This paper should be cited as: Marasek, K. (2006). Użyteczność interfejsów głosowych. Unpublished paper presented at Interfejs użytkownika - Kansei w praktyce Conference, Warszawa 2006. Warsaw.

Użyteczność interfejsów głosowych

KRZYSZTOF MARASEK

Polsko-Japońska Wyższa Szkoła Technik Komputerowych
02-008 Warszawa Koszykowa 86

Interfejsy głosowe, systemy dialogowe, rozpozanwanie i synteza mowy, użyteczność.

Wstęp

Komunikacja głosowa z komputerem już od wielu lat jest marzeniem zarówno inżynierów użyteczności jak i zwykłych użytkowników komputerów. Wydawać by się mogło, że stworzenie głosowego interfejsu do komputera nie powinno stanowić w dobie tak szybkiego rozwoju technik komputerowych specjalnego problemu. Niestety, na przeszkodzie w efektywnym wykorzystaniu mowy w technice komputerowej staje nie tyle brak mocy obliczeniowej, co kłopoty ze stworzeniem użytecznego interfejsu głosowego w warunkach ograniczonego języka rozpoznawanego przez komputer.
W poniższej pracy przedstawiono zagadnienia związane z designem interfejsu głosowego w zastosowaniach telefonicznych oraz dla urządzeń mobilnych.

Budowa systemu dialogowego

Systemy dialogowe (Voice User Interface - VUI), komunikujące się z użytkownikem za pomocą mowy składają się z 3 podstawowych modułów: rozpoznawania, syntezy mowy i nadzorcy dialogu [1]. Budowa kazdego z tych modułów jest dośc złożona i, pomijając najnowsze rozwiązania, nie mają one z zasadzie części wspólnych. Poniżej skrótowo omówiono ich funkcje.

Moduł rozpoznawania mowy
Moduł rozpoznawania mowy (ASR) wczytuje spróbkowany sygnał akustyczny, wyodrębnia z niego charakterystyczne cechy i porównuje z wzorcami. W praktycznych rozwiązaniach stosuje sie zwykle wzorce statystyczne, zapamiętane w postaci tzw. ukrytych modeli Markowa [1] pojedyńczych dźwięków, które łączone w łańcuchy odwzorowują cechy akustyczne ciągów głosek, a co za tym idzie wyrazów i fraz. Wynikiem takiego porównania jest ciąg najbardziej prawdopodobnych modeli akustycznych (zwykle głosek), których sekwencja jest modyfikowana poprzez model języka, opisujący, w zależności od implementacji – bądź wszystkie możliwe do rozpoznania wyrażenia (np. gramatyka), bądź określający prawdopodobieństwa określonych ciągów słów (model statystyczny języka). Wynikem działania modułu ASR jest zazwyczaj najbardziej prawdopodobna hipoteza jaki ciąg słów został wypowiedziany. Należy przy tym zauważyć, że w przypadku podania na wejściu systemu słowa spoza dopuszczalnego zbioru wyrazów, system także wygeneruje

hipotezę dopuszczalnego słowa najbardziej zbliżonego do wejściowego. Użycie słowa spoza słownika jest najczęstszą przyczyną błędów ASR.

Moduł syntezy mowy
Synteza mowy (SS) jest także złożonym procesem. Zwykle [2] jej pierwszym etapem jest dogłębna analiza tekstu wypowiedzi, pozwalająca na jego zamianę na ciąg jednostek fonetycznych (np. głosek) wzbogaconą o informacje prozodyczne (intonacja, czas trwania poszczególnych elementów, pauzy, itp.). Następnie z bazy danych utworzonej z przetworzonych i posegmentowanych nagrań mowy naturalnej wybiera się najlepiej pasujące elementy, które następnie łączy się ze sobą modyfikując jednocześnie ich cechy prozodyczne. W zależności od wielkości bazy danych i reguł konkatenacji jej elementów można uzyskać syntetyczną mowę o wysokiej zrozumiałości i dużej naturalności.

