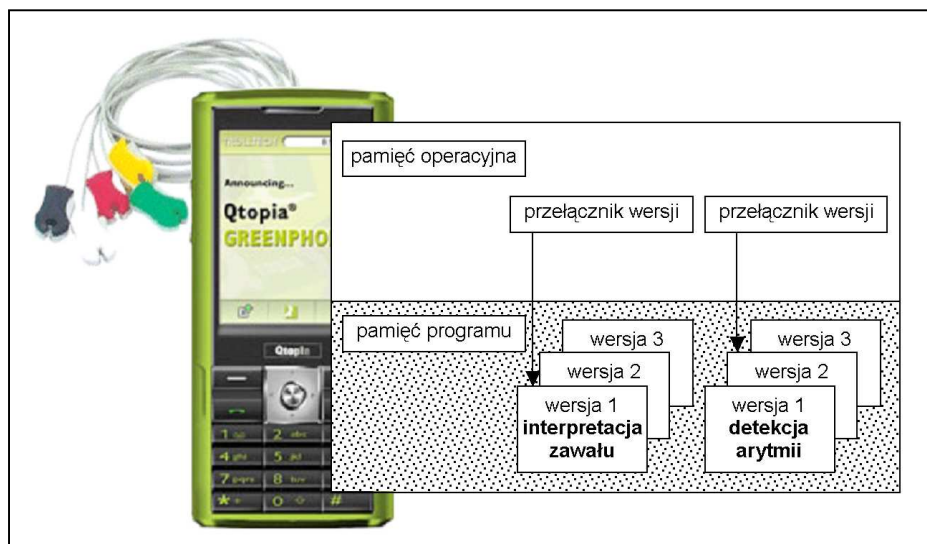
jakości przetwarzania oraz zwolnienie zasobów sprzętowych, które mogą zostać wykorzystane do dalszego podwyższenia tej jakości.

A. Przełączanie funkcjonalności oprogramowania

Wobec ograniczenia możliwości uruchomienia kodu wykonywalnego do pamięci programu o ograniczonej ilości cykli zapisu jedyną możliwością adaptacji oprogramowania pozostaje przełączanie aktywnych procedur za pomocą systemu semaforów zadeklarowanych jako zmienne w pamięci operacyjnej. Strategia ta ogranicza rozmiar bibliotek zawierających funkcje wymienne oraz ilość dostępnych wariantów każdej procedury (rys. 2). Ponieważ pamięć programu jest reprogramowalna, w miarę wolnego miejsca możliwe jest okresowe uaktualnianie oprogramowania urządzenia zdalnego także w zakresie procedur wymiennych (w tym przypadku bardziej właściwe byłoby określenie: funkcjonujących wymiennie). Zaletą tej strategii jest natomiast niewielka objętość danych wymagana do przesłania z serwera nadzorującego do urządzenia przenośnego. Komunikat adaptacji oprogramowania zawiera bowiem tylko nowe wartości flag określających wybór funkcjonujących procedur. Opisywana strategia wymaga zajęcia znacznej objętości pamięci programu, natomiast oprócz eliminacji zbędnych obliczeń pozwala także na oszczędności użycia pamięci operacyjnej. Zmienne dynamiczne związane z nie funkcjonującymi procedurami nie są bowiem deklarowane.

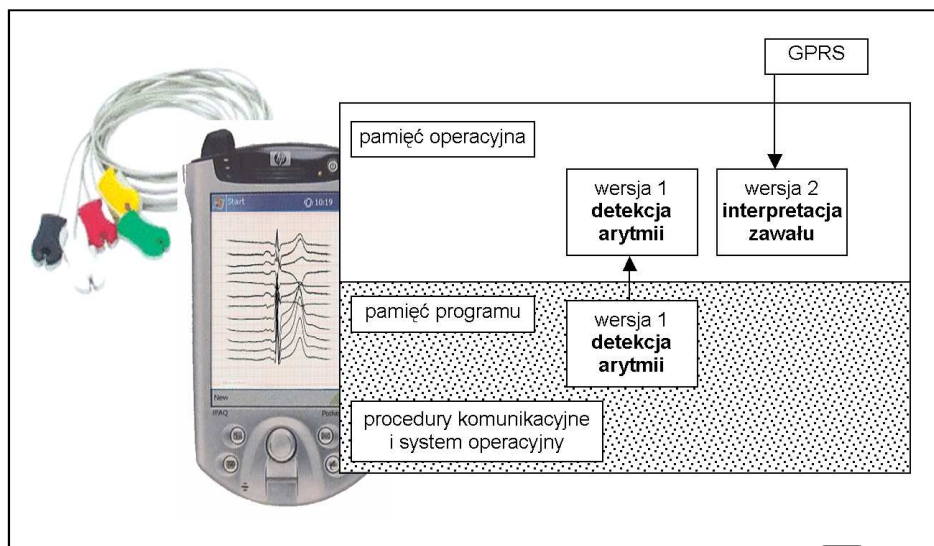
B. Wykonywanie kodu maszynowego z pamięci operacyjnej

