Zagadnienia

- Wstęp do systemów IVR
- Projektowanie interfejsów głosowych
- SRGS, VXML
- MRCP, SIP, RTP
- Wykorzystanie ASR, BVV, BVR
- NLP, semantyka - wstęp
- Modelowanie dialogu
- Protokoły

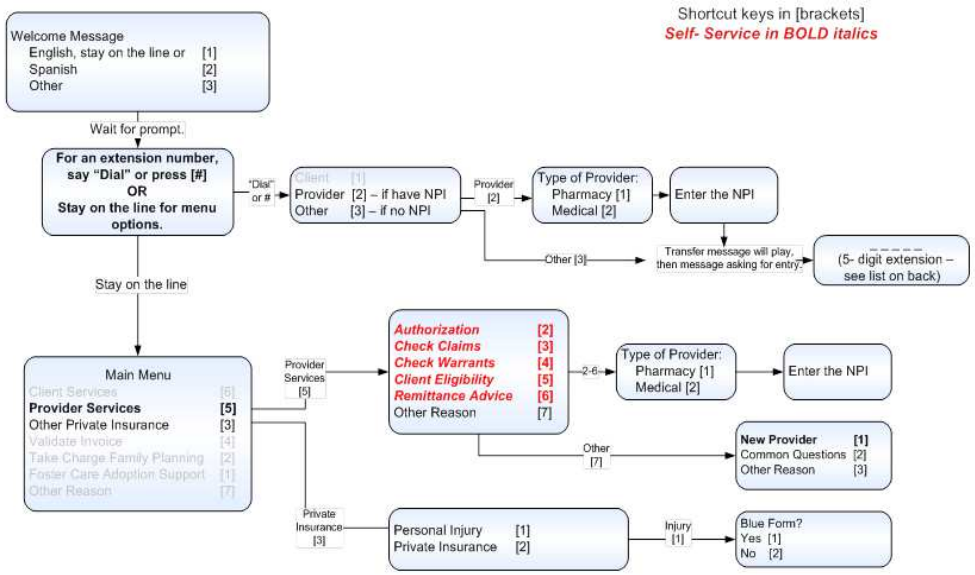
IVR

- http://www.youtube.com/watch?v=3nrdq9eHApI
- Początki: lata 70-tę w USA i Wielkiej Brytanii (*Call-centers*)

IVR

- **Interactive Voice Response**
- Telefoniczna (ale nie tylko) obsługa klienta.
- Zdefiniowany zakres akcji.
- Komunikaty
  - Statyczne („Witamy w naszej firmie, for English press 2.”).
  - Dynamiczne („Stan Twojego konta wynosi dwa miliony siedemset dwadzieścia euro.”)
  - Nagrania lektora (Wave)
  - TTS – T*ext-to-Speech synthesis*
- Interakcja za pomocą DTMF (*Dual-Tone Multi-Frequency*)
- Interakcja za pomocą ASR (*Automatic Speech Recognition*)
  - W przypadku kontaktu z ASR - Do 2004 roku 96% użytkowników stosowało wcześniej DTMF.
- Struktura definiowana za pomocą VoiceXML, SALT, T-XML
- Tworzenie za pomocą GUI z wykorzystaniem HTTP, JAVA

Przykładowy schemat IVR call-flow, graf dialogowy, ...



DTMF

- *Dual-Tone Multi-Frequency*
- Sinusoida, odporność na kodowanie
- Nie da się symulować za pomocą głosu
- Wspierane przez ASR

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Dlaczego IVR ?

- Obniżenie kosztów obsługi. (*Cost-per-call ratio*)
  - Ludzie, biua, infrastruktura, utrzymanie, szkolenia a właściwie ich brak, zintegrowane zarządzenie, analiza automatyczna, skrócenie czasu rozmowy
- Dostępność obsługi 24/7.
- Obsługa nagłego dużego ruchu
  - Awarie, promocje, …
- IVR to oczekiwana forma kontaktu.
  - Pod warunkiem możliwości połączenia z żywym agentem. :-)
- Dowolna dziedzina gospodarki i tematyka rozmowy.
- Prestiż ?

