

GRZEGORZ WOJARNIK

## GRAFICZNY INTERFEJS UŻYTKOWNIKA – NARZĘDZIA I METODY

### 1. Definicja graficznego interfejsu użytkownika

Wszyscy użytkownicy systemów komputerowych codziennie spotykają się ze swoistą magistralą komunikacyjną, która umożliwia komunikację z komputerem. Magistrala ta decyduje o jakości komunikacji użytkownika z systemem komputerowym i nosi nazwę interfejsu użytkownika. Aby móc o nim mówić, należy przytoczyć przynajmniej kilka jego definicji.

Według W.O. Galitza, interfejs użytkownika jest częścią komputera i działającego na nim oprogramowania, którą człowiek może zobaczyć, usłyszeć, dotknąć, mówić do niej lub zrozumieć w inny sposób, czy też nią kierować<sup>1</sup>.

Encyklopedia internetowa Wikipedia graficzny interfejs użytkownika (ang. *Graphical User Interface* – GUI) opisuje jako środowisko graficzne – ogólne określenie sposobu prezentacji informacji przez komputer i interakcji z użytkownikiem, polegające na rysowaniu i obsługiwaniu widgetów<sup>2</sup>. Widgety to wszelkiego rodzaju obiekty, za których pomocą system komunikuje się z użytkownikiem. Można do nich zaliczyć okna, przyciski, pola wyboru, przyciski radiowe (wybór jeden z wielu), przyciski zaznaczania, pola edycyjne, paski menu, menu rozwijalne, paski narzędzi, przyciski narzędzi, listy rozwijalne, struktury drze-

---

<sup>1</sup> W.O. Galitz: *The Essentials Guide to User Interface Design*. John Wiley & Sons Inc., New York 2002, s. 15

<sup>2</sup> wikipedia, [http://pl.wikipedia.org/wiki/Interfejs\\_graficzny](http://pl.wikipedia.org/wiki/Interfejs_graficzny).

wiaste, zakładka (karty, panele), okna dialogowe, paski przesuwania, etykiety, komórki, tabele i hiperłącza<sup>3</sup>.

Według encyklopedii [wiem.onet.pl](http://wiem.onet.pl), graficzny interfejs użytkownika to sposób komunikacji systemu operacyjnego (bądź uruchomionego w nim programu) z użytkownikiem<sup>4</sup>.

W innej definicji GUI jest elementem graficznym w programie lub systemie operacyjnym, pozwalającym na porozumiewanie się komputera z człowiekiem oraz prezentowanie wyniku działań programu za pomocą grafiki, piktogramów, rysunków, okienek. GUI zwykle wymaga myszy do wykonania poleceń<sup>5</sup>.

Na GUI składają się wszystkie elementy, które wspomagają człowieka w kontakcie z systemem operacyjnym: między innymi pulpit z ikonami, kierowany myszą kursor, okna dialogowe, funkcje systemu operacyjnego w rodzaju „przeciągnij i upuść”, etc<sup>6</sup>.

Z przytoczonych definicji wynika, że graficzny interfejs użytkownika jest rozumiany podobnie i ma powszechne odzwierciedlenie w praktyce komputerowej.

## 2. Metody tworzenia graficznego interfejsu użytkownika

Można wyróżnić dwa sposoby tworzenia wizualnego interfejsu użytkownika:

- programowanie w edytorze kodu,
- tworzenie kodu wg paradygmatu RAD.

Programowanie interfejsu użytkownika przez programowanie w edytorze kodu polega na opisie elementów interfejsu za pomocą składni danego języka programowania. O wiele szybszym i powszechniejszym sposobem jest tworzenie go według metodologii RAD (ang. *Rapid Application Development*), która jest technologią polegającą na udostępnieniu programiście dużych możliwości prototypowania i dużego zestawu gotowych komponentów (np. zapewniających gotowe obiekty wizualne czy dostęp do bazy)<sup>7</sup>.

---

<sup>3</sup> Por. R.M. Riordan: *Seeing Data: Designing User Interfaces for Database Systems Using*. NET, Addison Wesley Professional 2004, s. 14.