Nadzorca dialogu
Modułem spajającym wejście (ASR) i wyjście akustyczne (SS) jest moduł nadzorcy dialogu (DM). Określa on zasadnicze funkcjonalności interfejsu głosowego, sposób reakcji na żądania uzytkownika i przebieg samego dialogu. Zwykle jego zadaniem jest zapewnienie przejścia z aktualnego stanu dialogu do następnego, co połączone jest z reakcją na działanie użytkownika i wygenerowanie pewnej treści w postaci dźwiękowej (lub innej). Zwykle DM działa na tekstowej postaci informacji. Istotną częścią DM są moduły ekstrakcji istotnych informacji z wypowiedzi użytkownika, jak i generacji tekstu. Wykorzystują one zaawansowane techniki przetwarzania języka naturalnego.
Podstawowy parametr określający strategię działania systemu dialogowego to inicjatywa w dialogu – komputera, człowieka lub mieszana. W zależności od tej strategii system może (lub nie) przewidywać kolejne kroki dialogu, co wpływa na stopień komplikacji całego systemu oraz na możliwość sugerowania użytkownikowi kolejnych kroków niezbędnych do osiągnięcia celu dialogu. Zwykle systemy o mieszanej strategii dialogu uważa się za najefektywniejsze (i najbadziej skomplikowane), choć w niektórych zadaniach ta wysoka efektywnośc bywa kwestionowana [3]. Techniczna realizacja DM zależy od strategii dialogu. Najczęściej stosuje się automaty o skończonej ilości stanów, w których stany odpowiadają odpowiedziom komputera na wypowiedzi użytkowników zapisanych w postaci przejść pomiędzy stanami, np. CSLU Toolkit [4]. Architektury systemów o mieszanej inicjatywie są zwykle bardziej skomplikowane i wykorzystują struktury znane w systemach AI (ramy, reprezentacje zdarzeniowe itp [1]).
Z punktu widzenia projektanta aplikacji istotne jest, że wiele meta-systemów do tworzenia systemów dialogowych (np. WebSphere IBM, SpeechPearls Philipsa) wykorzystuje języki pozwalające na specyfikację dialogu np. VoiceXML, SALT czy HDDL. Języki te umożliwiają także łatwą integrację VUI z innymi modalnościami wejściowymi, np. tonami DTMF w telefonii.

Komunikacja międzyludzka a komunikacja głosowa człowiek-komputer

Z punktu widzenia użyteczności istnieją zasadnicze róznice w przebiegu komunikacji za pomocą mowy w komunikacji interpersonalnej a komunikacją człowiek – komputer

(KCK). Upraszczając, podstawowym problemem KCK jest to, że w przeciwieństwie do procesu komunikacji międzyludzkiej, proces ten nie jest akcją naturalną i spontaniczną, co ze szczególną mocą odnosi się do komunikacji przy pomocy mowy. Podczas bezpośredniej interakcji, 55% treści emocjonalnych wyrażany jest w sposób niewerbalny za pomocą wyrazu twarzy, postawy i gestu. 38 % przekazywane jest przez ton głosu. Okazuje się więc, że za pomocą języka wyrażamy jedynie 7 % naszych uczuć [5]. Podstawowym problemem KCK jest brak przekazu treści emocjonalnych. Nie jest to oczywiście przeszkodą w realizacji prostych celów, wpływa jednak decydująco na satysfakcję użytkownika z interakcji z systemem.

W porównaniu do interfejsów graficznych systemy dialogowe są „szeregowe", tzn. przekaz informacji jest skoncentrowany tylko na jednym aspekcie i uszeregowany czasowo – nie można się spodziewać, że użytkownik będzie mógł wykonać kilka czynności jednocześnie (np. otwarcie dodatkowego okienka w GUI i wyszukanie informacji poprzez klikanie myszką w dodatkowe menu). W systemach, w których mowa jest jedynym medium przekazu, interakcja z użytkownikiem musi brać pod uwagę specyficzne ograniczenia mentalne użytkowników. Przyjmuje się zwykle, że użytkownik jest w stanie zapamiętać 5 do 9 cyfr na około 20 sekund po ich usłyszeniu. Nie można zatem zakładać, że użytkownik będzie w stanie efektywnie korzystać z systemu, w którym głosowo zapowiadana jest długa lista możliwych opcji, bo po prostu zapomni on jaką cyfrę lub wyraz ma powiedzieć w reakcji na prompt.

Dodatkowo, korzystając ze słownyh zapowiedzi użytkownik koncentruje się na znacznie węższym i czasowo-zależnym kontekście przekazu. Klasycznym przykładem [11] jest tutaj różnica pomiędzy odczytywaniem e-maili, a ich przeglądaniem na ekranie. O ile przeglądając możemy łatwo przerwać odczytywanie kolejnej wiadomości i przeskoczyć do emaila z nią związanego (np. o podobym nagłówku lub od tego samego nadawcy(, to w przypadku VUI dokonanie takiej operacji jest trudne (o ile wogóle możliwe).

