

**Marek FLORCZYK**

UNIwersytet Zielonogórski, Instytut Metrologii Elektrycznej

## Liczba przypadków testowych w procesie testowania funkcjonalnego wirtualnych przyrządów pomiarowych

Dr inż. Marek FLORCZYK

Ukończył studia magisterskie na Wydziale Elektrycznym na Politechnice Zielonogórskiej w 1997 roku. Następnie pracował w firmie Max Elektronik S.A. W latach 1999-2007 pracował na stanowisku asystenta na Politechnice Zielonogórskiej, a następnie Uniwersytecie Zielonogórskim. Tytuł doktora nauk technicznych uzyskał w 2007 roku. Specjalizuje się w testowaniu oprogramowania, w szczególności oprogramowania wirtualnych przyrządów pomiarowych oraz serwisów internetowych.



e-mail: m.florczyk@ime.zu.zgora.pl

### Streszczenie

W artykule została przedstawiona metoda wyznaczania liczby testów na podstawie znajomości liczby elementów interfejsu użytkownika przyrządu wirtualnego. Powiązано liczbę przypadków testowych z ogólną liczbą przeprowadzonych testów, gdy testy przeprowadzane są dla kombinacji elementów interfejsu użytkownika przyrządu wirtualnego. Przedstawiono też sposób określania czasu przeprowadzenia testów na podstawie znajomości liczby przypadków testowych.

**Słowa kluczowe:** wirtualny przyrząd pomiarowy, testowanie oprogramowania, metoda analizy granic.

### The number of test cases used in process of functional testing virtual instruments

#### Abstract

In his paper presented is the method of test cases calculation on the basis of number of user interface elements of virtual instrument. The number of test cases was related to overall number of carried out tests, when the test are carried out for combination of user interface elements. The manner of determining time for the tests was presented. That was done on the basis of knowledge of number of test cases.

**Keywords:** virtual instrumentation, test cases, software testing, boundary values analysis.

### 1. Wstęp

Wirtualne przyrządy pomiarowe stanowią wyodrębnioną grupę przyrządów, które to składają się z komputera ogólnego przeznaczenia i dołączonych do niego - poprzez dowolny kanał komunikacyjny - sprzętowych bloków funkcjonalnych (wewnętrznych i/lub zewnętrznych), przy czym funkcje i możliwości są określane zarówno przez sprzęt jak i oprogramowanie, a obsługa odbywa się z wykorzystaniem graficznego interfejsu użytkownika [1].

Proces projektowania wirtualnego przyrządu pomiarowego przebiega najczęściej w oparciu o model kaskadowy (rys. 1) [2]. Proces poprawy jakości w obrębie projektowania oprogramowania ma charakter ciągły, a inspekcje, przeglądy i testy są przeprowadzane na każdym z etapów. Tworząc specyfikację oraz projekt przyrządu podejmowane są działania mające na celu zapobieżenie błędem związanym z założeniami, których wykrycie np. po oddaniu przyrządu użytkownikowi wymagało by znaczących i kosztownych modyfikacji kodu źródłowego oprogramowania. Podczas etapu powstawania kodu źródłowego przeprowadzane są testy, które powinny umożliwić wykrycie błędów związanych z implementacją określonych algorytmów obliczeniowych, odzwierciedlających np. określone funkcje pomiarowe wirtualnego przyrządu pomiarowego. Testowanie oprogramowania stanowi jeden z ważniejszych etapów projektowania wirtualnego przyrządu pomiarowego. W zależności od przyjętej metodyki prowadzenia procesu projektowania, testy oprogramowania zajmują zazwyczaj około

połowy czasu fazy związanej z tworzeniem oprogramowania [3]. Próba ograniczenia czasu przeprowadzania testów jest więc zasadna z ekonomicznego punktu widzenia.



Rys. 1. Model kaskadowy  
Fig. 1. The Waterfall model

Skrócenie czasu przeprowadzenia testów uzyskuje się poprzez zautomatyzowanie procesu testowania lub poprzez ograniczenie liczby przypadków testowych. Przypadek testowy stanowi konkretny przypadek zachowania się programu wybrany w celu przeprowadzenia eksperymentu [3]. Przypadkiem testowym będzie np. dokonanie nastawy wybranego elementu interfejsu użytkownika.