<sup>4</sup> [onet.pl/wiem](http://onet.pl/wiem), <http://portalwiedzy.onet.pl/80198,,,gui,haslo.html>.

<sup>5</sup> [algorytmy.pl](http://algorytmy.pl), <http://algorytmy.pl/encyklopedia/?id=256&t=graficzny+interfejs+u%BFytkownika>.

<sup>6</sup> [webstyle](http://www.webstyle.pl), [http://www.webstyle.pl/cms.php/en/netopedia/webdesign/graficzny\\_interfejs\\_uytkownika](http://www.webstyle.pl/cms.php/en/netopedia/webdesign/graficzny_interfejs_uytkownika).

<sup>7</sup> [wikipedia](http://pl.wikipedia.org/wiki/RAD), <http://pl.wikipedia.org/wiki/RAD>.

Według M. Manera, uzasadnienie wykorzystania RAD jest następujące<sup>8</sup>:

- a) na ogół 80% użyteczności rozwiązania może być stworzona przez 20% czasu potrzebnego na stworzenie całego rozwiązania;
- b) w wielu sytuacjach biznesowych wymagania systemu mogą być w pełni zaspokojone nawet wówczas, gdy niektóre z wymagań operacyjnych są niezadowolające;
- c) system można zaakceptować wówczas, gdy spełnia minimalny zestaw wymagań; system nie musi zatem spełniać wszystkich wymagań.

Warto się również przyrzeć zasadności stosowania RAD ze względu na następujące zagadnienia<sup>9</sup>:

- a) zakres projektu – dobrze zdefiniowane i znane zadania biznesu;
- b) dane do projektu – dostępna jest większość danych do projektu;
- c) decyzje w projekcie – są podejmowane przez relatywnie małą grupę osób;
- d) zespół projektowy – jest mały, liczy nie więcej niż sześć osób;
- e) architektura techniczna projektu – jest jasna i określona, a kluczowe komponenty technologii są dostępne i przetestowane;
- f) techniczne wymagania projektu – są uzasadnione i odpowiadają zaplanowanej technologii;
- g) szybka komunikacja z użytkownikiem końcowym<sup>10</sup>.

Wszystkie te cechy sprzyjają wykorzystaniu narzędzi RAD do tworzenia interfejsu użytkownika, ponieważ ze swojej definicji wspierają programistę, którego zadaniem jest tworzenie tego interfejsu.

### 3. Przegląd stosowanych narzędzi

Jest wiele narzędzi wspierających tworzenie interfejsu użytkownika. Biorąc pod uwagę historię ich rozwoju, początkowo wspierały jeden język programowania (np. w przypadku Delphi firmy Borland językiem tym był Object Pascal), by w końcu umożliwić tworzenie rozwiązań informatycznych przy użyciu tego samego interfejsu za pomocą różnych języków programowania. Przykładami takich środowisk są Microsoft Visual Studio i Borland Delphi Studio. Obydwa na-

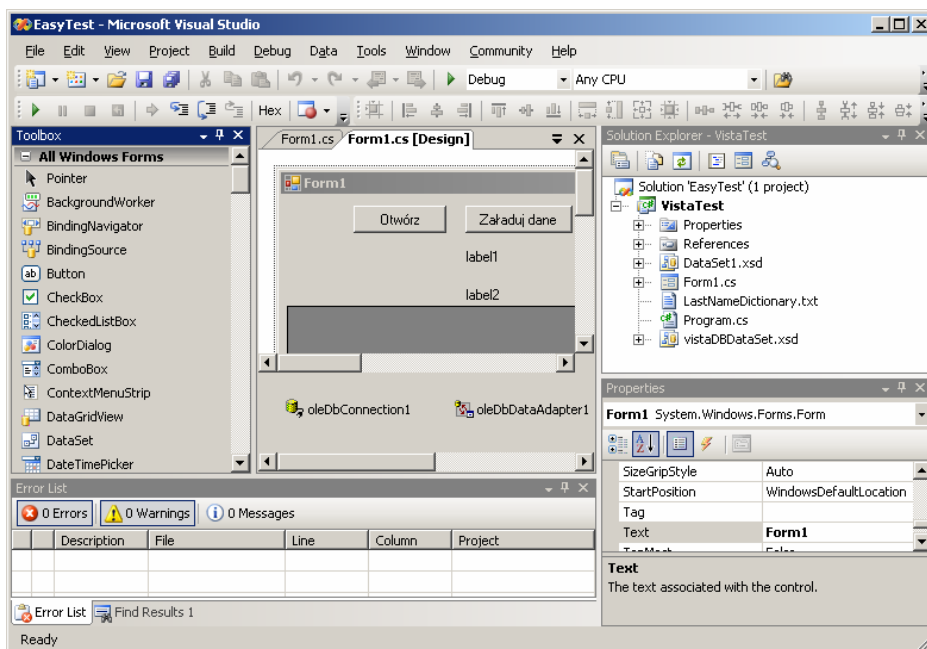
<sup>8</sup> Walter Maner, Rapid Application Development, <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>.

