

Jacek DŁUGOSZ

Politechnika Śląska, Instytut Informatyki

## TWORZENIE APLIKACJI MULTIMEDIALNYCH DLA OSÓB Z UPOŚLEDZENIEM WZROKOWYM

**Streszczenie.** W niniejszym artykule przedstawiono problemy związane z adaptacją i z tworzeniem aplikacji multimedialnych na potrzeby osób niewidomych. Artykuł powstał w wyniku prac prowadzonych w ramach projektu celowego realizowanego przez Instytut Informatyki Politechniki Śląskiej we współpracy z Ośrodkiem dla Dzieci Niewidomych w Laskach. W ramach projektu analizowano problemy związane zarówno z oprogramowaniem stacjonarnym, jak i sieciowym.

## THE RULES FOR BUILDING MULTIMEDIA APPLICATIONS FOR BLIND USERS

**Summary.** In this article has been presented problems which may be occurred during adaptation and creation the software for using by blind people. Article is one of results of scientific project, which was carried out by Institute of Computer Science of Silesian University of Technology in Gliwice and Centre for Blind Children in Laski. Among our interests was analysis and evaluation stationary and network software.

### 1. Wstęp

Współcześnie coraz więcej uwagi poświęca się osobom niepełnosprawnym, na stwarzanie im równych szans w życiu, na tworzenie przyjaznych budynków, urzędów, dróg. Coraz większą rolę w życiu człowieka zaczyna odgrywać komputer, staje się narzędziem, bez którego nie można obyć się praktycznie w żadnej kategorii pracy. Coraz powszechniej stosowany jest przez osoby niepełnosprawne, w tym niepełnosprawne wzrokowo. Paradoksalnie komputer osobisty, który stanowić ma pomoc dla człowieka na każdym kroku, często w przypadku osób niepełnosprawnych wzrokowo staje się utrudnieniem w pracy.

Wynika to z prostego faktu, że dysponujemy coraz lepszym, coraz bardziej efektywnym sprzętem, a producenci oprogramowania starają się maksymalnie wykorzystać jego potencjalne możliwości. Współczesne oprogramowanie pisane jest w sposób nieefektywny i w nadmierny sposób obciąża procesor i pamięć (za co winą z jednej strony można obarczyć programistów, z drugiej zaś nieefektywne kompilatory), przez co aplikacje mogą działać w sposób zbyt wolny po ich integracji z oprogramowaniem wspomagającym osoby niewidome. Dodatkowo oprogramowanie często ma wbudowany szereg atrakcyjnych elementów, dzięki którym program lepiej wygląda i lepiej się sprzedaje, a trudniej jest taki produkt zaadoptować na potrzeby osób niewidomych.

Starano się stworzyć wytyczne dla producentów oprogramowania multimedialnego, takich by tworzone przez nich produkty mogły być swobodnie używane przez niewidomych. Nie uzurpuje się przy tym prawa do wymuszenia na producentach poprawienia obecnie sprzedawanych aplikacji, lecz jedynie mamy nadzieję, że wezmą nasze sugestie pod uwagę przy realizacji kolejnych produktów.

## 2. Rodzaje dostępności do oprogramowania

Rozwiązanie zagadnienia dostępności oprogramowania sprowadza się do zastosowania jednego z dwóch podejść - dostępności bezpośredniej i niebezpośredniej, których definicje zamieszczono poniżej. By określić szansę uczynienia produktu dostępnym, należy znać kilka cech charakteryzujących jego potencjalnych użytkowników – przede wszystkim musimy wiedzieć, do jakiej grupy wiekowej produkt ten zostanie skierowany, na jakie ograniczenia sprzętowe napotka się w fazie projektowania oprogramowania, jakie cele i jakie informacje ma przekazać program. Nasza dyskusja dotyczy oprogramowania edukacyjnego.

