

Analiza doświadczalna złożoności symboli graficznych jako czynnika sterującego jakością pracy przy komputerze

Aleksy Patryn, Walery Susłow, Michał Statkiewicz

Wydział Elektroniki i Informatyki

Politechnika Koszalińska

ul. Śniadeckich 2, 75-453 Koszalin, Polska

Wstęp

Produktywność dialogu prowadzonego pomiędzy człowiekiem a komputerem za pośrednictwem graficznego interfejsu użytkownika (ang. Graphical User Interface, GUI) zależy między innymi od użyteczności zestawu obiektów graficznych, składających się na dany interfejs. W tym kontekście pojęcie „użyteczność” (ang. usability) daje możliwość całościowego ujęcia rozmaitych cech tych obiektów graficznych (budowy, kolorystyki, kontrastu) i aspektów ich interpretacji przez użytkownika komputera (pragmatyki). Projektanci, tworząc aplikacje komputerowe, ciągle poszukują nowych skuteczniejszych rozwiązań dla GUI w zakresie grafiki komputerowej. Grafika ta może być tworzona na bazie fotografii cyfrowej, skanowanych ilustracji bądź też od podstaw w edytorach graficznych (rysowanie) lub nawet w programach do generacji sztucznych form graficznych. Przydatność tych obiektów graficznych w kontekście danej aplikacji komputerowej istotnie zależy od poziomu intuicyjności metafory graficznej: przyszły użytkownik ma być w stanie szybko opanować nowe metafory [1] wprowadzone przez projektantów do przestrzeni wirtualnej i skutecznie manipulować programem za pośrednictwem kontrolki graficznych. Inaczej można powiedzieć, że sukces nowego zestawu grafik dla GUI (np. w postaci tradycyjnych ikon czy piktogramów) zależy od tego czy posiada on cechy łatwego do opanowania swego języka graficznego w ramach danej aplikacji.

Istnieją międzynarodowe i krajowe standardy (ISO/FDIS 9186-1, ISO 9241, PN-ISO 9186:2005), które wskazują projektantom sposób właściwego postępowania podczas tworzenia ergonomicznych symboli graficznych i piktogramów [2]. Standardy te, niestety, nie zawsze dają odpowiedzi na pytania praktyczne, ponieważ odzwierciedlają jedynie obecny na moment ich powstania poziom wiedzy w dziedzinie ergonomii. Zdaniem autorów, nie rozwiązano problemu rankingowania graficznych elementów GUI ze względu na ich użyteczność, brak jest ogólnie przyjętej skali do porównania przydatności elementów graficznych, które mają tworzyć wspomniany metaforyczny język interfejsu użytkownika. Trudność zdefiniowania takiej skali związana jest z niewystarczającą ilością badań złożonych procesów kognitywnych zachodzących podczas pracy użytkownika aplikacji komputerowej z GUI. Również

bezpośrednie przeniesienie modelu zjawisk zbadanych w sytuacjach realnych na przestrzenie wirtualne nie zawsze wydaje się być rozwiązaniem efektywnym.

Symbole graficzne i piktogramy można analizować jako obiekty składające się z samej grafiki oraz przypisanego jej znaczenia czy metafory. Procesy postrzegania takich symboli zależą, więc zarówno od cech grafiki zawartej w symbolach jak i ich pragmatyki. Najmniej zbadanym wydaje się być wpływ budowy samej grafiki, zawartej w symbolach – bez analizy przypisanego tej grafice znaczenia.

Autorzy niniejszego referatu, zajmując się badaniem alternatywnych inteligentnych graficznych interfejsów użytkownika [3, 4], proponują jako *jedno z kryteriów użyteczności elementów GUI poziom błędów użytkownika, rejestrowanych podczas wykonywania zadań rutynowych*. Tego typu zadania, zaplanowane w ramach prowadzonych badań mają strukturę testów psychodiagnostycznych komputerowych, takich jak test korektorski [5] czy test pamięciowy [3]. Seria zakończonych badań zawierała zastosowanie opracowanych przez autorów algorytmów i aplikacji testujących. Przewidywały one czynności krótkoterminowego zapamiętywania, odtwarzania i zliczania na materiale tradycyjnym dla takich pomiarów – symbolach alfanumerycznych, oraz na planszach złożonych z obiektów graficznych, nieznanymi osobom badanym z użytku codziennego. Obiekty te były tworzone w sposób sztuczny, algorytmicznie i interesowały badaczy jako formy (instancje, obiekty) graficzne, które podczas wykonania testu nie miały dla osób testowanych przypisanego (stałego, wyuczonego) sposobu interpretowania. Można powiedzieć, że dążono do tego, aby metafory tych obiektów nie istniały, aby nie można ich było zaliczyć do grupy symboli. Stwarzało to podczas testowania wymuszoną sytuację uczenia się tych nowych obiektów, bowiem wykonywane na tych obiektach czynności wymagały poprawnego ich rozpoznawania i zapamiętywania.