<sup>9</sup> Materiały gantthead.com, <http://www.gantthead.com/process/processMain.cfm?ID=2-19516-2>.

<sup>10</sup> Por. E. Larry: *Wood, User Interface Design: Bridging the Gap from User Requirements to Design*. CRC Press LLC 1997 s. 122.

rzędzia wspierają tworzenie programów dla technologii .NET (kod zarządzalny), natomiast narzędzie firmy Borland pozwala również na tworzenie aplikacji dla systemu Windows z kodem niezarządzalnym.

**Microsoft Visual Studio** to zintegrowany zestaw narzędzi programistycznych wspierającym tworzenie aplikacji w językach Visual Basic, Visual C#, Visual C++, Visual J+. Narzędzia te mają następujące funkcje: IntelliSense, edytor kodu, code snippets, wsparcie dla MS Office, kreator Windows Forms, kreator Web Forms, wsparcie urządzeń mobilnych, narzędzia do baz danych, Data Access Designers, Class Designer/Object Test Benach, edytor XML, deployment tools, zapisywanie, nagrywanie, uruchamianie makr, zapisywanie dodatków, instalowanie dodatkowych modułów, SQL Server Reporting Services/Crystal Reports, kontrola kodu, debugowanie lokalnie i zdalnie, kompilator 64-bit, server explorer, integracja z SQL Server 2005, profilowanie kodu, analizy statyczne, testowanie modułów, raportowanie kodu, zarządzanie projektem, zarządzanie przypadkami testowymi.

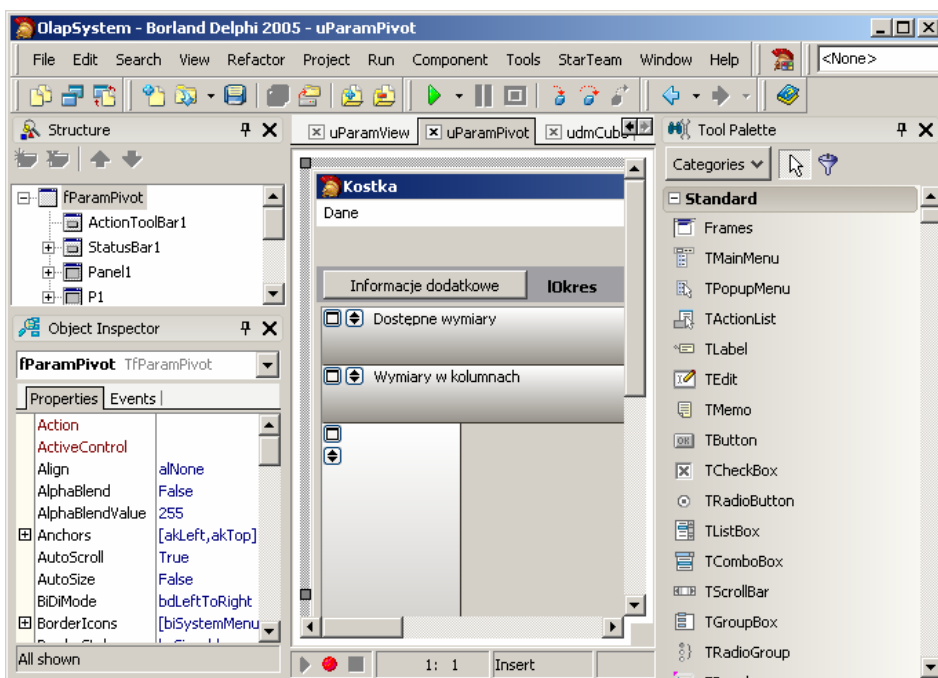


Rys. 1. Środowisko Microsoft Visual Studio

Źródło: opracowanie własne.