Model wdrożenia

- CPE (*Customer Premise Equipment*)
  - System IVR zainstalowany na sprzęcie i w siedzibie przedsiębiorstwa/instytucji, w której pracuje.
  - ARU – *Audio response unit*
- IVR *Hosting* / SaaS / *Cloud*
  - Udostępniony i utrzymywany przez operatora systemu IVR, na zlecenie klienta.
- Zalety i wady ?

Najważniejsi dostawcy systemów IVR

- Avaya Solutions
  - Avaya Aura, Avaya Contact Center, Avaya Interactive Response, Avaya Voice Portal
  - Integracja m.in. poprzez rozwiązania natywne Avaya: Avaya Agile Communication Environment™ (ACE) Toolkits, Avaya Aura Session Manager
  - Wymagana licencja Licencja Avaya Agile Communication Environment, Komunikacja z Avaya Media Processing Server
  - www.avaya.com

Interactive Intelligence

- Interactive Intelligence Customer Interaction Center IVR
- Interactive Intelligence Interaction Director
- Integracja poprzez API Interactive Intelligence
- Integracja poprzez standardy techniczne
  - SIP, MRCP, ...

Genesys Telecommunications Laboratories

- Genesys Inbound Voice
- Genesys Voice (IVR) Platform
- Integracja poprzez Genesys SIP Server
- Konieczna licencja na Genesys SIP Server
- Genesys Software Development Kits
- Genesys Customer Interaction Management Platform

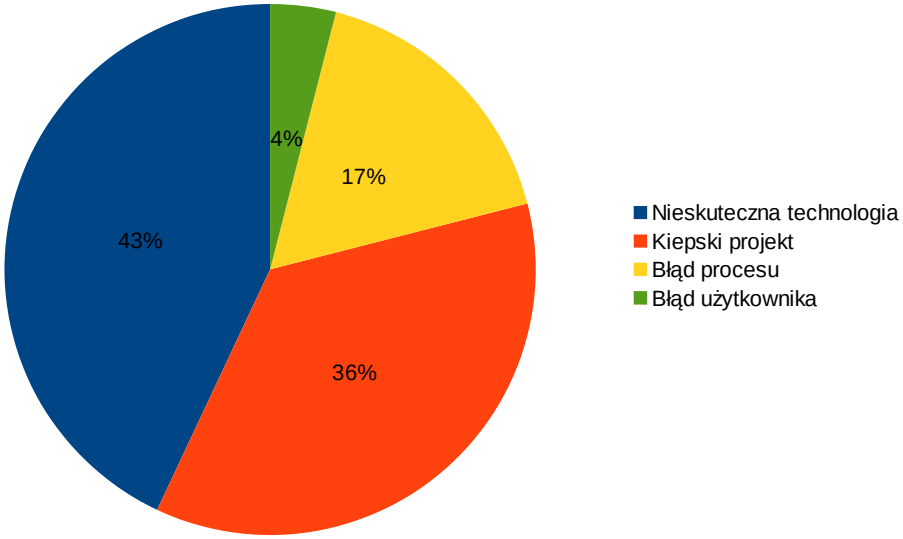
Projektowanie IVR – najważniejsze kwestie

- Intuicyjność
- Szybkość
- Struktura dialogu zależna od czasu i dnia
  - W czasie roboczym, w dni robocze
  - Poza godzinami pracy
- Obsługa języków („*For English press 2.*”)
- Obsługa nagłego przyrostu ruchu
  - Opcje (np. awarii) lepiej zaprojektować i uśpić niż budować nowy IVR na gorąco.