Niektóre urządzenia przenośne umożliwiają wykorzystanie pamięci operacyjnej do przechowywania kodu maszynowego i wykonywania programów. Nie może być wówczas mowy o optymalizacji użycia pamięci operacyjnej, gdyż jest ona dodatkowo zajmowana przez procedury wykonywalne, dla których przewidziano pamięć programu (rys. 3). Można jednak w dalszym ciągu dążyć do oszczędności energii poprzez eliminację zbędnych obliczeń. Kolejną zaletą takiego rozwiązania jest możliwość użycia procedur o znacznej objętości kodu, które nie zmieściłyby się w pamięci programu.



Rysunek 2. Schemat alokacji procedur wymiennych w pamięci urządzenia przenośnego przy strategii adaptacji wykorzystującej przełączanie funkcjonalności oprogramowania

Urządzenie przenośne może zaczynać funkcjonowanie od kontaktu z serwerem kodu wykonywalnego, a więc w pamięci programu zawierać jedynie procedury komunikacyjne i namiastkę systemu operacyjnego pozwalającego na uruchomienie aplikacji wbudowanej w całości z pamięci operacyjnej. To najprostsze rozwiązanie jest jednak niewygodne w sytuacji braku gwarancji połączenia, urządzenie przenośne pozostawałoby bowiem нефункционалне. Typowym rozwiązaniem proponowanym dla tej strategii jest podział oprogramowania wbudowanego na trzy warstwy: warstwę podstawową zawartą w pamięci programu, warstwę rozszerzającą domyślną złożoną z procedur przechowywanych w pamięci programu, ale przed uruchamianiem kopiowanych do pamięci operacyjnej i możliwych do odłączenia w biegu oraz warstwę rozszerzającą zewnętrzną ładowaną za pośrednictwem łącza cyfrowego z urządzenia nadzorującego.

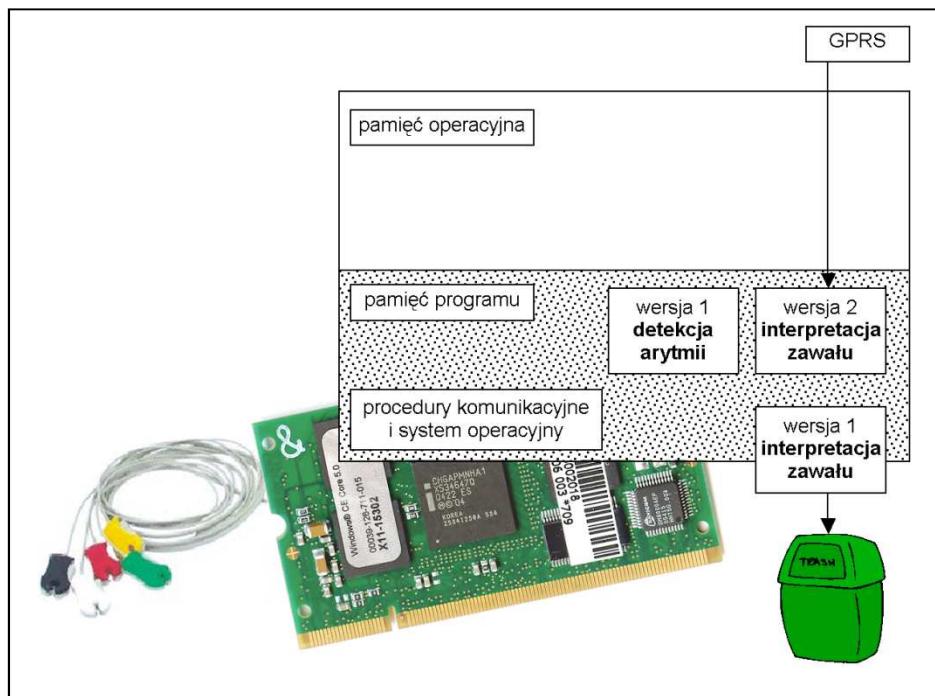


Rysunek 3. Schemat alokacji procedur wymiennych w pamięci urządzenia przenośnego przy strategii adaptacji wykorzystującej wykonywanie kodu maszynowego z pamięci operacyjnej