Metodologia badań

W badaniach użyteczności syntetycznych form graficznych zastosowano dwa typy testów.

Test korektorski przewidywał wizualne zliczanie obiektów graficznych (bez możliwości zliczania manualnego) w kolejnych wierszach planszy. Każdy wiersz planszy traktowany był jako odrębne zadanie korektorskie, dla którego rejestrowane były błędy w zliczaniu. Przebieg testu składał się z dziesięciu zadań. Błędy korektorskie popełniane podczas wykonania testu, podzielono na dwa rodzaje: pierwszy – to "*obce rozpoznano jako właściwe*", wyrażony prawdopodobieństwem fałszywej akceptacji P_{FAR} (False Acceptance Rate, FAR), drugi – to "*nieśluszenie odrzucono*", wyrażony prawdopodobieństwem fałszywego odrzucenia P_{FRR} (ang. False Rejection Rate, FRR). Zestawienie statystyk obu tych kategorii pokazało zdecydowanie mniejszy poziom prawdopodobieństwa błędów pierwszej kategorii: $P_{FAR} = 0,4\%$ symboli planszy.

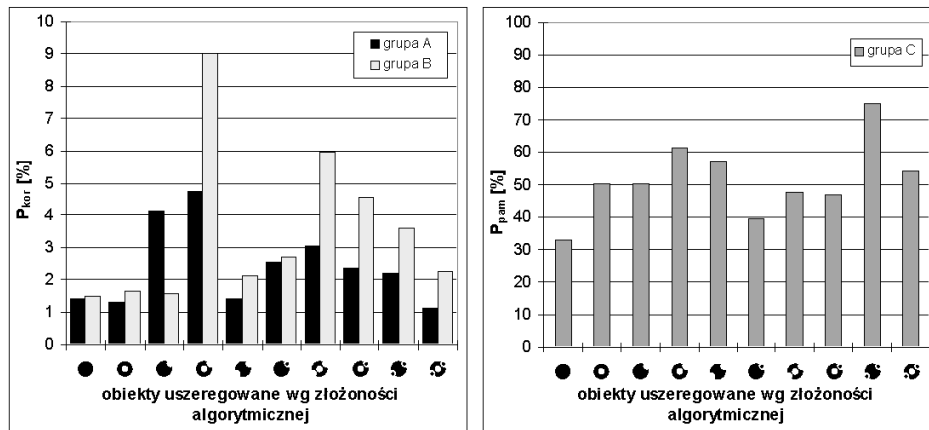
W zadanych warunkach testowania częściej popełniane były błędy, spowodowane tym, że przeoczono symbol występujący na planszy: $P_{FRR} = 1,22\%$ symboli planszy, co odpowiada około 3/4 całości zarejestrowanych błędów. W teście korektorskim można było spodziewać się zjawisk wywołanych zakłóceniami na etapie porównania wzorca poszukiwanych symboli (w pamięci operacyjnej) z zawartością pamięci ikonicznej. Od osoby testowanej nie wymagano określonego tempa pracy, ale proszono ich o precyzyjne zliczanie symboli.

Test pamięciowy przewidywał zapamiętywanie „słowa” złożonego z symboli graficznych oraz odtwarzanie tego słowa po pauzie, która gwarantowała wygaszenie śladu od zadania w pamięci ikonicznej. W stosunku do symboli tradycyjnych jak litery czy cyfry było to zadanie bardzo trudne, dlatego ilość obiektów do jednorazowego zapamiętywania zmniejszono do czterech. Błędy pamięciowe były wykrywane jako „podanie symbolu, którego nie było w słowie-zadaniu”, bądź też jako „niepodanie symbolu, który występował w słowie-zadaniu”. Zadaniem w teście pamięciowym było pojedyncze „słowo” do zapamiętywania, przebieg testu składał się z dziesięciu zadań. W teście pamięciowym spodziewaliśmy się zjawisk wywołanych zakłóceniami na etapie odtwarzania obiektów przechowywanych w pamięci operacyjnej. Tempo pracy osoby testowanej było w tej sytuacji wymuszone, co było dodatkowym czynnikiem powodującym błędy.