Dostępność bezpośrednia produktu umożliwia i zapewnia niewidomemu możliwość pracy na wszystkich widocznych na ekranie kontrolkach i obiektach. Ponadto może on pracować z oprogramowaniem bez potrzeby korzystania ze specjalizowanych aplikacji typu screen reader (opisanych w kolejnym rozdziale). Należy brać pod uwagę fakt, że zwłaszcza dzieci nie są zapoznane z technologiami asystującymi, które są dość skomplikowane w obsłudze. Dlatego zwłaszcza przy projektowaniu aplikacji skierowanych do dzieci warto, o ile jest to możliwe, zaopatrzyć program w rozwiązania zapewniające dostępność bezpośrednią. Produkt zapewniający dostępność w sposób bezpośredni powinien umożliwiać współpracę z klawiaturą, oraz informować zarówno o wynikach działania programu, jak i kolejnych krokach pracy z programem przy użyciu dźwięku. Dźwięk powinien obwieszczać obecność i



informować o stanie wszystkich kontrolek dostępnych na ekranie i przekazywać „atmosferę” oprogramowania.

Dostarczenie bezpośredniej dostępności zapewnia również szereg innych korzyści. Najważniejszą jest to, że użytkownik może uzyskać dostęp do aplikacji bez potrzeby korzystania ze specjalnego oprogramowania bądź sprzętu asystującego. Fakt ten znacznie zmniejsza koszty zakupu oprogramowania czy sprzętu. Ponadto pozwala to, by niewidomi korzystali z jakiegokolwiek komputera, a nie tylko ze specjalnie zaadaptowanego do ich potrzeb. W końcu tworzenie dostępnego bezpośrednio interfejsu przez twórcę oprogramowania może zaowocować lepszym końcowym efektem, gdyż projektanci lepiej pojmują znaczenie i istotę tworzonych przez siebie kontrolek, niż interpretuje je oprogramowanie asystujące [5].

W przypadku dostępności niebezpośredniej fragmenty oprogramowania projektowane są w taki sposób, by sprawnie współpracowały z programem typu screen readerem. Ten poziom dostępności zakłada, że użytkownik ma zainstalowane preferowane przez siebie oprogramowanie asystujące i potrafi je w miarę dobrze obsługiwać. Oprogramowanie powinno być projektowane z wykorzystaniem pewnych zasad, co powinno ułatwić jego komunikację ze screen readerem. Usprawnienia te mogą być zaimplementowane przez programistów przy użyciu takich narzędzi, jak Microsoft Active Accessibility (MSAA) ([3]) czy Java access API ([6]). Dodatkowo, by tworzone oprogramowanie mogło być dostępne w sposób niebezpośredni, należy stosować standardowe kursory i czcionki.

Dostępność niebezpośrednia również może mieć pewne korzyści. Może dla przykładu wymusić, a co za tym idzie zapewnić konsekwencje w tworzeniu oprogramowania – narzuca pewne standardy. W pewnych przypadkach może być taniej stworzyć oprogramowanie tą drogą, chociażby z oszczędności przestrzeni dyskowej. Dodatkowo zastosowanie screen readerów zapewnia w najprostszym możliwym sposób obsługę linijek braillofskich, gdyż dostęp do nich jest zazwyczaj mało znany programistom i realizowany w dość szczególny sposób. Dodatkowym zyskiem przy tworzeniu aplikacji zgodnych ze screen readerami jest to, że program taki prawdopodobnie będzie równie dobrze współpracował z innymi technologiami asystującymi, włączając w to urządzenia skanujące, klawiatury ekranowe, systemy poleceń głosowych, dostosowujących klawiaturę dla specyficznych potrzeb użytkownika, oprogramowanie przewidujące wprowadzane z klawiatury wyrazy.

Ważne jest, by odnotować, że nie każdy produkt można uczynić dostępnym. Niektórych rzeczy po prostu nie da się przekazać osobom niewidomym, zwłaszcza informacji o kolorach.

Wielu niewidomych ma pozostałości wzroku. Może ona przybierać różne postacie począwszy od postrzegania światła, a skończywszy nawet na widzeniu bardzo powiększonych obrazów. Najlepszym podejściem przy projektowaniu aplikacji dla tej grupy osób jest



założenie, że takiej zdolności nie posiadają (traktowanie ich jak osoby zupełnie pozbawione wzroku), ale nie uniemożliwiać im korzystania ze swoich zdolności [2].