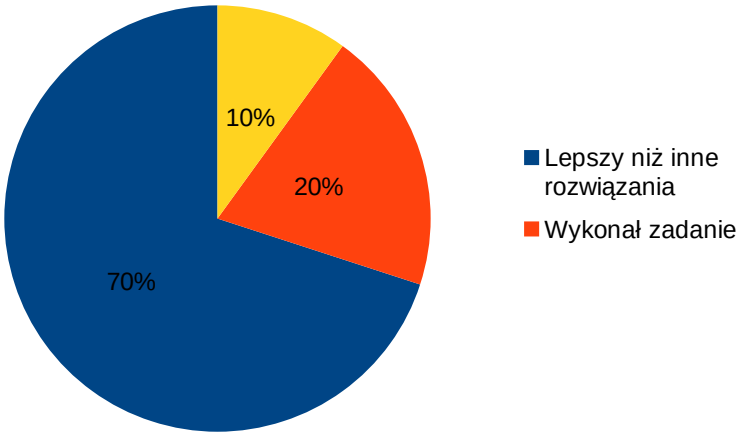
Projektowanie IVR – najważniejsze kwestie

- Określenie z jakimi oczekiwaniami dzwoni klient.
  - Jakie zadania chce zrealizować ?
- Jaki jest cel wdrożenia IVR ?
  - Obniżenie kosztów
    agent: 0.3 USD – 1.5 USD / min
  - Koszt załatwienia sprawy:
    - Email: 10 USD, Chat: 8 USD, WEB: 7 USD
    - Telefon - agent: 6 USD, ASR: 0.2 USD
  - Podniesienie jakości obsługi (QoS, QoE)
  - Uzyskanie informacji (np. ankiety)
  - Badanie zadowolenia klientów CRM (*Customer Relationship Management*)
  - Zwrot inwestycji, średnio po 11 miesiącach (*Opus Research,* Miller 2006)

Użytkownicy niezadowoleni w systemach typu *Self-service*



Użytkownicy zadowoleni w systemach typu *Self-service*



Analiza użytkownika

- Cechy i możliwości użytkownika
  - fizyczne, umysłowe, zmysły
  - Wiek (np. pasmo do 4 kHz)
- Pamięć robocza
- Pamięć semantyczna (znaczeniowa)
- Pamięć proceduralna
- Skupienie, uwaga
- Kompetencje językowe

Analiza użytkownika

- Jak często wykorzystywany jest system?
- Jaka jest motywacja jego wykorzystania ?
- W jakim otoczeniu system będzie najczęściej wykorzystywany ? (biuro, ulica, dom,...)
- Jaki typ połączenia głosowego będzie obsługiwać system ?
  - PCM, ISDN, GSM, VoIP, …
- W jakich językach ?
- Czy użytkownicy są przyzwyczajeni do samoobsługi ?
- Jaki % użytkowników stanowić będą osoby starsze ?

Analiza użytkownika - zadania

- Jakie zadania są najczęstsze ?
- Czy zadania są znane użytkownikom (np. z innych kanałów), czy są nowe ?
- Czy zadania można wykonać innymi kanałami ?
- Czy będzie opcja transferu do Agenta w kontekście danego zadania ?
- Jakie **typowe** słowa i zwroty są wykorzystywane przez użytkowników do opisu zadania ?

Analiza użytkownika - czy korzystać z ASR

- Za:
  - Oszczędność czasu, pieniędzy, zawsze dostępne, możliwość realizacji zadań niedostępnych innymi metodami, unikanie kontaktu z żywym człowiekiem
  - Brak dostępu do komputera/Internetu
  - Ograniczenia psychoruchowe (inwalidzi)
- Przeciw:
  - Zadanie dotyczy modalności graficznej (np. map, obrazów, instrukcji składania mebli)
  - Użytkowanie w zaszumionych warunkach
  - Wady słuchu, mowy
  - Wykorzystanie w miejscach wymagających ciszy (sądy, biblioteki, itp.)