Oczywiście, także wiele czynników wynikajacych z ograniczeń technologii rozpoznawania i syntezy mowy wpływa na niedoskonałośc głosowych systemów KCK, np.:

- mowa syntetyczna nie brzmi całkiem naturalnie,
- mowa syntetyczna wymaga wiekszego zaangażowania słuchacza [6],
- rozpoznawanie mowy jest zawodne, trudne jest poprawianie błędów rozpoznawania,
- słowniki, modele języka i gramatyki są zawsze ograniczone, a więc nienaturalne w użyciu,
- systemy zarządzania dialogiem nie są w stanie przewidzieć wszystkich możliwych kombinacji we/wy.

Mimo tych wad i ograniczeń interfejsy głosowe mogą być efektywne i satysfakcjonujące dla użytkownika. Muszą jednak uwzględniać podstawowe rożnice w stosunku do interfejsów graficznych:

- sekwencyjna struktura przekazu,
- mniej opcji do wyboru w danym kroku dialogu ,
- węższy kontekst dialogu niż w przypadku GUI (użytkownik nie ma dostępu do dodatkowej wizualnej informacji o możliwych opcjach).

Użyteczność interfejsów głosowych

Zgodnie z projektowaniem ukierunkowanym na użytkownika, projektując interfejs głosowy musimy opowiedzieć sobie na podstawowe pytanie: kto i w jaki warunkach korzystać będzie z interfejsu głosowego? Mowa jako interfejs KCK używana jest najchętniej w tych sytuacjach , kiedy jest jedynym lub preferowanym medium przekazu informacji – zatem naturalne obszary zastosowań to telefonia, urządzenia mobilne (ze względu na niewygode korzystania z interfejsu graficznego), systemy tłumaczące, kioski informacyjne, systemy dla specjalnych grup użytkowników (radiologiczne, prawnicze, call centers, dla osób niepełnosprawnych itp.). Postępy technologii, upowszechnienie rozwiązań wykorzystujących VoiceXML zdeaktualizowały pogląd, że mowa jest wykorzystywana tylko wtedy, gdy: 1) uzytkownik nie ma innej alternatywy 2) z jakiegos powodu nie jest w stanie użyć klawiatury telefonu do wybou opcji [7].

Z punktu widzenia technik użyteczności przygotowanie interfejsu głosowego nie różni sie od przygotowania interefejsu graficznego. Określenie profilu użytkownika, zakresu stosowania interefejsu, iteracyjny design i testowanie rozwiązań to podstawowe kroki przygotowania aplikacji. Pewne elementy interfejsów głosowych są jednak unikalne: myśle tu o przygotowaniu promptów, korekcji błędów i testowaniu systemu.
Prompty, czyli wypowiedzi formułowane przez komputer, są szczególnie istotne dla przebiegu dialogu, bowiem [8]:
- użytkownicy adaptują sie do sposobu wypowiedzi komputera i starają się używac tych samych zwrotów i wyrazów co komputer,
- prompty mogą sterować przebiegiem dialogu, ułatwiając interpretację wypowiedzi użytkownika przez DM,
- źle skonstruowane prompty mogą skonfundować uzytkownika, nawet jeśli odpowiedzią ma być proste Tak lub Nie,
- prompty muszą być dostosowane do wszystkich użytkowników, nie mogą być zatem nużące dla zaawansowanych i nie za trudne dla początkujących użytkowników.

Proste heurystyki pozwalające osiągnąć te cele, to [7,8,11]:
- reguła „7+/- 2” czyli zwięzłe prompty, co pozwala także na oczekiwanie zwięzlych odpowiedzi ze strony użytkownika,
- struktura systemu powinna umożliwiac przerywanie promptu (barge-in),
- wykorzystywać słowa rozpoznawane przez ASR,
- nie dawać użytkownikowi zbyt wielu możliwości wyboru sformuowania odpowiedzi,
- utrzymywanie zbliżonej składni promptów,
- unikanie zbliżonych do siebie promptów (aby ułatwić użytkownikowi nawigację) jak i unikanie podobnych fonetycznie słów,
- użycie dodatkowych promptów (rozszerzonych) przy braku reakcji użytkownika lub skracanie promptów przy szybkich reakcjach użytkownika,
- wykorzystanie tutoriali dla nowych użytkowników,
- jasny związek z procedurą unikania i korekcji błędów.

