

Testowanie oprogramowania wirtualnego przyrządu pomiarowego

Marek Florczyk*

W wirtualnych przyrządach pomiarowych składających się z części programowej i sprzętowej, część programowa obejmuje program znajdujący się w komputerze i oprogramowanie zaimplementowane w sprzęcie. Przeprowadzenie testów takiego przyrządu jest trudne, bo trudno określić założenia, tak by testy były kompleksowe i obejmowały wszystkie najistotniejsze elementy przyrządu. Model wirtualnego przyrządu pomiarowego można stworzyć na podstawie zbioru parametrów opisujących części programową i sprzętową oraz powiązania pomiędzy nimi (zwykle interfejs komunikacyjny). Na podstawie takiego modelu można przeprowadzać kompleksowe testy części programowej wirtualnego przyrządu pomiarowego oraz analizować uzyskane wyniki, co zostanie przedstawione w artykule.

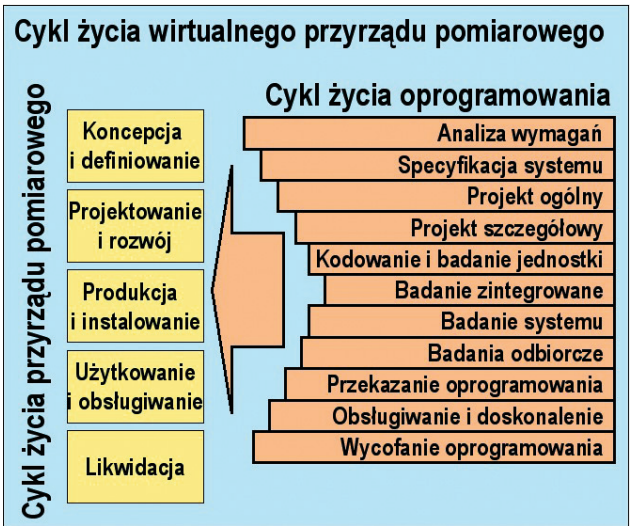
Testing of Virtual Instruments software. The methods of VI software testing are very complicated. Virtual Instruments are build form software and hardware part. The software consist of GUI part and software part, that is implemented in hardware. New parametric method, presented in this article, allow to do complete functional Virtual Instrument software tests.

Wprowadzenie

Rosnąca konkurencja na rynku wytwórców przyrządów pomiarowych powoduje, że ulega skróceniu czas ich projektowania i w rezultacie pojawienia się na rynku. Jedną z najdłużej trwających faz procesu projektowania jest testowanie przyrządu pomiarowego. Chęć skrócenia fazy testowania powoduje, że istnieje duże zapotrzebowanie na nowe metody testowania, które pozwoliłyby na jego skrócenie bez pogorszenia parametrów metrologicznych i funkcjonalnych projektowanego przyrządu pomiarowego. Zgodnie z normą PN-IEC 60300-3-6 [1] (rys. 1), jeżeli wyrób ma część programową, to testom powinna podlegać także jego część programowa. Testy części programowej powinny obejmować:

- badanie jednostek programowych (kod źródłowy)
- badania zintegrowane (całych modułów – kod źródłowy)
- badanie całego systemu (kod źródłowy)
- badania odbiorcze (testy akceptacyjne).

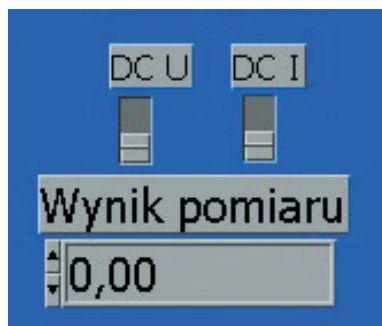
Przeprowadzenie testów tylko jednego typu, np. związanych z inspekcją kodu źródłowego (przeglądem kodu źródłowego pod kątem usunięcia ewentualnych błędów) aplikacji jest niewystarczające, gdyż testy te nie dotyczą parametrów funkcjonalnych przyrządu pomiarowego, a w szczególności interfejsu użytkownika [3, 4]. Na podstawie przeprowadzonej inspekcji kodu źródłowego nie jest możliwe określenie proble-



Rys. 1.