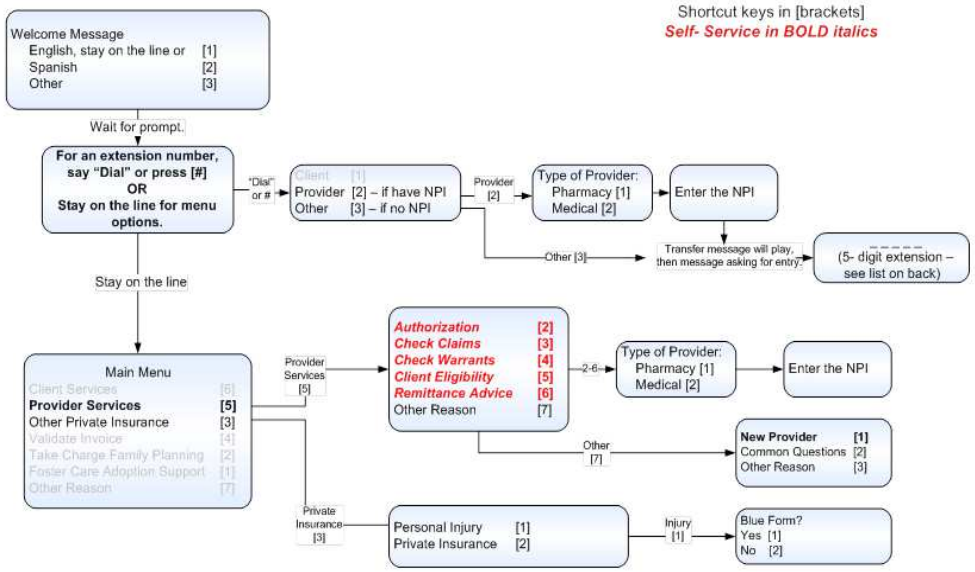
Projektowanie

- *„Nie można zaprojektować tego czego się nie da zdefiniowac!”* (Larson 2005)
- Koncepcja – przykłady dialogów, realizacji zadań
- Założenia wysokopoziomowe:
  - Głos / TTS
  - Rodzaj gramatyki, Zapytania
  - ASR/DTMF, Komendy globalne, agent, języki
- Założenia niskopoziomowe:
  - Reżim czasowy, projekt dialogu, projekt menu i zapytań, gramatyki
- Diagram dialogu (*Call-flow*)
- Specyfikacja dialogu
- Prototypowanie

Projektowanie...

- Diagram dialogu (*Call-flow*)
  - Jakie zadania ma wykonac użytkownik ?
  - Jakie etapy musi wykonać by zakończyć zadanie ?
  - Gdy użytkownik odpowie, lub nie odpowie (z różnych względów) co system ma zrobić ?
    - Typowe odpowiedzi,
    - Nietypowe odpowiedzi, „Gdzie są moje pieniądze?”
    - Błędy
  - Co system może zrobić automatycznie ?
  - Reprezentacja w postaci grafu.
    - Węzły – zapytania (np. gramtyki SRGS)
    - Krawędzie – możliwe odpowiedzi
- Specyfikacja dialogu – na podstawie przykładów rozmów
- Prototypowanie

Przykładowy schemat IVR
call-flow, graf dialogowy, … → dokumenty



Prototypowanie

- Prototypowanie
  - Metoda „Czarodziej z Oz”
  - Gotowy wydrukowany skrypt i graf
  - Dwoje testujących
    - Bez kontaktu wzrokowego (telefon lub plecy-w-plecy)
    - IVR i użytkownik
  - Czarodziej zaznajomiony ze skryptem.
  - Użytkownik nie zna skryptu.
  - Odgrywają scenki od początku do końca.
  - Metoda pozwala ocenić wygodę (*usability*).
  - Sprawdzić diagram dialogu (*Call-flow*)
  - Implementacja w Voice XML
  - Implementacja w systemie (programowanie)