C. Wymiana bibliotek dołączanych dynamicznie

Niektóre systemy przenośne, szczególnie w postaci zestawów uruchomieniowych, są pozbawione ograniczeń liczby cykli zapisów do pamięci programu. Pomimo że techniczna granica ilości zapisów dla pamięci flash wynosi obecnie 105...106 a nawet więcej, przy założeniu przeprowadzania adaptacji oprogramowania co godzinę, żywotność pamięci pozwala na nieprzerwaną 10-letnią pracę urządzenia przenośnego. Brak ograniczeń pozwala na zaimplementowanie strategii adaptacji oprogramowania wykorzystującego biblioteki dołączane dynamicznie w pamięci programu (ang. dynamically linked libraries) bez konieczności umieszczania kodu maszynowego w pamięci operacyjnej (rys. 4). Rozwiązanie takie pozwala na przechowywanie w pamięci programu tylko aktualnie wykorzystywanych procedur, a w pamięci operacyjnej tylko aktualnie zadeklarowanych zmiennych, a więc w danej pojemności zasobów pozwala osiągnąć najlepszą jakość przetwarzania. Jest to okupione koniecznością przesyłania łączem cyfrowym nieco większej ilości informacji – komunikat adaptacji oprogramowania składa się z kodu wykonywalnego procedur zamieniających, natomiast kod procedur zamienianych jest usuwany z pamięci programu w miarę potrzeby odzyskania jej pojemności. Dodatkowa ilość informacji nie wpływa zresztą znacząco na podwyższenie kosztów eksploatacji - automatyczna modyfikacja funkcjonalności w warunkach typowych wykonywana jest rzadko. Ponadto, w przypadku rejestratora EKG, objętość danych przesyłana w ramach raportów diagnostycznych oraz objętość

informacji wysyłana okresowo w celu weryfikacji poprawności pracy aplikacji wbudowanej wielokrotnie przekraczają objętość przesyłanego kodu maszynowego.



Rysunek 4. Schemat alokacji procedur wymiennych w pamięci urządzenia przenośnego przy strategii adaptacji wykorzystującej wymianę bibliotek dołączanych dynamicznie

3. Implementacja i testowanie

W przypadku przenośnego rejestratora elektrokardiograficznego z interpretacją optymalizacja funkcjonalności polega na wyznaczeniu z największą dokładnością tych parametrów diagnostycznych EKG, które są uznane za najbardziej istotne w kontekście bieżącego stanu pacjenta. W systemach bez adaptacji, czyli wszystkich stosowanych obecnie, na podstawie zapisu elektrokardiogramu wyliczane są wszystkie parametry diagnostyczne, a następnie lekarz znając stan pacjenta wybiera najistotniejsze stosownie do hipotezy diagnostycznej. Ponieważ w wyniku odrębnych prac badawczych [6] uzyskano specyfikację medycznej istotności parametrów diagnostycznych EKG w kontekście dwunastu najczęstszych patologii, a stan pacjenta może być opisany na podstawie wartości parametrów diagnostycznych z medycznie akceptowaną dokładnością [7], możliwe jest automatyczne wyznaczenie funkcji błędu (por. 2A) na podstawie odchyłek poszczególnych parametrów. Wartość

prawdziwa parametrów diagnostycznych nie jest znana, a główną przyczyną niedokładności parametrów obliczanych przez urządzenie przenośne są ograniczenia jego zasobów. Zatem w roli wartości odniesienia zastosowano parametry diagnostyczne obliczane przez serwer adaptacji, a więc w środowisku pozbawionym ograniczeń, na podstawie okresowo przesyłanych fragmentów zapisu elektrokardiogramu. Dodatkową trudność w oprogramowaniu serwera adaptacji stanowił brak niezależności pomiędzy parametrami diagnostycznymi, spowodowany faktem, że drzewo procedur diagnostycznych zawiera wspólne węzły (procedury) w ścieżkach obliczeń poszczególnych parametrów. Wymieniając takie procedury w ramach adaptacji oprogramowania należy uwzględnić wzajemną zależność parametrów, aby przy okazji uproszczenia obliczeń parametrów o małej istotności nie obniżyć jakości parametru współzależnego jeżeli jest on bardzo istotny.