W teście korektorskim brały udział dwie trzydziestoosobowe grupy uczestników: uczennice liceum w wieku około 18 lat (grupa B) oraz studenci politechniki (mężczyźni), wiek około 22 lat (grupa A). W badaniach związanych z testem pamięciowym brała udział dwunastoosobowa grupa studentów (mężczyzn) w wieku około 20 lat (grupa C). Badania zorganizowane zostały w placówkach dydaktycznych: w Centrum Informatycznym Wydziału Elektroniki Politechniki Koszalińskiej oraz w Prywatnym Centrum Edukacyjnym w Koszalinie. W badaniach wykorzystywane były typowe stanowiska komputerowe, służące normalnie prowadzeniu zajęć. Autorskie oprogramowanie używane do badań pracowało pod kontrolą systemu Windows, umożliwiając automatyczny pomiar czasu oraz poprawności odpowiedzi użytkowników podczas testu i gromadzenie tych wyników razem z podstawowymi danymi osobowymi w bazie danych. Zostało wykazane, że standardowe urządzenia wejściowe współczesnych komputerów zapewniają praktycznie wystarczającą (1ms) dokładność pomiaru czasu [6].

Testowanie odbywało się w trybie jednoosobowym, ze względu na większą wiarygodność tak uzyskanych wyników. Grupy testowe najpierw zostały wstępnie przeszkolone – odbyła się demonstracja testu, następnie każdy uczestnik przystępował do testu pod kontrolą prowadzącego.

Diagramy występowania błędów



Rys. 1. Porównanie poziomu błędów P_{kor} , P_{pam} dla zbioru obiektów graficznych, testy korektorski (po lewej) i pamięciowy.

Średni poziom błędów korektorskich P_{kor} około 2-3% obrazuje niewielką trudność zadania dla układu poznawczego uczestników testu. Korelacja między wynikami grup wynosi 0,65, przy usunięciu ze zbioru „anomalnego” obiektu nr 3 (jedeny przypadek, gdy błędy grupy A przewyższają błędy grupy B) poziom korelacji osiąga 0,95. Tak duża różnica wprowadzana przez pojedynczy obiekt sugeruje istnienie znaczących różnic w jego postrzeganiu – z powodu odmiennych charakterystyk grup badanych.

Średni poziom błędów odtwarzania obiektów graficznych pamięci krótkotrwałej P_{pam} wynosi prawie 50%, czyli każdy drugi obiekt był odtwarzany błędnie. Można zauważyć tendencję wzrostu poziomu błędów ze wzrostem złożoności algorytmicznej obiektu. Poziom błędów w teście pamięciowym zależał także od położenia obiektu w słowie – obserwowano minimalny poziom dla pierwszego obiektu, i maksymalny poziom dla ostatniego czwartego obiektu (monotoniczna zależność od numeru pozycji w słowie).

W dużym przybliżeniu wyniki testu korektorskiego przedstawione na rys. 1 można opisać odcinkiem odwróconej paraboli, która jest opisywana wzorem:

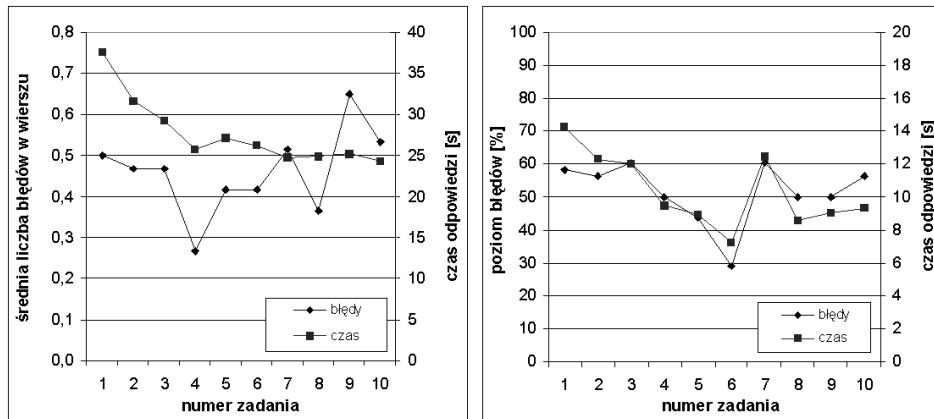
$$P_{kor} = f(\mathfrak{R}_A) = -0,057\mathfrak{R}_A^2 + 0,862\mathfrak{R}_A + 0,367 \quad (1)$$

Tendencja wzrostowa, zauważona w wynikach testu pamięciowego (patrz rys. 1) może być wyrażona liniową funkcją regresji:

$$P_{pam} = f(\mathfrak{R}_A) = 1,852\mathfrak{R}_A + 38,098 \quad (2)$$

Objaśnienia, co do wartości \mathfrak{R}_A są podane w niniejszym artykule poniżej.