mów związanych z użytkowaniem przyrządu pomiarowego. Przyrząd, mimo tego że jego kod źródłowy nie będzie zawierał błędów, może zachowywać się niezgodnie z oczekiwaniami użytkownika. Na przykład, w tradycyjnym przyrządzie pomiarowym zazwyczaj niemożliwe jest wybranie jednocześnie dwóch funkcji pomiarowych, np. pomiaru napięcia i prądu. Przy stosowaniu przyrządu wirtualnego jest to możliwe, jeżeli interfejs użytkownika nie został skonstruowany

* Mgr inż. Marek Florczyk – Uniwersytet Zielonogórski, Instytut Metrologii Elektrycznej



Rys. 2.

prawidłowo (rys. 2). Dlatego, na etapie projektowania przyrządu pomiarowego konieczne jest określenie wyglądu i zasad działania interfejsu użytkownika, a przed udostępnieniem urządzenia do sprzedaży przeprowadzenie pełnego zakresu testów obejmujących testowanie zarówno kodu źródłowego aplikacji jak i jego funkcjonalności.

Do przeprowadzenia testów oprogramowania można wykorzystać cały wachlarz dostępnych metod testowania.

Techniki przeprowadzania testów można podzielić na następujące rodziny [2]:

- *random testing techniques*, w których dane generowane do przeprowadzania testów mają charakter losowy
- *functional testing techniques*, w których dane generowane do testów bazują na specyfikacji testowanego systemu, a testy są przeprowadzane zazwyczaj metodą *black-box*
- *control flow testing techniques*, w których testowane są ścieżki przepływu informacji w systemie, a testy są przeprowadzane metodą *white-box*
- *data flow testing techniques*, w których, w ramach poszczególnych stanów w jakich znajduje się system, analizowane są wartości poszczególnych danych
- *mutation testing techniques* – bazujące na modelowaniu typowych błędów popełnianych przez użytkowników i obserwacji skutków ich popełnienia.
- *regression testing techniques* – w których przeprowadza się testy wykorzystujące wcześniej zebrane dane z testów (zamknięta pętla)
- *improvement testing techniques*, w których testy mają za zadanie jak najwcześniejsze wykrycie ewentualnych błędów.

Najczęściej do przeprowadzania testów oprogramowania są stosowane następujące metody [3, 4, 5, 6, 7, 8]:

- Black-Box (czarna skrzynka)
- White-Box (biała lub przezroczysta skrzynka)
- związane z testowaniem automatycznym (m.in. wykorzystanie sieci neuronowych, algorytmów genetycznych i ewolucyjnych).

Niektóre wybrane metody testowania oprogramowania można zastosować do testowania wirtualnych przyrządów pomiarowych [9, 10], jednak do tej pory brako-

wało metody testowania, która pozwalałaby uwzględnić podczas przeprowadzania testów zarówno część sprzętową jak i programową wirtualnego przyrządu pomiarowego oraz która także pozwalałaby przeprowadzić analizę uzyskanych wyników testów.

Poruszane dotychczas w publikacjach metody przeprowadzania testów dotyczyły między innymi:

- badania modelu przyrządu pomiarowego [11]
- badania algorytmów przetwarzania przyrządu pomiarowego [12]
- testowania czasu przetwarzania określonych danych przez system komputerowy [10]
- diagnostyki uszkodzeń przyrządów pomiarowych [13].

Brak jest natomiast badań prowadzonych w zakresie testowania funkcjonalnego przyrządów pomiarowych, a szczególności wirtualnych przyrządów pomiarowych. Lukę związaną z brakiem takich metod, może wypełnić zaproponowana przez autora parametryczna metoda testowania funkcjonalnego wirtualnych przyrządów pomiarowych.