Przenośny aparat EKG został zaimplementowany w postaci dwóch modułów. Moduł akwizycji sygnału był oparty na rejestratorze Aspekt 500 (prod. Aspel) zawierający 8 kanałów, o rozdzielczości 12 bitów przy częstotliwości 500 próbek na sekundę i połączony z modułem interpretacyjno-komunikacyjnym łączem Bluetooth. Połączenie bezprzewodowe umożliwiała spełnienie norm bezpieczeństwa dla urządzeń medycznych. Moduł interpretacyjno-komunikacyjny został zaimplementowany na trzech platformach sprzętowych stosownie do oferowanych możliwości z użyciem trzech opisanych strategii adaptacji (tab. 1).

Tabela 1. Porównanie platform sprzętowych dla auto-adaptacyjnego systemu monitoringu kardiologicznego

Urządzenie	Uwagi	Procesor	Pamięć RAM / Flash	Interfejsy wejścia – wyjścia	Strategia adaptacji
Greenphone	programowany telefon	PXA270 (312 MHz)	64 / 128 MB	USB, MiniSD, Bluetooth, GSM	Przełączanie funkcjonalności oprogramowania
HP iPAQ H6300	komputer klasy PDA	TI OMAP 5110 (168 MHz)	64 / 64 MB	USB, SD / MMC, Bluetooth, GSM, WiFi	wykonanie kodu maszynowego z pamięci operacyjnej
Colibri PXA270	moduł uruchomieniowy	PXA270 (512 MHz)	64 / 32 MB	USB, SD / MMC, Bluetooth, GSM, WiFi, X VGA, I2C	Wymiana bibliotek dołączanych dynamicznie

Komunikacja z serwerem adaptacji obsługującym jednocześnie archiwizację parametrów diagnostycznych monitorowanego elektrokardiogramu była realizowana za pomocą dwukierunkowego cyfrowego łącza GPRS.

4. Rezultaty

Prototyp systemu nadzoru kardiologicznego o ograniczonej skali wykorzystywał po stronie rejestratora auto-adaptacyjne oprogramowanie interpretacyjne, a po stronie serwera oprogramowanie kontrolno-zarządzające na platformie Linux OS. Do testów wykorzystano bazę 857 sygnałów połączonych sztucznie z zapisów fizjologicznych oraz zawierających 12 najczęstszych patologii [8]. Głównym celem testu było badanie zachowania procedur auto adaptacyjnych w warunkach nagłej zmiany stanu pacjenta. Wprowadzono względną wartość progową funkcji błędu wynoszącą w zależności od przyczyny: 2% dla błędów detekcji QRS i częstości akcji serca, 5% dla określenia granic załamków i poziomu odcinka ST, 10% dla błędów klasyfikacji morfologii i 20% dla pozostałych parametrów. W przypadku, gdy pojedyncza modyfikacja oprogramowania interpretacyjnego nie powodowała obniżenia funkcji błędu poniżej założonego progu, dopuszczono wykonanie trzech kolejnych kroków adaptacji oprogramowania.

Spośród całkowitej liczby adaptacji, 768 (89,6%) przypadków było prawidłowych podczas gdy w pozostałych 10,4% przypadków adaptacja została przeprowadzona błędnie, głównie z powodu niewłaściwego oszacowania dostępnych zasobów. Przeszacowanie zasobów prowadzące do próby uruchomienia procedury żądającej niedostępnych zasobów i w konsekwencji przerwy w interpretacji wykonywanej przez system przenośny miało miejsce w 27 (3%) przypadkach. W przypadku 63.1% modyfikacji pojedyncza iteracja powodowała przywrócenie funkcji błędu w założone granice. W przypadku 19.3% modyfikacji wykonanie czterech iteracji modyfikujących nie powodowało osiągnięcia założonej zbieżności. Opóźnienie parametrów diagnostycznych wynikłe z przeprowadzanej w biegu adaptacji oprogramowania interpretacyjnego zostało zmierzone w rzeczywistych warunkach łączności GPRS i wynosiło średnio 6 sekund (maksymalna wartość: 17.1 s w przypadku czterech iteracji).