Automatyzacja testowania wirtualnego przyrządu pomiarowego, dotyczy zazwyczaj testowania strukturalnego (tzw. metody białej skrzynki), gdzie dostępny jest dla osoby przeprowadzającej testy kod źródłowy oprogramowania [4]. Ponieważ do przeprowadzania testów oprogramowania wirtualnego przyrządu pomiarowego wykorzystywany jest komputer, liczba przypadków testowych nie ma tak wielkiego znaczenia dla czasu przeprowadzenia testów, przy założeniu, że wpływ mają tylko opóźnienia związane z możliwościami obliczeniowymi komputera. Przykładem oprogramowania wspomagającego przeprowadzanie testów strukturalnych, w sposób automatyczny, (m.in. oprogramowania wirtualnego przyrządu pomiarowego napisanego w LabVIEW) jest program TestStand firmy National Instruments [5].

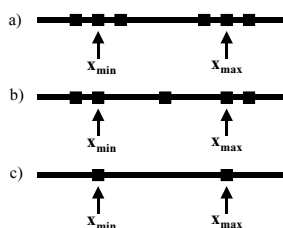
Jeżeli przeprowadzane testy obejmują także część sprzętową, a nie tylko oprogramowanie, należy uwzględnić czas wykonywania pomiarów. Liczba przypadków testowych może mieć decydujące znaczenie dla czasu trwania testów.

Testowanie funkcjonalne oprogramowania wirtualnego przyrządu pomiarowego nie zawsze może zostać przeprowadzone w sposób automatyczny, ze względu na zróżnicowany charakter obiektów znajdujących się w obrębie interfejsu użytkownika (np. elementy interfejsu użytkownika służące do realizacji nastaw, prezentacji informacji pomiarowej itd.). Także ze względu na czas realizacji określonych procedur pomiarowych przez wirtualny przyrząd pomiarowy testy przeprowadzone w sposób automatyczny mogą dawać nieprawdziwe wyniki. Różne nastawy przyrządu mogą powodować różny czas pomiaru określonych wielkości. Czas wykonywania pomiaru może stanowić jeden z warunków zaliczenia testu. Oprogramowanie służące do przeprowadzenia testu w sposób automatyczny mogłoby być bardziej skomplikowane, niż oprogramowanie wirtualnego przyrządu pomiarowego.

Ponieważ automatyzacja procesu testowania funkcjonalnego jest trudnym zadaniem, wydaje się, że jednym ze sposobów skrócenia czasu przeprowadzania testów jest taki dobór przypadków testowych, aby ich liczba była jak najmniejsza, przy jednoczesnym zachowaniu skuteczności wykrywania błędów i defektów przyrządu.

## 2. Metoda analizy granic

Metoda analizy jest jedną z częściej stosowanych metod do przeprowadzania testów elementów nastaw w wirtualnych przyrządach pomiarowych. Metoda ta pozwala na przeprowadzenie testów sprawdzających, czy istnieje możliwość wybrania (lub dokonania nastawy) dla określonego w specyfikacji przyrządu zakresu możliwych do ustawienia wartości. W opisie metody analizy granic prezentowanych jest kilkanaście sposobów przeprowadzania testów [6]. Na rys. 2 została przedstawiona idea przeprowadzania testów metodą analizy granic, dla trzech najczęściej występujących przypadków. Kwadratami zostały wyróżnione punkty przeprowadzania testów. Wartość  $x_{\min}$  i  $x_{\max}$  określają dopuszczalny zakres wartości określony w specyfikacji przyrządu dla danego elementu interfejsu użytkownika.