### **3. Techniki umożliwiające dostęp do komputera przez osoby niewidome**

Dla osób, które nie są w stanie odczytać zawartości ekranu (dla osób niewidomych), wymagane są pewne mechanizmy prezentacji informacji w formie niewizualnej. Istnieją dwa podstawowe takie mechanizmy - mowa i alfabet Braille'a.

Screen readersy – oprogramowanie dla osób niewidomych, które wyszukuje na ekranie monitora komputera pewne informacje i przekazuje je korzystającej z komputera osobie niewidomej w postaci głosowej. Screen reader nie jest syntezerem mowy, wykorzystuje w tym celu gotowe oprogramowanie innych producentów bądź urządzenia sprzętowe zamieniające tekst (ciąg znaków alfanumerycznych) na mowę. Oprócz dźwiękowych urządzeń wyjściowych mogą być również stosowane urządzenia braillofskie. Jako że alfabet Braille'a jest alfabetem dotykowym, może być używany jako uzupełnienie mowy, by przedstawiać informację użytkownikowi. Specjalne linijki braillofskie posiadają 20 bądź 40 znaków z elektromechanicznymi pinami, które tworzą dynamiczny braillofski wyświetlacz, którego zawartość może być nieustannie i dynamicznie zmieniana przez komputer. Jako wynik - wszystko co jest drukowane w postaci alfanumerycznej bądź jest opisane w postaci mowy może być przedstawione za pomocą linijki braillofskiej. Jest to bardzo efektywny i preferowany sposób dostępu do informacji tekstowej przez osoby niewidome. Należy przy tym jednak zauważyć, iż istnieje duży odsetek niewidomych, którzy nie znają Braille'a, a co za tym idzie nie są w stanie korzystać z linijek. W związku z powyższym nie można nigdy założyć, że korzystanie z linijek braillofskich będzie jedyną drogą dostępu do informacji zawartej na ekranie monitora przez niewidomych.

Screen readersy zazwyczaj projektowane są tak, by pobierały informacje z ekranu, stosując standardowe procedury, metody, kontrolki systemu operacyjnego. Oprogramowanie stosujące standardowe procedury dostępu do ekranu może być w prosty sposób przysposobione na potrzeby screen readera. W sytuacji gdy wykorzystuje własne, nietypowe metody, wówczas dostęp do takiego oprogramowania może być bardzo trudny, a często wręcz niemożliwy. Screen readersy pozwalają użytkownikom przemieszczać się po ekranie i odczytywać każdy tekst, który jest na nim zapisany. W środowiskach graficznych, wielooknowych screen readersy muszą ponadto być w stanie umożliwić użytkownikom przemieszczanie się między oknami oraz po innych elementach ekranu, takich jak suwaki, kontrolki wielkości ustawiania okna,

przyciski radio i inne. Ponadto muszą zapewniać możliwość dostępu do takich elementów graficznych, jak ikony czy paski narzędzi. Dla stereotypowych obrazów, które zawsze wyglądają tak samo (standardowe obiekty graficzne okien dialogowych), mogą zostać przypisane nazwy. Kiedy użytkownik przemieszczając się po ekranie napotka na swej drodze taki właśnie element graficzny, wówczas usłyszy odpowiedni komunikat.

Screen readersy zdolne są dostarczać informacji do podstawowych elementów systemu operacyjnego, takich jak okna, menu, okna dialogowe etc., jak również do tekstu wyświetlanego w oknach aplikacji (pod warunkiem ich wyświetlania przy użyciu standardowych poleceń i obraz jest obrazem tekstowym, a nie graficznym).

Dużo bardziej skomplikowana sytuacja jest w przypadku rysunków, schematów, wykresów. Rozwijane są różne strategie dostępu, ale chyba żadna nie jest doskonała. Jedno z podejść polega na wykorzystaniu wirtualnych tabletek z drobnymi pinami, które obrazują zawartość monitora znajdującą się w najbliższym otoczeniu kursora myszy. Podejmowane są również próby rozpoznawania i opisywania najbardziej typowych elementów przedstawianych w postaci graficznej - takich jak wykresy.