Podsumowanie

Strategie automatycznej adaptacji oprogramowania wbudowanego zostały przedstawione na przykładzie telemedycznego rejestratora EKG. Wykazano konieczność dostosowania metody adaptacji do ograniczeń wynikających z użytego sprzętu. Przedstawione rozważania i wnioski z testów prototypu mogą zostać wykorzystane w szerokim zakresie urządzeń testowanych i uaktualnianych zdalnie przez producenta lub system nadzorujący. Analogiczny sposób modyfikacji funkcjonalności urządzenia wydaje się stwarzać nowe obszary zastosowań dla usług abonenckich związanych z nawigacją drogową (omijanie korków, ostrzeganie przed przeszkodami) lub telefonią komórkową (włącznie dodatkowych usług, przesyłanie plików). W rezultacie z punktu widzenia użytkownika urządzenie przenośne może lepiej spełniać jego oczekiwania bez konieczności wdrażania krótkoseryjnej i kosztownej produkcji urządzeń o wysokim stopniu specjalizacji.

Literatura:

- [1]. Fayn J. et al., Towards New Integrated Information and Communication Infrastructures in E-Health. Examples from Cardiology, Computers in Cardiology, vol. 30, pp.113–116, 2003.
- [2]. Pinna G. D. et al., Home Telemonitoring of Chronic Heart Failure Patients: Novel System Architecture of the Home or Hospital in Heart Failure Study, Computers in Cardiology, vol. 30, pp. 105-108, 2003.
- [3]. Paoletti M., Marchesi C., Low computational cost algorithms for portable ECG monitoring units. IFMBE Proc. Medicon 2004, paper 231.
- [4]. Augustyniak P., Content-Adaptive Signal and Data in Pervasive Cardiac Monitoring, Computers in Cardiology, vol. 32, pp. 825-828, 2005.
- [5]. Augustyniak P., Tadeusiewicz R., Modeling of ECG Interpretation Methods Sharing Based on Human Experts Relations, Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, str. 4663-4669, Aug 30-Sept 3, 2006.
- [6]. Augustyniak P., Tadeusiewicz R., Assessment of electrocardiogram visual interpretation strategy based on scanpath analysis Physiological Measurement, vol. 27, pp. 597-608, 2006.
- [7]. Straszecka E., Straszecka J., Uncertainty and imprecision representation in medical diagnostic rules. IFMBE Proc, Medicon 2004, paper 172.
- [8]. Willems JL., Common Standards for Quantitative Electrocardiography 10-th CSE Progress Report, Leuven: ACCO publ. 1990.

Akty normatywne:

- [1]. Polskie Towarzystwo Kardiologiczne: Standardy postępowania w badaniu holterowskim. Warszawa, 1997.
- [2]. IEC (2001) 60601-2-47. Medical electrical equipment: Particular requirements for the safety, including essential performance, of ambulatory electrocardiographic systems.

Noty techniczne:

- [1]. Colibri Xscale PXA270 DataSheet, Toradex,
<http://www.toradex.com/downloads/Colibri%20PXA270%20datasheet.pdf>
- [2]. Handhelds.org – Open Source for handheld devices,
<http://www.handhelds.org/moin/moin.cgi/HPiPaqH6315>
- [3]. HP iPAQ h6300 Pocket PC Series, Hewlett-Packard,
<http://h20000.www2.hp.com/bizsupport/TechSupport/Home.jsp?&lang=pl&cc=pl&prodTypeId=215348&prodSeriesId=430120&lang=pl&cc=pl>
- [4]. Intel X-Scale Microarchitecture – Technical Summary, Intel Corporation.
- [5]. Aspel, AsPEKT 500 - Nadajnik KBEKG2 v.001,
<http://www.aspel.com.pl/index.php?lang=pl&pg=372>

Informacja o autorze:

dr hab. inż. Piotr Augustyniak, profesor nadzwyczajny w Wyższej Szkole Bankowości i Finansów w Bielsku-Białej