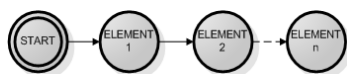


Rys. 2. Metoda analizy granic  
Fig. 2. Boundary value analysis technique

Poszukiwanie błędów elementów realizacji nastaw odbywa się najczęściej tak jak to ma miejsce na rys. 2a gdzie sprawdzeniu podlegają nie tylko wartości dopuszczalne dla zakresu przewidzianego w specyfikacji, ale także wartości, które się znajdują poza dopuszczalnym zakresem. Wewnątrz dopuszczalnego zakresu sprawdzana jest możliwość wybrania (ustawienia) wartości o jedną jednostkę (wynikającą najczęściej z rozdzielczości) większej (lub mniejszej) niż wartość graniczna. Na rys. 2b przedstawiono sposób weryfikacji specyfikacji, gdy sprawdzeniu podlega sam fakt możliwości wybrania (ustawienia) wartości z określonego zakresu (bądź spoza zakresu). Rysunek 2c prezentuje sposób sprawdzenia wartości granicznych.

## 3. Określanie liczby przypadków testowych

Obsługa wirtualnego przyrządu pomiarowego odbywa się zwykle w oparciu o przedstawioną przez producenta przyrządu instrukcję użytkownika [7]. Instrukcję stanowi scenariusz wykonywanego pomiaru. Najczęściej kolejne kroki określają wybór poszczególnych elementów interfejsu użytkownika (rys. 3). Elementami interfejsu użytkownika będą elementy realizacji nastaw oraz prezentacji informacji pomiarowej. Elementy realizacji nastaw mają najczęściej charakter numeryczny, co umożliwia zastosowanie metody analizy granic do przeprowadzenia testów.



Rys. 3. Scenariusz pomiaru  
Fig. 3. The measurement scenario

Przeprowadzenie testów funkcjonalnych powinno pozwolić na określenie stopnia poprawności wykonania przez programistę implementacji założeń dotyczących określonych algorytmów pomiarowych znajdujących się w obrębie scenariusza pomiaru wirtualnego przyrządu pomiarowego. Poszczególne przypadki testowe m.in. dla scenariusza pomiaru można przedstawić, tak jak to ma miejsce w parametrycznej metodzie testowania funkcjonalnego wirtualnych przyrządów pomiarowych, w postaci następującej macierzy [7]:

$$f = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1j} \\ f_{21} & & & f_{2j} \\ \vdots & & & \vdots \\ f_{i1} & & & f_{ij} \end{bmatrix} \quad (1)$$

gdzie np. element  $f_{12}$  oznacza wybór kolejno elementu 1 a następnie elementu 2 znajdującego się w obrębie interfejsu użytkownika.

Macierz  $f$  obejmuje więc nie tylko przypadki objęte scenariuszem pomiaru ale także te, które nie znajdują się w jego obrębie. Przeprowadzenie testów tylko w obrębie scenariusza pomiaru nie pozwala na określenie poprawności wykonania interfejsu użytkownika. Testy przeprowadzone tylko w obrębie scenariusza pomiaru nie pozwalają sprawdzić zachowania przyrządu, dla przypadków, gdy osoba przeprowadzająca pomiar będzie się kierować intuicją lub doświadczeniem i nie będzie korzystała ze wskazówek zawartych w instrukcji obsługi przyrządu [7]. Natomiast przeprowadzenie testów poza scenariuszem pomiaru wiąże się ściśle ze zwiększeniem liczby przypadków testowych, a także powoduje wydłużenie czasu przeprowadzenia testów.

Liczba przypadków testowych będzie zależała także od wybranej metody przeprowadzenia testów. Jeżeli testy dla macierzy przedstawionej we wzorze (1) zostaną przeprowadzone metodą analizy granic, bez uwzględnienia przypadków testowych znajdujących się poza scenariuszem pomiaru (tak jak to ma miejsce na rys. 3), można zaproponować następującą zależność określającą liczbę testów:

$$L_{test} = (4n - 2) \frac{l_{ptest}^2}{2} \quad \text{dla } l_{ptest} > 1 \quad (2)$$

$$L_{test} = 4n - 2 \quad \text{dla } l_{ptest} = 1 \quad (3)$$

gdzie:  $L_{test}$  – liczba przeprowadzonych testów;  $n$  – liczba elementów interfejsu użytkownika poddawana testom;  $l_{ptest}$  – liczba przypadków testowych dla pojedynczego elementu interfejsu użytkownika.