**Borland Delphi Studio** jest rozwijanym od jedenastu lat zestawem aplikacji, które w obecnej postaci pozwala na tworzenie programów komputerowych

w językach Object Pascal, C#, C++. Lista funkcji udostępnianych przez to narzędzie wygląda następująco: prowadnice z automatycznym wyrównywaniem do krawędzi tekstu i marginesów, komponenty VCL FlowPanel i GridPanel do projektowania formularzy z automatycznym pozycjonowaniem, podobnie jak HTML, technika „Live Templates” – przeskakiwanie między wybranymi punktami szablonów w kodzie źródłowym, inteligentne uzupełnianie bloków – przewiduje i poprawnie zamyka bloki kodu w trakcie jego powstawania, Database Explorer z obsługą baz danych przez sterowniki dbExpress metodą „przeciągnij i upuść”, maszyna stanów ECO programowana przez wizualne diagramy stanów, obsługa wielu baz danych, jednoczesne zachowywanie obiektów w wielu heterogenicznych bazach danych, w pełni natywna integracja środowiska z Borland StarTeam i Borland CaliberRM, wizualne tworzenie stron internetowych, wizualne tworzenie aplikacji internetowych, przechowywanie źródła HTML/ASP.NET, rozszerzony edytor stron HTML/ASP.NET, dwukierunkowe modelowanie klas „LiveSource”, obsługa diagramów UML i wzorców projektowych, zintegrowane audyty i metryki, tworzenie dokumentacji kodu i projektu.

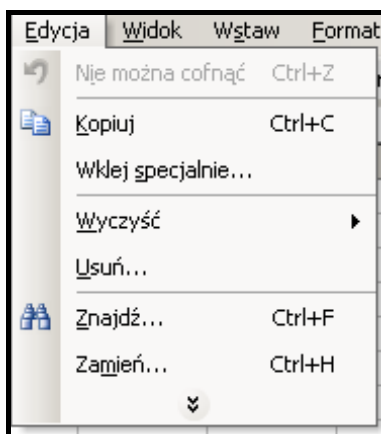


Rys. 2. Środowisko Borland Delphi

Źródło: opracowanie własne.

#### 4. Przykłady zastosowań


Jako przykłady wykorzystania GUI do kontaktu z użytkownikiem wybrano trzy programy, które należą do grup aplikacji biurowych, przeglądarek internetowych i systemów wspomagania decyzji OLAP. Opis dotyczy charakterystycznych widgetów, które decydują o przyjazności tych systemów dla użytkowników.



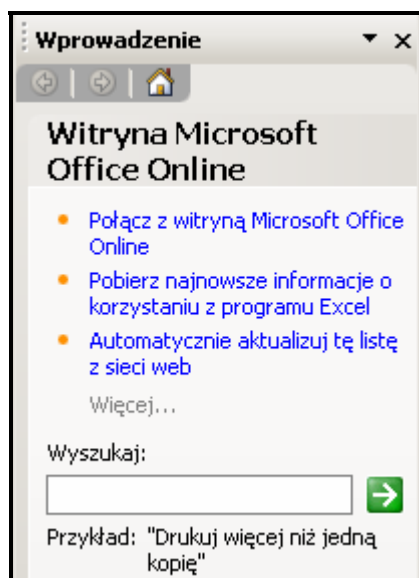
Rys. 3. Menu z najczęściej używanymi opcjami

Źródło: opracowanie własne.

#### Microsoft Office

Przykładem widgetu, który ma duży wpływ na komfort pracy z programami Office, jest menu zawierające tylko ostatnio używane opcje (rysunek 3). Dostęp do pozostałych opcji, rzadziej używanych, odbywa się poprzez przycisk .