Parametryczna metoda testowania funkcjonalnego wirtualnego przyrządu pomiarowego

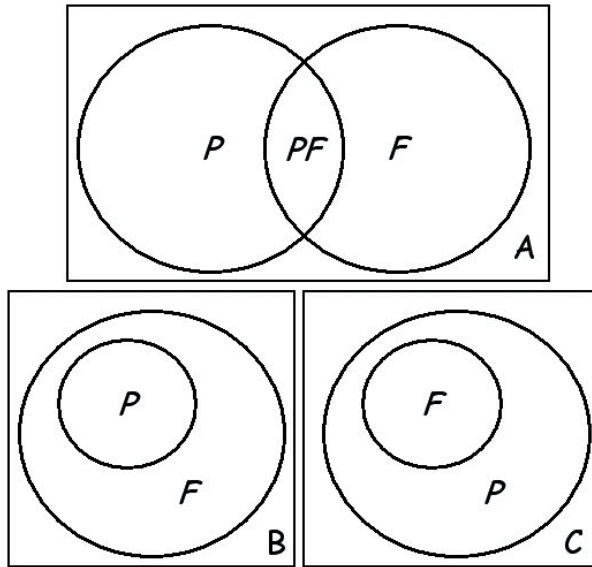
Każdy wirtualny przyrząd pomiarowy można scharakteryzować za pomocą trzech zbiorów parametrów. Te trzy zbiory to:

- zbiór F (część programowa) zawierający parametry programowe stanowiące część programową i interfejs użytkownika
- zbiór P (część sprzętowa) zawierający parametry sprzętowe części pomiarowej
- zbiór PF (powiązania) zawierający parametry opisujące powiązania pomiędzy częścią programową i sprzętową.

Zbiory parametrów opisujących wirtualny przyrząd pomiarowy są zbiorami pewnych wartości, które z danym przybliżeniem można określić dla każdego systemu. Na rys. 3 zostały przedstawione trzy najbardziej charakterystyczne przypadki:

- pierwszy (A) przedstawia system, w którym istnieją powiązania pomiędzy częścią programową i sprzętową PF
- drugi (B), gdzie parametry sprzętowe stanowią podstawę dla parametrów programowych
- trzeci (C), gdzie parametry programowe ściśle odzwierciedlają pewne parametry sprzętowe.

Biorąc pod uwagę rozwój współczesnej techniki pomiarowej można stwierdzić, że rys. 3. przedstawia trzy fazy rozwoju przyrządów pomiarowych. Począwszy od prostych przyrządów pomiarowych zawierających mikroprocesor (rys. 3B) poprzez urządzenia podłączane interfejsem do komputera (rys. 3A) aż do zaawansowanych wirtualnych przyrządów pomiarowych, w których część programowa stanowi najbardziej rozbudowany element przyrządu (rys. 3C). Wirtualny przyrząd po-



Rys. 3.

miarowy możemy przedstawić za pomocą zależności:

$$MJWPP = \sum_i \sum_j f_{(i,j)} w_{f(j,i)} + \sum_i \sum_j p_{(i,j)} w_{p(j,i)} + \sum_i \sum_j pf_{(i,j)} w_{pf(j,i)} \quad (1)$$

gdzie:

- f – macierz parametrów związanych z częścią programową
- p – macierz parametrów związanych z częścią sprzętową
- pf – macierz parametrów opisujących powiązania pomiędzy częścią programową i sprzętową
- w_f – macierz wag parametrów f
- w_p – macierz wag parametrów p
- w_{pf} – macierz wag parametrów pf .

MJWPP stanowi miarę jakości wirtualnego przyrządu pomiarowego.

Zaproponowana przez autora zależność (1) umożliwia przeprowadzenie testów oprogramowania wirtualnego przyrządu pomiarowego przy założeniu, że poszczególne parametry części programowej będą stanowiły funkcje realizowane przez wirtualny przyrząd pomiarowy, natomiast waga danego parametru będzie określała prawdopodobieństwo wyboru danej funkcji przez użytkownika (lub wirtualny przyrząd pomiarowy). Przyjęcie jako wagi – prawdopodobieństwa wyboru określonego parametru jest zasadne, ponieważ zazwyczaj pomiar określonej wielkości za pomocą wirtualnego przyrządu pomiarowego jest oparty na scenariuszu pomiaru, który stanowi instrukcja użytkownika tego przyrządu. Interfejs użytkownika każdej aplikacji, w tym panelu wirtualnego przyrządu pomiarowego, w fazie prototypowania powstaje w wyniku uzgodnień pomiędzy użytkownikiem projektowanego przyrządu a jego projektantem [14]. Od samego początku się zakłada, w jaki sposób przyrząd pomiarowy będzie obsługiwany – powstaje więc scenariusz jego obsługi.