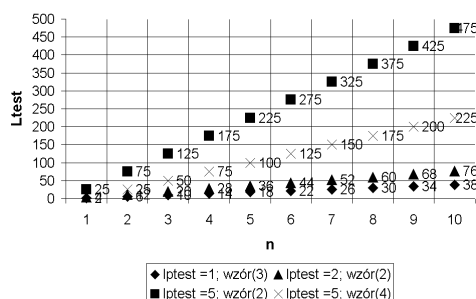
Jeżeli przyjmijemy, że testy zostaną przeprowadzone z wyłączeniem elementów znajdujących się na przekątnej macierzy  $f$ , wzory (2, 3) przyjmą następującą postać:

$$L_{test} = (2n - 2) \frac{l_{ptest}^2}{2} \quad \text{dla } l_{ptest} > 1 \quad (4)$$

$$L_{test} = 2n - 2 \quad \text{dla } l_{ptest} = 1 \quad (5)$$

Wyłączenie elementów na przekątnej macierzy  $f$  oznacza, że testom nie podlega sytuacja gdy np. następuje wybór elementu pierwszego, a następnie jeszcze raz dokonywany jest wybór tego samego elementu.

Porównanie liczby przeprowadzonych testów w zależności od liczby elementów interfejsu użytkownika oraz liczby przypadków testowych dla każdego z elementów zostało przedstawione na rys. 4.



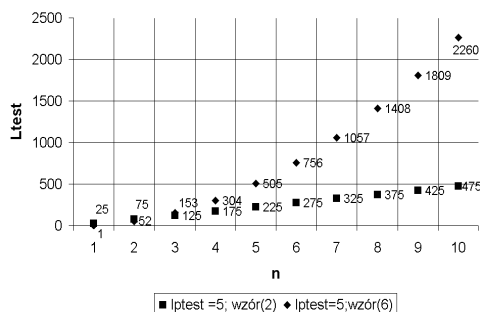
Rys. 4. Liczba przeprowadzonych testów w zależności od liczby przypadków testowych oraz liczby elementów interfejsu użytkownika  
Fig. 4. The number of carried out tests depending on test cases and elements of user interface elements

Jeżeli testy dla macierzy przedstawionej we wzorze (1) zostaną przeprowadzone z uwzględnieniem przypadków testowych znajdujących się także poza scenariuszem pomiaru, liczba wykonanych testów będzie określana na podstawie zależności:

$$L_{test} = \left( \left( (2n-2) \frac{l_{ptest}^2}{2} \right) n + n \right) \quad (6)$$

$$L_{test} = ((2n-2)n + n) \quad \text{dla} \quad l_{ptest} = 1 \quad (7)$$

Przeprowadzenie testów także poza scenariuszem pomiaru oznacza, że testy są przeprowadzane dla wszystkich możliwych kombinacji elementów interfejsu użytkownika. Porównanie liczby przeprowadzonych testów dla wariantów, gdy testy są przeprowadzane w obrębie scenariusza pomiaru oraz w obrębie scenariusza pomiaru i poza scenariuszem pomiaru zostało przedstawione na rys. 5.



Rys. 5. Porównanie liczby przeprowadzonych testów w zależności od liczby elementów interfejsu użytkownika

Fig. 5. Comparison of carried out test depending on number of elements of user interface

Na przykład zakładając, że przeprowadzanie pojedynczego testu w sposób automatyczny zajmuje 0,1s, a w sposób manualny około 1 minuty, możliwe jest określenie w oparciu o proponowane wzory, czasu przeprowadzenia testów uwzględniających różną liczbę przypadków testowych, różną liczbę elementów interfejsu użytkownika, a także różny zakres przeprowadzania testów.

Dla tak określonych założeń w tabeli 1. zostało przedstawione zestawienie określające liczbę przeprowadzonych testów dla różnej liczby przypadków testowych, gdy liczba testowanych elementów interfejsu użytkownika wynosi 25.