Dodatkowo oprócz problemów z dostępem do zawartości ekranu monitora, niewidomi użytkownicy mają również problemy w korzystaniu z urządzeń wejściowych, których obsługa wymaga wzroku. Dla przykładu klawiatury posiadają klawisze NumLock, ScrollLock, CapsLock - które zatrzaszkują się w jednej z pozycji on/off. O ich aktualnym stanie informują użytkownika świecące się bądź zgaszone kontrolki. Oczywisty jest fakt, że niewidomi tych światełek nie zobaczą i przydatność tych kontrolki dla nich jest praktycznie żadna. Stan tych klawiszy screen reader rozpoznaje po wartościach znaczników znajdujących się w odpowiednich rejestrach klawiatury. Ważne jest, by aplikacje stosowały standardowe metody obsługi klawiatury, a nie tworzyły własne programowe rejestry stanu tych klawiszy.

Dużo bardziej poważny problem dla osób niewidomych stanowią programy, które wymagają korzystania z myszy. Mysz stanowi niemal oczywiste narzędzie obsługi komputera. Twórcy scen readerów starają się ułatwić posługiwanie się tym urządzeniem. Jedną z strategii polega na odczytywaniu zawartości ekranu znajdującej się pod myszą. Inna polega na wykorzystaniu touchpadów - wówczas niewidomy może mniej więcej wiedzieć, gdzie znajduje się kursor myszy. Również ciekawym rozwiązaniem jest stosowanie ekranów dotykowych.



#### 4. Zakres badań

W trakcie prac zwracano uwagę szereg elementów funkcjonalnych oprogramowania, do których dostęp może być dla niewidomych utrudniony, a także na fakt występowania w programie elementów mogących ułatwić nawigację.

I tak, zwrócono uwagę na łatwość instalacji produktu – z jednej strony na dostęp do informacji na ekranie za pomocą screen readera, z drugiej zaś na ewentualne interferencje instalowanego programu ze screen readerem (wyjątki krytyczne itp.).

Drugim przebadanym aspektem była swoboda pracy w aplikacji. Oczywiście dla badających było, iż liczba napotkanych problemów będzie tym większa, im bardziej skomplikowany i odbiegający od standardowego będzie interfejs badanego programu. Wprowadzenie własnego sposobu wyświetlania informacji w oknie powoduje, że programista może zapomnieć o istotnych rzeczach załatwianych za niego w normalnym przypadku przez kompilator.

I tak rzeczywiście było w przypadku testowanych aplikacji. Nawigacja w szeregu programach była bardzo utrudniona, a często wręcz niemożliwa. Pomimo tych faktów starano się znaleźć nawet w praktycznie nie współpracujących ze screen readerami programach dobre strony. Nie zawsze było to jednak możliwe.

W trakcie badań starano się używać (badać) maksymalnie dużej liczby funkcji oferowanych przez poszczególne aplikacje. W przypadku części programów było to jednak niemożliwe, gdyż nawet osoba widząca musiałaby poświęcić kilka dni na opanowanie podstaw nawigacji.

Po zapoznaniu się z dokumentacją screen readerów stwierdzono, że część przebadanych aplikacji prawdopodobnie można „zmusić” do poprawnej pracy ze screen readerami. Proces ten jest jednak zazwyczaj bardzo czasochłonny i wymaga nie tylko biegłej znajomości obsługi screen readera bądź nawet pisania makroprogramów (o ile screen reader oferuje tę metodę obsługi oprogramowania), ale również bycia osobą widzącą. W sytuacjach spotykanych na co dzień biegłe posługującymi się screen readerami są zazwyczaj osoby niewidome (o ile są), które korzystają z pomocy osób widzących przy konfiguracji oprogramowania.

W testach zwracano uwagę na fakt, czy są potrzebne specjalne zabiegi do poprawnego funkcjonowania aplikacji, a jeżeli tak, to czy istnieją szanse na ich skuteczne zastosowanie. Ze względu na wspomnianą wcześniej czasochłonność – spowodowaną indywidualnym podejściem do każdego problemu w przypadku każdej pary screen reader – aplikacja multimedialna, a także faktem stosunkowo krótkiego czasu obcowania ze screen readerem, a co za tym idzie niezbyt dużego doświadczenia, próby zintegrowania programów zostaną podjęte w drugiej fazie projektu.