Chronometria błędów



Rys. 2. Zmiana średniego poziomu błędów i czasu wykonania zadania dla zestawu badanych obiektów graficznych podczas wykonania poszczególnych zadań w testach korektorskim (po lewej) i pamięciowym.

W obu sytuacjach zauważamy występowanie zjawiska uczenia się czynności rutynowych, co objawia się tendencją zmniejszenia czasu potrzebnego na wykonanie kolejnych zadań, w teście korektorskim jest ona prawie monotoniczna. Współczynnik uczenia się w teście korektorskim (parametr logarytmicznej funkcji regresji) wynosi 5,43 – przy dopasowaniu do danych empirycznych na poziomie wartości współczynnika korelacji wielorakiej, $R^2 = 0,92$, co wskazuje na silny i stabilny wpływ procesu uczenia się zadania na przebieg testu. W teście pamięciowym współczynnik uczenia wynosi 2,24 przy dopasowaniu do danych $R^2 = 0,55$ – co świadczy o słabszym oraz mniej stabilnym zjawisku uczenia się. Może to być skutkiem zdecydowanie różnego poziomu komplikacji zadania dla umysłu osób testowanych, co odpowiada różnym ilościom informacji przepływającej między użytkownikiem a aplikacją testującą, jak zauważono we wcześniejszych badaniach [7]. W teście pamięciowym, ponadto zachodzi bardzo silna korelacja czasu odpowiedzi z poziomem błędów (wartość współczynnika korelacji 0,75), czego brak w danych z testu korektorskiego.

Sam pomiar czasu wykonywania elementarnych zadań w konkretnych warunkach może również dostarczać istotnych informacji, co jest wykorzystywane w metodzie projektowania interfejsów użytkownika KLM (ang. Key-stroke Level Model) [8]. Podczas naszych badań zauważyliśmy również występowanie korelacji między poziomem popełnianych błędów a czasem wykonywania zadania w teście pamięciowym. Zaobserwowane zjawisko wymaga dalszych badań.











Poziom błędów a złożoność obiektów graficznych

Do oceny złożoności obiektów graficznych zastosowaliśmy podejście Chaitina-Kolmogorowa [9], które wymaga znalezienia optymalnego (najkrótszego w danym języku) kodu algorytmu do generacji obiektów za pomocą maszyny liczącej (jak maszyna Turinga). To znaczy, że obiekty opisywane poprzez najdłuższe algorytmy uważamy za bardziej złożone (można stwierdzić, że takie obiekty charakteryzują się większą entropią). Jeśli umysł analizowałby grafikę podobnie do maszyny liczącej, to obiekty te wymagałyby większego nakładu pracy od układu poznawczego, a co za tym idzie zwiększałoby się prawdopodobieństwo popełniania błędów. Przy tym nie istotne jest, jak konkretnie umysł przetwarza formę pierwotną – analiza poznawcza może sprowadzać się do poszukiwania szeregu analogii do obiektów znanych oraz do kompresji upraszczającej, ale zawsze większa złożoność formy pierwotnej będzie wymuszała większe nakłady pracy.

W celu ułatwienia odnalezienia najkrótszego algorytmu generacji symboli dla naszego zestawu zdecydowaliśmy się na wprowadzenie własnej nieistniejącej, ale możliwej do realizacji maszyny graficznej-produkcyjnej. Może to być urządzenie typu drukarki, plotera czy wyrzynarki. W zasadzie, do sterowania taką maszyną wystarczy pseudojęzyk, który ma zdefiniowane metodę rysowania koła, np. za pomocą polecenia *draw_circle(int r, boolean is_black)*; oraz metodę ustawienia pozycji „głowicy” *move_center(int x, int y)*; . Wtedy za złożoność danego symbolu można przyjąć *minimalną ilość instrukcji maszyny graficznej, niezbędnych do powstania odpowiedniego obrazu.*