Tab. 1. Liczba testów z zależności od liczby przypadków testowych  
Tab. 1. The number of test depending on test cases

Liczba przypadków testowych	Liczba testów w obrębie scenariusza pomiaru	Liczba testów poza scenariuszem pomiaru
2	196	2425
4	784	9625
5	1225	15025
6	1764	21625
8	3136	38425
10	4900	60025

W tabeli 2, dla tych samych założeń zostało przedstawione zestawienie czasów przeprowadzenia testów dla różnej liczby przypadków testowych, gdy liczba testowanych elementów interfejsu użytkownika wynosi 25.

Tab. 2. Czas przeprowadzenia testów w zależności od liczby przypadków testowych

Tab. 2. The time of conduct tests depending on test cases

Sposób wykonania testów:	automatyczny	automatyczny	manualny	manualny
Liczba przypadków testowych	Testy w obrębie scenariusza pomiaru [s]	Testy poza scenariuszem pomiaru [min]	Testy w obrębie scenariusza pomiaru [godz.]	Testy poza scenariuszem pomiaru [godz.]
2	19,6	4,04	3,27	40,42
4	78,4	16,04	13,07	160,42
5	122,5	25,04	20,42	250,42
6	176,4	36,04	29,40	360,42
8	313,6	64,04	52,27	640,42
10	490,0	100,04	81,67	1000,42

Z danych przedstawionych w tabelach wynika, że możliwe jest, na podstawie zaproponowanych zależności oszacowanie zarówno czasów wykonania testów, jak i określenie liczby testów jakie należy przeprowadzić dla określonego interfejsu użytkownika przyrządu wirtualnego. Możliwe jest także wskazanie, które testy powinny zostać przeprowadzone w sposób manualny, a które w sposób automatyczny.

#### 4. Podsumowanie

W artykule zaproponowano sposób określenia liczby przypadków testowych oraz oszacowania czasu przeprowadzenia testów przyjmując założenie, że testy są wykonywane dla kombinacji elementów interfejsu użytkownika, tak jak ma to miejsce m.in. w parametrycznej metodzie testowania funkcjonalnego wirtualnych przyrządów pomiarowych. Wskazano jaki wpływ na ogólną liczbę przeprowadzonych testów (dla całego przyrządu) ma zwiększenie liczby pojedynczych przypadków testowych dla danego elementu interfejsu użytkownika.

Do przeprowadzenia testów została wykorzystana metoda analizy granic, która jest najczęściej stosowaną metodą przeprowadzenia testów. W toku dalszych prac wydaje się zasadne uwzględnienie w proponowanych zależnościach możliwości wykorzystania do przeprowadzenia testów różnych metod, dla różnych przypadków testowych. Pozwoli to na poszerzenie zakresu zastosowania proponowanych wzorów.

Należy zwrócić uwagę na fakt, że przedstawione zależności mogą znaleźć także zastosowanie w procesie testowania tradycyjnych przyrządów pomiarowych.

#### 5. Literatura

- [1] W. Winięcki: Wirtualne Przyrządy Pomiarowe. Elektronika z. 145. Politechnika Warszawska, 2003.
- [2] W.W. Royce: Managing the Development of Large Software Systems. Proceeding of IEEE WESCON, 1970.
- [3] J. Górski: Inżynieria oprogramowania w projekcie informatycznym. MIKOM, Warszawa, 2000.
- [4] G. Myers, C. Sandler, T. Badget, T. Thomas: The Art of Software Testing. John Wiley & Sons, 2004.
- [5] N. D'Anna: Next generation multitier.net enabled test systems. IEEE, Autotestcon.2005.
- [6] H.-S. J. Tsao, Ye Wu: Testing and Quality Assurance for Component-Based Software. Artech House, 2003.
- [7] M. Florczyk, W. Winięcki: The parametric method for functional testing of virtual instruments. IDAACS 2005: proceedings of the third IEEE Workshop, 2005.