Istotną cechą programu dla niewidomych jest możliwość korzystania ze skrótów klawiaturowych. Dużo aplikacji zakłada jednak, że będzie używana wyłącznie przy zastosowaniu myszy. Często nie ma dostępnego nawet przycisku przenoszenia fokusu po elementach okna.

W trakcie badań zwrócono również uwagę na z pozoru drobny fakt, ale dla niewidomych bardzo istotny – obecnie często spotyka się aplikacje, które po uruchomieniu witają użytkownika kolorową planszą, w obrębie której (w dowolnym miejscu lub ściśle określonym) należy kliknąć myszą, by przejść do właściwego modułu aplikacji.

Ostatnim przebadanym elementem programów był dostęp do multimediiów – czy niewidomy jest w stanie odtworzyć dołączane pliki dźwiękowe lub filmy, czy znajdują się podpisy pod obrazkami, na podstawie których może dowiedzieć się, co się na nich znajduje.

W większości przypadków stwierdzono, że screen readersy nie są w stanie poprawnie obsłużyć polskich wydawnictw multimedialnych. Spowodowane jest to nie słabą jakością screen readerów, lecz stosowaniem przez wydawnictwa multimedialne niestandardowych interfejsów. Fakt ten wynika prawdopodobnie z chęci napisania jak najbardziej przyciągającego wyglądem programu. W marketingu liczy się pierwsze wrażenie wywarne na kliencie. Ładny, lecz często niefunkcjonalny, interfejs może w prosty sposób znacznie zwiększyć obroty firmy. Głównymi problemami, jakie napotkano, były – brak możliwości poruszania się po obiektach okna za pomocą przycisku TAB, przyciski w postaci graficznej, brak skrótów klawiaturowych, tekst wyświetlany jako grafika, interferencja między screen readerem a lektorem programów multimedialnych.

## 5. Zalecenia dla projektowania aplikacji

Istnieje szereg elementów, na które powinni zwrócić uwagę projektanci i twórcy aplikacji, by użytkownicy korzystający ze screen readerów byli w stanie sprawnie posługiwać się ich produktami, by byli w stanie odnaleźć i zidentyfikować obiekty znajdujące się na ekranie. Do rzeczy tych należą:

- stosowanie systemowych narzędzi wszędzie tam, gdzie tylko jest to możliwe, do wyświetlania i usuwania tekstu, a także do wyświetlania kursorów,
- używanie standardowych kontrolek wszędzie tam, gdzie jest to tylko możliwe,
- rysowanie narzędzi w paskach narzędzi, paletach i menu jako osobne obiekty, nie jako jedną zbiorczą grafikę. Dzięki temu możliwe staje się zidentyfikowanie kształtu, pozycji, wyglądu narzędzia, a następnie jego nazwania i późniejszego zidentyfikowania w trakcie pracy.

Kompatybilność oprogramowania ze screen readerami może być jeszcze rozbudowana o:

- stosowanie specjalnych technik w celu przekazania screen-readerowi czy tekst jest osadzony w obrazie graficznym,
- automatyczne przenoszenie kursora przy zmianie fokusu bądź przy podświetlaniu,
- stosowanie stałych bądź przewidywalnych układów w oknach i dialogach,
- unikanie stosowania „dymków” pomocy, które znikają przy zmianie fokusu, chyba że istnieje sposób zablokowania ich, tak by mogły zostać odczytane np. poprzez przemieszczenie kursora (zmiana fokusu),
- używanie jednokolumnowego tekstu,
- nadawanie kontrolkom logicznych i sensownych nazw nawet wtedy, gdy nie są one widoczne na ekranie,
- zapewnienie dostępu z klawiatury do wszystkich narzędzi, menu i okien dialogowych, bez konieczności używania myszy. Można również pokusić się o wbudowanie systemu rozpoznawania poleceń głosowych.