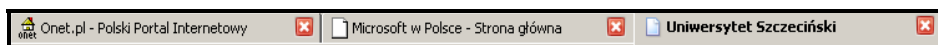
Kolejnym elementem graficznego interfejsu użytkownika jest tak zwany panel podręczny (rysunek 4), w którym mogą być umieszczone ważne funkcje z punktu widzenia pracy z danym programem.



Rys. 4. Panel podręczny  
Źródło: opracowanie własne.

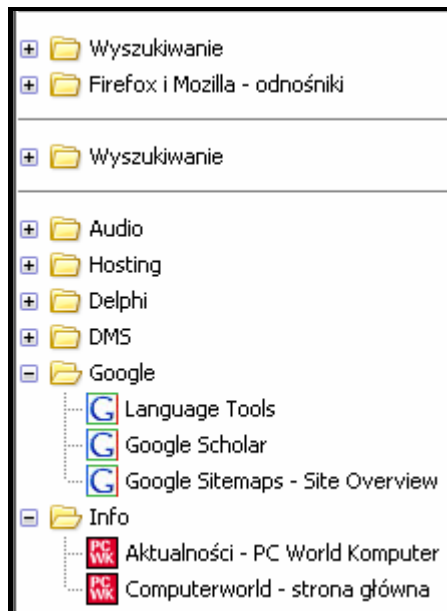
## Mozilla Firefox

W świecie przeglądarek internetowych można się również spotkać z licznymi udogodnieniami, których zadaniem jest ułatwienie pracy użytkownikowi wykorzystującemu taki program. Przykładem jest wykorzystanie zakładek, w których w jednym oknie przeglądarki można mieć otwarte różne strony internetowe (rysunek 5).



Rys. 5. Zakładki stron internetowych  
Źródło: opracowanie własne.

Następnym przykładem widgetu ułatwiającym dostęp do często odwiedzanych stron internetowych są struktury drzewiaste z zapisanymi odnośnikami do wybranych stron internetowych (rysunek 6).



Rys. 6. Struktura drzewiasta z odnośnikami  
Źródło: opracowanie własne.

### Rozwiązania OLAP na przykładzie XtraPivotGrid

Wiele ciekawych rozwiązań jest dostępnych w ramach oprogramowania wspomagającego podejmowanie decyzji. Przykładem takich rozwiązań są komponenty XtraPivotGrid, które mogą być używane w programach tworzonych za pomocą narzędzi programistycznych kompatybilnych z technologią NET.

Order Year ↑ ▾		Order Quarter ↑ ▾		
[-] 1995		1995 Total	[-] 1996	
Quarter 1	Quarter 2		Quarter 1	Quarter 2

Rys. 7. Nagłówek kolumny  
Źródło: opracowanie własne.

Ciekawym przykładem podejścia do wyglądu nagłówka kolumny jest widжет przedstawiony na rysunku 7, dostępny w opisywanym pakiecie. Za jego po-



można rozwinąć daną kolumnę w ten sposób, aby zostały uwidocznione szczegóły związane z pewnym abstraktem.

Innym elementem graficznego interfejsu użytkownika, obecnego również w oprogramowaniu wspierającym podejmowanie decyzji, są filtry (rysunek 8), za których pomocą można zawęzić wyświetlony zestaw danych do potrzebnych w danym momencie.



Rys. 8. Filtr

Źródło: opracowanie własne.

Z przedstawionych przykładów wynika, że elementy wykorzystywane w nowoczesnym graficznym interfejsie użytkownika są już zaawansowanymi technicznie obiektami, które niewątpliwie w dużym stopniu decydują o ergonomii oprogramowania. Celem takiego podejścia jest stworzenie magistrali łączącej użytkownika z komputerem i oprogramowaniem, która umożliwiłaby jak najbardziej intuicyjną obsługę systemu komputerowego.

## GRAPHICAL USER INTERFACE – METHODS AND TOOLS

### Summary

Graphical User Interface (GUI) it is highway splitting computer and software world with human world. It is important to swing to compatibility of those worlds. RAD

programming tools are supported quick application building with good-looking user interface. They can support in building different solutions, which are available in office systems, internet browsers, decision support systems and many others.

*Traslated by Grzegorz Wojarnik*