Do konkretyzacji idei wybrano wariant warstwowego natryskiwania czarnej lub białej farby na białą powierzchnię, cztery podstawowe polecenia dla takiej maszyny mają postać *paint()*, *switch_color()*, *reduce()*, *move(int x, int y)*. Istotnym założeniem jest, że maszyna ma rozpoczynać rysowanie zawsze od identycznych ustawień – pozycja głowicy w centrum rysowanego obiektu, rozmiar dyszy ustawiono na maksymalne (podstawowe) koło, kolor na czarny. Szczegółowo instrukcje do generacji obiektów i odpowiednia pozycja rankingowa na skali złożoności zostały pokazane w tabeli 1.

Tabela 1. Złożoność algorytmiczna generacji (produkcji) obiektu graficznego

Obiekt	Współczynnik złożoności \mathfrak{R}_A / Kod do produkcji obiektu graficznego
	1 paint()
	4 paint(), switch_color(), reduce(), paint()
	5 paint(), switch_color(), reduce(), move(x, y), paint()
	6 paint(), switch_color(), reduce(), paint(), move(x, y), paint()
	7 paint(), switch_color(), reduce(), move(x, y), paint(), move(x, y), paint()
	8 paint(), switch_color(), reduce(), move(x, y), paint(), reduce(), switch_color(), paint()
	8 paint(), switch_color(), reduce(), paint(), move(x, y), paint(), move(x, y), paint()
	9 paint(), switch_color(), reduce(), paint(), move(x, y), paint(), reduce(), switch_color(), paint()
	12 paint(), switch_color(), reduce(), move(x, y), paint(), move(x, y), paint(), switch_color(), reduce(), paint(), move(x, y), paint()
	13 paint(), switch_color(), reduce(), paint(), move(x, y), paint(), move(x, y), paint(), switch_color(), reduce(), paint(), move(x, y), paint()

Jako alternatywne do algorytmu produkcyjnego rozwiązanie przeanalizowaliśmy podejście intuicyjno-analityczne, przy którym integralny współczynnik złożoności optycznej obiektu graficznego obliczyliśmy metodą ekspercką poprzez wykrywanie binarnych elementów budowy, symetrii i samopodobieństwa w sylwetce (jak we fraktalach). Dotychczas wykonana analiza pozwala stwierdzić, że możliwa jest liniowa aproksymacja zależności poziomu błędów w teście korektorskim od wartości binarnych parametrów opisujących złożoność symboli graficznych. Wynika z tego, że da się posegregować symbole graficzne wg współczynnika trudności wyznaczonego poprzez zsumowanie wartości parametrów opisujących złożoność symboli graficznych z wykorzystaniem wag opartych na powyższej aproksymacji.

Tabela 2. Fenomenologiczna analiza złożoności badanych obiektów graficznych

Obiekt	Asymetria, tak/nie	Centrum, tak/nie	Wada koła, tak/nie	Intruz, tak/nie	Współczynnik złożoności \mathfrak{R}	Błędy dla testu korektorskiego, %	Błędy dla testu pamięciowego, %
	0	0	0	0	0,00	1.52	32,93
	0	1	0	0	0,10	1.64	50,56
	1	1	1	0	0,67	8.99	61,49
	1	1	1	1	0,93	4.56	47,04
	0	1	1	0	0,26	5.96	47,64
	0	1	1	1	0,52	2.26	54,31
	1	0	1	0	0,58	1.60	50,56
	1	0	1	1	0,83	2.72	39,50
	0	0	1	0	0,16	2.14	57,22
	0	0	1	1	0,42	3.59	74,92

Aproksymacja metodą statystyczną daje możliwość interpretacji uzyskanych danych poprzez następujący wzór:

$$\mathfrak{R}_E = 0.013*A + 0.012*I + 0.020*W + 0.014*C \quad (3)$$

Zmienne występujące w równaniach przyjmują wartości binarne i opisują takie cechy, jak: A – występowanie asymetrii obiektu, I – obecność intruza, wtrącenia na tle kształtu głównego, W – obecność wady obiektu w postaci ubytku oraz C – występowanie wyraźnie zaznaczonego centrum (patrz tabelę 2).