Ponieważ screen readery potrafią jedynie czytać tekst bądź nadawać nazwy separowalnym i identyfikowalnym ikonom i narzędziom, wskazane jest, by:

- unikanie stosowania niezaetykietowanych hot-spotów w postaci obrazków (chyba że istnieje równocześnie dostęp z menu),
- unikanie pozycji w menu nie zawierających tekstu bądź widoczne, lub niewidoczne wskazówki,
- unikanie powtarzających się elementów graficznych w paskach narzędzi.

I w końcu:

- cała dokumentacja i pomoc on-line powinna być tak zaprojektowana, by mogła być całkowicie zrozumiała jedynie po przeczytaniu samego tekstu (dla przykładu rysunki i grafiki powinny mieć opis w postaci tekstu),
- zsynchronizowana ścieżka dźwiękowa uruchamiana wraz ze wszelkimi prezentacjami i filmami,
- powinny być stosowane standardowe systemy kodowania informacji, takie jak HTML i XML dla tekstu, GIF i JPEG dla obrazów, WAVE, QuickTime, AVI, MPEG dla dźwięku i obrazu [1],
- oprogramowanie powinno umożliwiać zmniejszenie szybkości prezentowania informacji, a także jej zatrzymania.

Ostatecznie, dostęp do oprogramowania:

- znacznie powinna ułatwić dokumentacja dostępna w postaci elektronicznej, w formie dostępnej dla screen readerów oraz



- personel biura obsługi klienta powinien zdawać sobie sprawę z ułomności fizycznej części swoich klientów, powinien wiedzieć, na jakie problemy może natrafić i jak te problemy rozwiązywać.

Innym zagadnieniem, które powinno być rozwiązywane w trakcie tworzenia aplikacji, jest tzw. opis dźwiękowy. Zapewnia on dostęp niewidomym do multimediiów poprzez dodanie ścieżki z lektorem opisującym to, co dzieje się na wizualnej prezentacji włączając w to akcje, zmiany zachodzące na scenie, grafikę i tekst na scenie.

Istnieje szereg formatów zapewniających takie rozwiązanie – Synchronized Multimedia Integration Language (SMIL), Synchronized Accessible Media Interchange (SAMI) oraz QuickTime. We wszystkich trzech formatach użytkownik może wybrać podpisy, pod warunkiem że twórca aplikacji je dostarczył. Dodatkowo w QuickTime'ie użytkownik może włączyć lub wyłączyć opis dźwiękowy. Nie ma takiej możliwości w SMIL 1.0, lecz ma być udostępniona w nowej wersji 2.0. SMIL został stworzony przez World Wide Web Consortium (W3C) międzynarodowe konsorcjum publikujące internetowe protokoły. Multimedialna prezentacja SMIL składa się z kilku elementów: dźwięku, obrazu, tekstu, które przechowywane są oddzielnie i mogą być z sobą synchronizowane w trakcie prezentacji. Prezentacje SMIL mogą być rozpowszechniane w Internecie, a także na płytach CD-ROM i DVD-ROM. Jednym z popularniejszych narzędzi potrafiących odtwarzać klipy SMIL jest Real Player firmy RealNetworks w wersji G2 lub nowszej.

SAMI jest ogólnie dostępną specyfikacją firmy Microsoft, pozwalającą na dołączanie podpisów w Media Playerze. Podpisy mogą być włączane i wyłączane przez użytkownika wówczas, gdy uruchamiana jest w odrębnym oknie. Gdy prezentacja odtwarzana jest z innej aplikacji (np. Encarta), wówczas ta aplikacja musi udostępniać odpowiednie narzędzia potrafiące wykorzystać funkcje SAMI.

W QuickTime'ie podpisy dołączane są jako ścieżka tekstowa, a opis zapisywany w oddzielnej ścieżce dźwiękowej, która może być odtwarzana obok standardowych ścieżek wideo i dźwięku. Jedynie wersja Pro umożliwia użytkownikowi włączanie i wyłączanie ścieżki podpisów i opisu dźwiękowego.