Procedury korekcji błędów wykorzystują bezpośrednie lub pośrednie potwierdzanie zrozumienia użytkownika, oferowanie innych modalności wprowadzania danych (np. DTMF), upraszczanie opcji wyboru (aż do Tak lub Nie). Zwykle wykorzystują one też

oferowaną przez ASR miarę wiarygodności rozpoznania – jeśli spada ona poniżej określonej wartości reakcja użytkownika jest obligatoryjnie potwierdzana.

Poniżej omówiono poszczególne funcjonalności budujące poprawną użyteczność VUI, wg. [11]:

Help i Tutorial
Efektywnym substytutem modelu mentalnego użytkownika jest użycie kontekstowych helpów i tutoriali dla początkujących użytkowników. Ich działanie powinno być uaktywniane poprzez szereg zawsze dostępnych komend głosowych (takich jak: HELP, WYCOFAJ, PRZERWIJ, ZACZNIJ OD NOWA). Powinny one dobrze tłumaczyć ograniczenia technologii (np. wpływ hałasu na jakość rozpoznawania mowy) i zawierać jasne wskazówki (z przykładami) co można powiedzieć w danym kroku dialogu. Dobrze też, aby ułatwić użytkownikowi zadanie podając w prompcie ilość możliwych odpowiedzi.

Efektywne i eleastyczne gramatyki ASR
Uwzględnienie wszystkich możliwych odpowiedzi użytkownika w danym kroku dialogu nie jest zwykle możliwe. Dlatego tak ważne jest umiejętne sterowanie użytkownikiem poprzez prompty, tak aby powiedział tylko takie słowa, których się spodziewamy (np. wielokrotnie powtarzając typową frazę). W przypadku jednak języka polskiego, języka o swobodnym szyku zdania (i bogatej fleksji) , gramatyka opisująca możliwe wyrażenia musi być bardzo elastyczna i dopuszczać wiele możliwych form wypowiedzi. Z drugiej strony, gramatyki zawsze zezwalają na zbyt wiele możliwych wypowiedzi (np. gramatyka opisując możlwie numery kart kredytowych opisuje zazwyczaj sekwencję 4 grup po 4 cyfry). Konieczna jest zatem dodatkowa analiza, czy rzeczywiście podana sekwencja słów odpowiada dozwolonej wypowiedzi (np. obliczenie sumy kontrolnej dla numeru karty kredytowej)
Szczególnie istotna dla prawidłowego skonstruowania gramatyki jest analiza danych zebranych przy okazji testowania prototypu VUI. Dopiero bowiem na tym etapie można skontrontować przyjęte założenia z rzeczywistymi reakcjami użytkownika.

Szybka reakcja systemu i pauzy
Użytkownicy przyzwyczajeni są do szybkiej reakcji interlokutora. System musi też szybko udzielić jakieś odpowiedzi, bowiem dłuższa pauza spowoduje wrażenie zerwania rozmowy. Z drugiej strony, system musi wybaczać długie pauzy ze strony użytkownika, który może być nagle zajęty czymś innym (np. w trakcie jazdy samochodem system VUI klimatyzacji musi być przygotowany na to, że kierowca będzie przez dłuższą chwilę zajęty sytuacją na drodze).

Prawidłowość języka i prozodii
Trzeba zwrócić szczególną uwagę na lingwistyczną jakość systemu. Oczekuje się, że użytkownik nie będzie w swoich wypowiedziach używać agramatyzmów, nieprawidłwych form wyrazów czy przekręcać ich wymowy. To wymaga oczywiście użycia odpowiednich form w promptach. Trzeba też odpowiednio zbudować prozodię generowanej wypowiedzi, zwracając szczególną uwagę na intonację pytań, czy też na kontynuację wypowiedzi (obniżenie głosu na końcu zdania sugeruje koniec wypowiedzi

– użytkownik może w tym momencie zacząć mówić choć system nadal kontynuje wypowiedź)

Osobowość systemu
Postępy technologii syntezy mowy pozwalają na tak finezyjne sterowanie jej generacją, że można upodobnić głos i sposób mówienia systemu do cech realnych postaci (osób publicznych, bohaterów filmowych). Wiedząc, że użytkownicy wolą rozmawiać ze znanymi postaciami (ale niekoniecznie realnymi, np. bohaterami seriali TV), można próbować nadać VUI cechy danej postaci. Można się spodziewać podwyższonej akceptacji takiego systemu.