Przed wyborem metody przeprowadzenia testów należy określić:

- ryzyko związane z przeprowadzeniem testów (czy możliwe jest uszkodzenie przyrządu pomiarowego w wyniku przeprowadzania określonych testów)
- czas jaki będzie potrzebny na przeprowadzenie określonego typu testów (może się okazać, że uwzględnienie wszystkich

parametrów podczas testów nie będzie możliwe)

- zasoby jakie mają być wykorzystane podczas przeprowadzania testów (infrastruktura pomiarowa i sprzęt komputerowy)
- liczba osób przeprowadzających testy lub biorących udział w testach
- budżet przeznaczony na przeprowadzenie testów.

Metoda przeprowadzenia testów zależy w dużej mierze od architektury wirtualnego przyrządu pomiarowego. Podczas testów architektury rozproszonej główny nacisk powinien być położony na przetestowanie interfejsu komunikacyjnego.

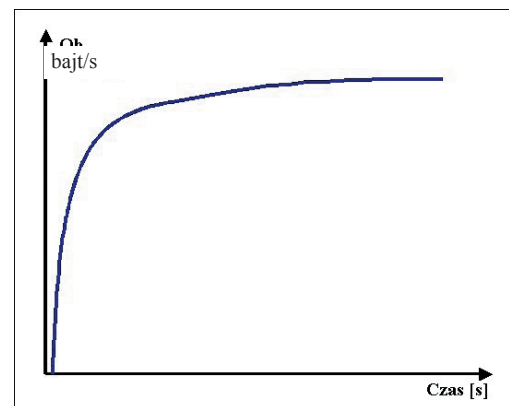
Gdy mamy do czynienia z jednym serwerem pomiarowym udostępniającym wyniki pomiaru poprzez sieć Internet, należy przetestować wydajność serwera w myśl zaproponowanej przez autora zależności [15]:

$$Ob = \frac{l_x \cdot w_f}{t_x - t_o} \quad (2)$$

gdzie:

- l_x – liczba pobranych wyników pomiaru
- w_f – liczba bajtów, które zajmuje wynik pomiaru
- t_o – czas rozpoczęcia testu
- t_x – czas zakończenia testu.

Przeprowadzając symulację polegającą na wymuszeniu pobrania określonej ilości razy danego wyniku pomiaru w określonym czasie, otrzymujemy wykres (rys. 4), na podstawie którego jesteśmy w stanie określić maksymalną liczbę pobrań, na którą pozwala serwer.



Rys. 4.

Tak przeprowadzony test obciążeniowy powinien uwzględniać, że na serwerze znajdują się skrypty przetwarzające inne informacje, oraz że być może serwer ma inne udostępnione do pobrania dokumenty.

Dla takiego przypadku zależność (2) przyjmie postać:

$$Ob_{\max} = \frac{\sum_{i=1}^n (l_{xi} \cdot w_{fi}) + \sum_{j=1}^m (p_{xj} \cdot w_{pj})}{t_x - t_0} \quad (3)$$

gdzie:

- l_{xi} , p_{xi} – i-ty (j-ty) wynik pomiaru pobrany z (wysłany do) serwera
- w_{fi} , w_{pj} – liczba bajtów, które zajmuje i-ty (j-ty) dokument
- t_x – chwila zakończenia testu
- t_0 – chwila rozpoczęcia testu.

Testy takie można przeprowadzić za pomocą specjalistycznego oprogramowania, które służy do generowania żądań wykonania określonych operacji przez serwer (pozwala ono na wygenerowanie od kilku do kilkunastu tysięcy jednoczesnych zapytań o określone informacje do serwera). Dla serwera plików pomiarowych będą generowane zapytania o pliki.

Jeżeli przyjmiemy na podstawie zależności (1), że:

$$Func = \sum_i \sum_j f_{(i,j)} w_{f(j,i)} \quad (3)$$

$$Tech = \sum_i \sum_j p_{(i,j)} w_{p(j,i)} \quad (4)$$

$$Zal = \sum_i \sum_j pf_{(i,j)} w_{pf(j,i)} \quad (5)$$

to informację użyteczną z punktu widzenia przeprowadzonych testów na podstawie wzoru (3, 4, 5), będą stanowiły elementy umieszczone na przekątnej macierzy. Ślad macierzy $slTech$ można wyznaczyć:

$$slFunc = \sum_i a_{ii} \quad (6)$$

gdzie: a_{ii} – jest elementem macierzy $Func$.