Darmowe narzędzie Media Access Generator (MAGpie) umożliwia producentom oprogramowania nadawanie cech dostępności multimedialnym klipom. MAGpie może być stosowany w celu tworzenia i dodawania opisów tekstowych we wszystkich trzech opisanych wcześniej formatach, a także do dodawania opisu dźwiękowego do prezentacji SMIL.

Obrazy często zawierają bardzo istotne informacje. Bez przekazu dźwiękowego niewidomi użytkownicy nie są w stanie skorzystać z tej informacji. Stworzenie w pełni funkcjonalnego opisu dźwiękowego wymaga dużego doświadczenia i wycucia, jak najlepiej werbalnie

przekazać zawartość obrazu. Narracja powinna być precyzyjnie tworzona, tak by umieszczana była w naturalnych przerwach w ścieżce dźwiękowej programu, by uniknąć interferencji.

QuickTime Pro umożliwia dodawanie ścieżki z opisem dźwiękowym. W przypadku SMIL najlepszym narzędziem służącym do tego celu jest MAGpie.

## 6. Aplikacje sieciowe – wybrane mechanizmy

### 6.1. Java

Java stanowi atrakcyjne środowisko programistyczne dla celów tworzenia aplikacji multimedialnych dostępnych dla osób niewidomych. Atrakcyjność tę można uzasadnić kilkoma cechami charakteryzującymi Javę:

- niezależny od platformy językowej i systemowej język programowania aplikacji,
- wbudowane w jądro Javy mechanizmy wspierające ideę dostępności – w tym obsługa interfejsu ekranowego i efektywne możliwości obsługi klawiatury,
- API Javy dostarcza szeregu dodatkowych właściwości obiektów, które ułatwiają programiście określenie obecności i stanu obiektu na ekranie – w tym „dostępna nazwa” i „dostępny opis”.

Dostępna nazwa stanowi zwięzłe wyjaśnienie znaczenia obiektu. Użytkownik nawigując wśród graficznego interfejsu napotyka na wiele obiektów i opis każdego z nich składa się zazwyczaj z jednego bądź dwóch słów, dzięki czemu unika się szumu informacyjnego w trakcie pisania programu. Dla większości typowych obiektów (pole tekstowe, menu, przycisk, przycisk radio, elementy listy) nazwy te nadawane są automatycznie na podstawie statycznego tekstu występującego w jego opisie. Obiektom, które nie posiadają etykiety, programista musi nadawać nazwy dostępne własnoręcznie.

Dostępny opis stanowi bardziej rozbudowane wyjaśnienie charakteru i celu umieszczenia obiektu na ekranie i powinien być stosowany wtedy, kiedy zamieszczenie takiej dodatkowej informacji może być użyteczne. Dostępny opis tworzony jest automatycznie w trakcie przypisywania tooltipsów do obiektu. Dodatkowo opis jest aktualizowany przy każdej zmianie treści tooltipsa.

Jeżeli nazwa bądź opis zostały nadane ręcznie, wówczas zmiana tooltipsa, bądź etykiety nie powoduje odpowiednich zmian opisu, bądź nazwy.

Środowisko Javy umożliwia nie tylko tworzenie aplikacji mogących być w łatwy sposób obsługiwanych przez screen reader, ale również dostarcza narzędzi umożliwiających tworzenie przyjaznych aplikacji – dostępnych w sposób bezpośredni. Dzieje się tak z powodu



możliwości zastosowania pakietu IBM SVK, dzięki któremu można tworzyć aplikacje z interfejsem głosowym [4].

## 6.2. Micromedia Director

Jednym z popularnych mechanizmów prezentacji danych w Internecie jest Macromedia Director. Niestety, środowisko to posiada wiele ograniczeń, nie ma również dostępnego żadnego opisu zasad tworzenia aplikacji w tym środowisku. Programy tworzone przy zastosowaniu Micromedia Director nie są zgodne z technologiami ułatwiającymi dostęp. Dla przykładu screen readersy potrafią odczytać tekst napisany na ekranie, nie istnieje jednak możliwość uzyskania informacji o obiekcie. Mówiąc w sposób bardziej obrazowy – użytkownik może odczytać nawę przycisku, lecz nie może go nacisnąć. Dodatkowo często zdarza się, że elementy ekranu są czysto graficzne. Nie istnieje możliwość przypisania im tekstowego zamiennika bądź opisu.