Iteracyjne tworzenie systemu
Iteracyjny design interfejsu głosowego wymaga przede wszystkim dużego wysiłku związanego ze zbieraniem danych, w tym wypadku nagrań użytkowników prototypowego systemu. Rozmowy użytkowników są nagrywane, analizowane i wykorzystane do re-designu dialogu, ale też do treningu systemu ASR. Popularna reguła głosi, że po każdej iteracji przygotowania systemu stopa błędów rozpoznawania spada o połowę. Na wczesnych etapach przygotowania systemu wykorzystuje się paradygmat nagrań Wizard-of-Oz (np.system SUEDE [9]), przechodząc z czasem do systemów w pełni automatycznych (np. VoxNauta firmy Loquendo). Trzeba przy tym zauważyć, że doprowadzenie do akceptowalnej jakości działania systemu wymaga z reguły bardzo wielu iteracji designu. Istotnym elementem iteracyjnego podejścia do tworzenia VUI jest metodyka tworzenia gramatyk i promptów, w której dokładnie i po wielokroć analizuje się dane z eksperymentów, tak aby polepszyć iterakcję użytkownika z systemem.

Ocena jakości VUI
W ostatnich latach zaproponowano metryki opisujące ilościowo jakość systemów dialogowych, np. [10] – dotyczą one głównie parametrów ASR, zwykle mniej informacji wnoszą do oceny zadowolenia użytkownika. Przykładowe proponowane miary to:
Query Density, opisująca ilość nowych koncepcji wprowadzonych w pytaniu użytkownika,
Concept Efficiency, mierząca średnią ilość wypowiedzi konieczną do zrozumienia danego konceptu przez system.
Miary takie stosowane są łącznie z miarami jakości ASR (*Word Error Rate, Sentence Error Rate*) oraz ankietami oceny satysfakcji użytkownika.
Warto także wspomnieć o inicjatywach gremiów standaryzujących (np. W3). Dotyczą one głównie systemów IVR, ale niekiedy także tworzenia aplikacji wykorzystujących rozpoznawanie i syntezę mowy.

Podsumowanie

Użyteczność interfejsów głosowych daje się zapewnić przy pomocy klasycznych metod – projektowanie zorientowane na użytkownika, analiza użytkowania, iteratywne projektowanie i ocena użyteczności są konieczne także w tym przypadku. Ich specyfika

daje sie zauważyć w kilku elementach – tworzeniu promptów, korekcji błędów i metodach testowania.

Literatura

1. De Mori, Spoken Dialogues with Computers: Signal Processing and Its Applications, Academic Press, 1998,
2. Huang X., Acero A., Hon H.. Spoken Language Processing, Prentice Hall, 2001
3. Walker, M., Fromer, J., Fabbrizio, G., Mestel, C., Hindle, D. What can I say?: Evaluating a spoken language interface to Email. In Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems: 582-589, 1998.
4. CSLU Toolkit, http://cslu.cse.ogi.edu/toolkit/
5. Mehrabian, A. (1971). Silent messages. Wadsworth, Belmont, California.
6. Lai J., Wood D., Considine M. The Effect of Task Conditions on the Comprehensibility of Synthetic Speech. In Proceedings of CHI 2000 (The Hague, April 2000), ACM Press: 321-328. 2000.
7. Mane A., Boyce S., Karis D., Yankelovich N., Designing the User Interface for Speech Recognition Applications, A CHI 96 Workshop, SIGCHI Bulletin Vol_28 No_4, October 1996.
8. Turunen M., Speech Application Design and Development, University of Tampere, http://www.cs.uta.fi/hci/spi/reports/SADD.pdf
9. http://guir.berkeley.edu/projects/suede
10. Glass J., Polifroni J., Seneff S., Zue V., Data Collection And Performance Evaluation Of Spoken Dialogue Systems: The MIT Experience, Proc. ICSLP, Beijing, China October 2000
11. Byrne B., Turning GUIs into VUIs: Dialog Design Principles for Making Web Applications Accessible By Telephone, VoiceXML Review, Vol. 1, Issue 6, June 2001
12. IBM Redbook, Speech User Interface Guide, May 2006, ibm.com/redbooks