Na tej podstawie:

$$Pr = \frac{slFunc}{\max \text{praws}Func} * 100\% \quad (7)$$

Pr określa jaki procent kombinacji parametrów można wykonać bezbłędnie oraz określa maksymalną niezawodność testowanego oprogramowania wirtualnego przyrządu pomiarowego. Sam współczynnik Pr nie daje jednak szczegółowych informacji o kombinacjach parametrów, których kolejne użycie powoduje powstanie błędu. Poszczególne elementy przekątnej macierzy $Func$, na podstawie której wyznacza się ślad macierzy, grupują wszystkie prawdopodobieństwa kombinacji danego parametru z pozostałymi, co pozwala na określe-

nie współczynnika Pr_{param} , którego wartość wynosi:

$$Pr_{param} = a_{ii} * 100\% \quad (8)$$

gdzie a_{ii} stanowi element macierzy $Func$, $i \in \langle 1, ilparam \rangle$, $ilparam$ określa liczbę parametrów podlegających analizie. Współczynnik Pr_{param} określa prawdopodobieństwo tego, że użytkownik nie spowoduje błędu urządzenia, wybierając kolejno określony parametr a następnie dowolny inny. W celu określenia szczegółowego rozkładu prawdopodobieństwa dla poszczególnych parametrów należy skorzystać z wzoru:

$$MJWPP = \sum_i \sum_j (f_{(i,j)} w_{f(i,j)}) + \sum_i \sum_j (p_{(i,j)} w_{p(i,j)}) + \sum_i \sum_j (pf_{(i,j)} w_{pf(i,j)}) \quad (9)$$

Elementy tak powstałych macierzy ($Func$, $Tech$, Zal) obliczone na podstawie zależności (9) określają prawdopodobieństwo tego, że zostanie wykonana przez użytkownika dana kombinacja parametrów i że nie spowoduje to powstania błędu urządzenia (Na rys. 5. został przedstawiony przykład zobrazowania macierzy $Func$. Czym wyższa wartość określonego elementu macierzy, tym większe prawdopodobieństwo tego, że użytkownik wybierze dany parametr i nie spowoduje on powstania błędu lub defektu oprogramowania wirtualnego przyrządu pomiarowego).

Badania

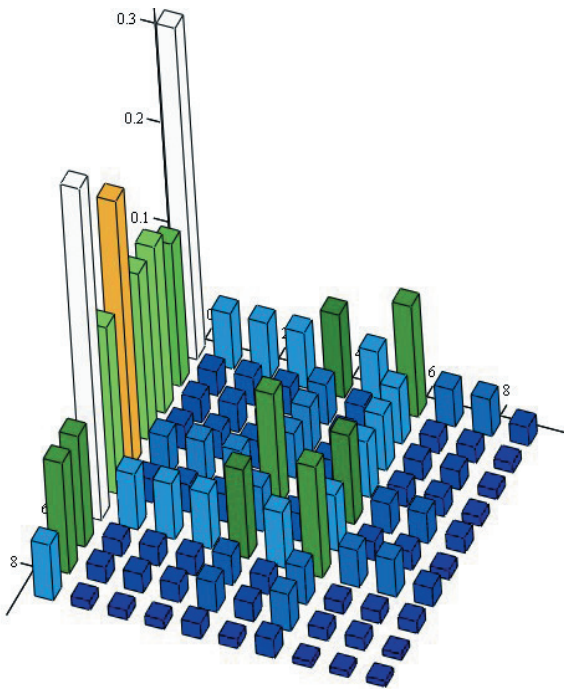
W toku przeprowadzonych dotychczas badań opracowano parametryczną metodę testowania funkcjonalnego wirtualnych przyrządów pomiarowych. Metodę tę można również wykorzystać do przeprowadzania testów tradycyjnych przyrządów pomiarowych. Autor wykonał takie badania dla multimetru V563, które wykazały między innymi, że prawdopodobieństwo wybrania określonej dowolnej kombinacji parametrów i niespodowodowania błędu wyniosło 93 %.