Podczas analizy statystycznej sprawdzono również alternatywną hipotezę: wagi oparte na kolejnych potęgach współczynnika złotego podziału (sectio aurea) dają znacząco wysokie dopasowanie powstałej zależności poziomym błędów od współczynnika trudności do krzywej regresji liniowej z uwzględnieniem współczynnika normalizującego $K = 0,159$. Wzór empiryczny do obliczania współczynnika złożoności przy akceptacji idei wykorzystania „złotej liczby” $Z=1.618$ może być zapisany następująco:

$$\mathfrak{R}_{ZP} = K (Z^2 * A + Z * I + Z^0 * W + Z^{-1} * C) \quad (4)$$

Podsumowanie

Wyniki przeprowadzonych badań potwierdzają istnienie korelacji między cechami konstrukcyjnymi form graficznych i poziomem błędów popełnianych przez użytkowników komputera analizujących te formy, a to znaczy poziomem obciążenia umysłu tymi formami. W obydwu przeprowadzonych testach udało się bowiem stwierdzić występowanie zależności między jakością wykonywania zadań testowych a złożonością algorytmiczną (także fenomenologiczną) obiektów. Należy wspomnieć również zaobserwowanie innych zjawisk, jak na przykład silnej korelacji czasu wykonywania zadań z poziomem błędów w kolejnych zadaniach w teście pamięciowym, czy istnienie „obiekty sporne” w teście korektorskim.

Uzyskane dane potwierdzają również zasadność wybranej metody, tj. praktyczną przydatność komputerowego badania jakości form graficznych za pomocą testów syntetycznych. Użyta metoda analizy pozwala również spodziewać się możliwości trafnego prognozowania jakości nowych form graficznych, poprzez analizę konstrukcji algorytmicznej lub cech fenomenologicznych tychże form, oraz użycie opisanych funkcji aproksymacyjnych.

Podziękowania

Autorzy aplikacji wyrażają podziękowania mgr Piotrowi Gackowskiemu za przeprowadzenie testu pamięciowego, mgr Beacie Reterskiej za udział w przeprowadzeniu testu korektorskiego, mgr Szymonowi Wanatowiczowi za zaprojektowanie zestawu badanych form graficznych.

Literatura

- [1] Spolsky J. „Projektowanie interfejsu użytkownika”, wyd. Mikom 2001
- [2] International Standards for Business, Government and Society, <http://www.iso.org>

- [3] Patryn A., Statkiewicz M., Susłow W. „Perspektywa optymalizacji aplikacji komputerowych dydaktycznych poprzez modelowanie układu uczeń-komputer” *Kognitywistyka i Media w Edukacji*, tom 7, nr 1 (2003), s. 49-64.
- [4] Statkiewicz M., Susłow W. „Tworzenie inteligentnych interfejsów użytkownika do aplikacji edukacyjnych jednym z aspektów respektowania osobowości ucznia” K. Pająk, A. Zduniak (red.) *Podmiotowość w edukacji ery globalnego społeczeństwa informacyjnego*, Wydawnictwo Dom Wydawniczy ELIPSA, Warszawa - Poznań 2004, tom 3, s. 96-104.
- [5] Susłow W., Statkiewicz M., Reterska B., Wanatowicz S.(2005) „Metoda korektorska weryfikacji użyteczności piktogramów w interfejsie graficznym aplikacji e-learningowych” W: W. Tarnowski (red.): *I Konferencja Nowe Technologie w Kształceniu na Odległość*, Koszalin-Osieki, 9-11 czerwiec 2005, Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin 2005, s. 234-241.
- [6] Patryn A., Susłow W., Statkiewicz M., Ileczo G. „Problem chronometrii zdarzeń w układzie operator-komputer”, w materiałach konferencyjnych *IX Poznańskich Warsztatów Telekomunikacyjnych*, Poznań 12.2004, s. 51-55.
- [7] Patryn A., Susłow W., Statkiewicz M., Kadowski K. „Metoda psychodiagnostyczna oceny interfejsu graficznego użytkownika: zróżnicowanie przepływu informacji w układzie człowiek-komputer”, *Kognitywistyka i media w edukacji*, nr 1-2 (2006), Toruń
- [8] Kieras D. „Using the Keystroke-Level Model to Estimate Execution Times” 2001, <ftp://www.eecs.umich.edu/people/kieras/GOMS/KLM.pdf>
- [9] Knuth Donald E. „The Art of Computer Programming. Vol. 2: Seminumerical Algorithms” 3rd ed. by Reading, MA: Addison-Wesley, 1997.

Experimental analysis of the complexity of the graphic symbols as a factor controlling the quality of the work at the computer