Micromedia Director nie umożliwia stosowania klawiatury do obsługi aplikacji w sposób standardowy. Owszem jest możliwe stosowanie klawiatury, ale wymaga to użycia odpowiedniego, własnoręcznie stworzonego przez programistę programu.

Część elementów aplikacji do złudzenia przypomina odpowiednie elementy systemu. Niestety, podobne są tylko z wyglądu, a ich obsługa programowa przebiega w zupełnie inny sposób, obcy i nieznanym screen readerom.

Podsumowując, można stwierdzić, że jeżeli programista chce stworzyć aplikację przyjazną osobom niewidomym, to jedynym sposobem jest napisanie programu dostępnego w sposób bezpośredni. Wymaga to więc od programisty stworzenia pełnego systemu obsługi klawiatury, a także wbudowanie interfejsu głosowego.

## 7. Podsumowanie

W trakcie przeglądania istniejących na rynku polskim aplikacji multimedialnych nie natrafiono na żaden produkt w pełni zapewniający dostępność osobom niewidomym. Z jednej strony stykano się z aplikacjami, które wykorzystywały bogaty interfejs graficzny, można powiedzieć, że go nadużywały – umieszczając nawet informację tekstową w postaci rysunków (aplikacje Optimus Pascal). Z drugiej zaś strony stykano się z aplikacjami, które potencjalnie mogłyby poprawnie współpracować ze screen readerami, ale zapomniano / nie pomyślano o kilku podstawowych, niezbędnych elementach, takich jak chociażby możliwość przenoszenia fokusu po elementach okna, czy nadawanie obiektom przyjaznych i zrozumiałych nazw.

W trakcie wyszukiwania aplikacji nie natrafiono na aplikacje wykorzystujące mechanizmy opisów dźwiękowych. Nie zetknięto się również z aplikacjami wykorzystującymi technologie Javy, z którą zapoznano się jedynie na przykładowych programach zagranicznych.

Problem niewidomych i ich prawa do dostępu i swobodnego korzystania z aplikacji multimedialnych powinien być nagłaśniany i poruszany w kręgach programistycznych.

## LITERATURA

1. Wakefield D., Dixon J. M.: How Blind People Can Access the Audio and Video Formats of the WWW: I-wave, Mpeg, Jpeg, Gif – Potholes or Billboards. CSUN 99 Papers.
2. Vickery L. J.: Techniques and Technology for Teaching Students with Vision Impairment. CSUN 2000 Papers. CSUN 2000 Papers.
3. Zajicek M., Venetsanopoulos I.: Using Microsoft Active Accessibility in a Web Browser for the Blind and Visual Impaired. CSUN 2000 Papers.
4. Freed G.: Advances in Accessible Web-based Multimedia. CSUN 2000 Papers.
5. Rothberg M., Wlodkowski T.: Making Multimedia Accessible: Guidelines and Solutions. CSUN 2000 Papers.
6. Sun Microsystems Accessibility Team: The Basics of the Java™ Platform – A User-Focused Discussion. CSUN 2000 Papers.

Recenzent: Dr inż. Ryszard Winiarczyk

## Abstract

In this paper were showed problems related with accessing by blind users to the multimedia software. In general we may say about two main ways of creating accessible software. The first one is directly approach and in such way developer must take care about all aspects of accessing to the application by blind user. Developer must create appropriate interface, keyboard access and voice system. The second way is a compatible access. In this approach developer must only take care about compatibility with screen reader (interface between application and voice synthesiser). Developer must only use objects that are typical for operating system and set proper properties.

We also try to analyse a state of multimedia software on polish market. On the one hand there is a lot of multimedia software, but on the other hand there is no software that can be



accessed by blind user with easy. It seems to be necessarily to publicize this situation and to try to improve it.

In this paper we pay attention to an internet applications. The one solution of creating such applications is Java language. It still becomes more and more popular. The main reason is independence of hardware and operating system. Very important are also integrated objects properties that can be used by screen-readers.