Przeprowadzenie testów multimetru V563 pozwoliło także stwierdzić, że możliwości programowania tego multimetru są znacząco większe niż opisane w jego instrukcji obsługi, a także, że multimetr ten ma znaczące wady m.in. następuje zmiana funkcji pomiarowej w chwili nieotrzymania pełnego rozkazu (np. przy przełączaniu zakresu z pomiaru rezystancji na pomiar napięcia stałego).

W toku dalszych badań zostanie stworzone oprogramowanie, które bazując na wzorze (1) pozwoli na przeprowadzenie testów wirtualnego przyrządu pomiarowego składającego się z karty akwizycji danych pomiarowych i oprogramowania.

Podsumowanie

W artykule przedstawiono ogólny zarys parametrycznej metody testowania funkcjonalnego wirtualnych przyrządów pomiarowych. Przedstawioną metodę można wykorzystać do testowania części programowej jak i sprzętowej przyrządu pomiarowego, a po przeprowadzeniu testów do przeprowadzenia analizy uzyskanych wyników. W artykule został przedstawiony również sposób testowania wydajności serwera plików pomiarowych, który udostępnia dane poprzez sieć internetową



Rys. 5.

Bibliografia

- [1] Norma PN-IEC 60300-3-6:2002 Zarządzanie niezawodnością. Część 3: Przewodnik zastosowań. Arkusz 6: Aspekty niezawodności związane z oprogramowaniem.
- [2] Jurysto N., Moreno A., Vegas S.: Reviewing 25 Years of Testing Technique Experiments, *Empirical Software Engineering*, 9, (2004), s. 7-44.
- [3] Florczyk M.: *Przegląd wybranych metod testowania aplikacji informatycznych pod kątem ich zastosowania do testowania wirtualnych przyrządów pomiarowych*, *Pomiary Automatyka Kontrola*, nr 7-8 (2002), s. 25-28.
- [4] Florczyk M.: *Testowanie interfejsu użytkownika wirtualnego przyrządu pomiarowego*, *Elektryka* nr 1(6), (2003), *Prace naukowe Politechniki Radomskiej*, s. 25-30.
- [5] Ross K.: *Practical guide to software system testing*, K. J. Ross & Associates Pty. Ltd., (1998).
- [6] Binder R.: *Testowanie systemów obiektowych, modele wzorce i narzędzia*, WNT, (2003).
- [7] Patton R.: *Testowanie oprogramowania*, MIKOM, (2002).
- [8] Paulk Mark C., Curtis B., Chrissis M.B., Weber C.V., *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute, Carnegie Mellon University Pittsburgh, Pennsylvania, (1996).
- [9] Wichmann B.: *Measurement System Validation: Validation of Measurement Software*, National Physical Laboratory, Middlesex, UK, (2000), s.18-56.
- [10] Winiecki W.: *Wirtualne przyrządy pomiarowe*. *Prace Naukowe Elektronika* z. 145, Politechnika Warszawska, (2003).
- [11] Gajda J., Szyper M.: *Modelowanie i badania symulacyjne systemów pomiarowych*, AGH, Kraków (1998).
- [12] Bogacz R., Jakubiec J., Al Raimi H., Roj J.: *System do badań własności metrologicznych różnych rozwiązań sprzętowych i programowych systemów pomiarowych*. *Mat. Konf. „Systemy Pomiarowe w Badaniach Naukowych i w Przemysle”*, SP'96, Zielona Góra, (1996), s. 7-14.
- [13] Hojda J., Zięba J.: *Nowoczesne metody diagnozowania uszkodzeń przyrządów pomiarowych*, *Pomiary Automatyka Kontrola*, nr 10 (2003), s. 33-35.
- [14] Carroll J.M., *Five reasons for scenario-based design*, *Interacting with Computers*, 13, (2000), s. 43-60.
- [15] Florczyk M.: *Monitorowanie i testowanie systemu obiegu dokumentacji pobieranej i przesyłanej drogą elektroniczną*. Referat zgłoszony na konferencję *Komputerowe Systemy Wielodostępne*, Ciechocinek (2004). ■