Podstawowe zasady projektowania IVR

1) Pozwolić klientowi wybrać najczęściej stosowane opcje na początku
   - Płaskie menu, analiza statystyk użycia.
2) Należy utrzymać umiarkowaną (max 5) liczbę opcji.
   - Zbyt duża liczba opcji powoduje spadek jakości obsługi. *(Patrz ten slajd!)*
3) Zawsze powinna być możliwość połączenia z żywym agentem (0-DTMF).
4) Należy stosować język potoczny, komunikaty powinny być krótkie.
5) Nie należy stosować zbędnych uprzejmości:
   - *Twój telefon jest ważny, dziękujemy, że dzwonisz, proszę bardzo, dziękuję...*
6) Należy przypominać (co jakiś czas) o możliwości powrotu do Menu Głównego lub poziomu wyżej.
7) Należy definiować stałe akcje domyślne do obsługi stałych elementów interakcji: agent, język, menu główne.
8) Należy stosować ten sam głos w obrębie rozmowy.
9) Przed wdrożeniem należy wykonać testy i optymalizację systemu pod kątem jego wydajności i intuicyjności. (np. *Wizard-of-Oz*)

Skala do oceny dialogu wg Polkosky'ego

Czynnik	Opis
Liczność	Zmiany tematu: Czy system prawidłowo odpowiedział gdy zapytano o inny temat ?
Jakość	Ścisłość: Czy system zapewnił właściwą, jasną odpowiedź ? Wieloznaczność: Czy odpowiedź była wieloznaczna ?
Relacja	Wydajność: Czy informacja została podana w sposób wydajny? Zwartość: Czy system odpowiedział treściwie ? Szybkość: Czy szybko wykonano oczekiwane zadania?
Zachowanie	Ewentualność: Czy system zapewnił odpowiedź na podstawie zapytania? Rozwojowość tematu: Czy system poszerzył zakres rozmowy na dany temat ? Pomoc: Czy system był pomocny? Przeidywalność: Czy interakcja z systemem przebiegała w sposób oczekiwany ? Zwartość: Czy interakcja przypominała znane interakcje z innymi systemami ? (DTMF, rozmowa z człowiekiem, WWW)

Reżim czasowy (elementy)

- *„Właściwe słowo jest wiele warte, ale żadne z nich nie będzie warte tyle, co właściwie postawiona pauza.”* Mark Twain

- Typowa prędkośc mowy to 150-250 ms na sylabę.
- Człowiek zaczyna przetwarzać mowę po zgromadzeniu ok 250 ms danych.
- Rozmówcy dostosowują swoją prędkość w trakcie dialogu.
- Przerwy w rozmowie dłuższe niż 1 sekunda mogą zostać uznane za oznakę problemu.
- Naturalne przerwy w trakcie dialogu są nie dłuższe niż 500 ms.
- Przerwy do 250 ms nie powodują chęci odpowiedzi.
- Przerwy powyżej 1000 ms (95%), 1300 ms (99%) powodują silną chęć odpowiedzi.

Czas oczekiwania na konsultanta

- Bo ograniczone zasoby:
  - ludzkie, sprzętowe, programowe (licencje TTS, ASR)
- Szybka obsługa = wysoki koszt utrzymania
- Długi czas oczekiwania = niskie koszty, ale i niska satysfakcja klientów
- Ludzie oczekują obsługi typu FIFO (sprawiedliwość społeczna, Larson 1987)
- Zbyt długi czas oczekiwania wywołuje wzrost irytacji i stratę klienta (rozłączenie).

Czas oczekiwania, c.d.

- Skrócenie czasu oczekiwania
  - Rzeczywiste (właściwe kolejkowanie, *load-balancing*, itp.)
  - Postrzegane
    - Dźwięki: ton, klik, szum (nieskuteczne)
    - Komunikaty
      - Preferowane potwierdzenie wykonania akcji
      - Komunikaty w trakcie oczekiwania są irytujące
    - Muzyka (Polkosky 2001)
      - Właściwy dobór muzyki.
      - Dla oczekiwania dłuższego niż 4 sekundy.
      - Muzyka skraca postrzegany czas oczekiwania (redukuje irytację).
      - Wtrącanie przeprosin za czas oczekiwania nie redukuje irytacji.
      - Informowanie o postępie kolejkowania lub skracaniu się czasu oczekiwania nie skraca postrzeganego czasu ale redukuje irytację.
      - Najlepiej odbierane gatunki muzyki: klasyczna, jazz, relaksacyjna.

Decyzje wysokopoziomowe

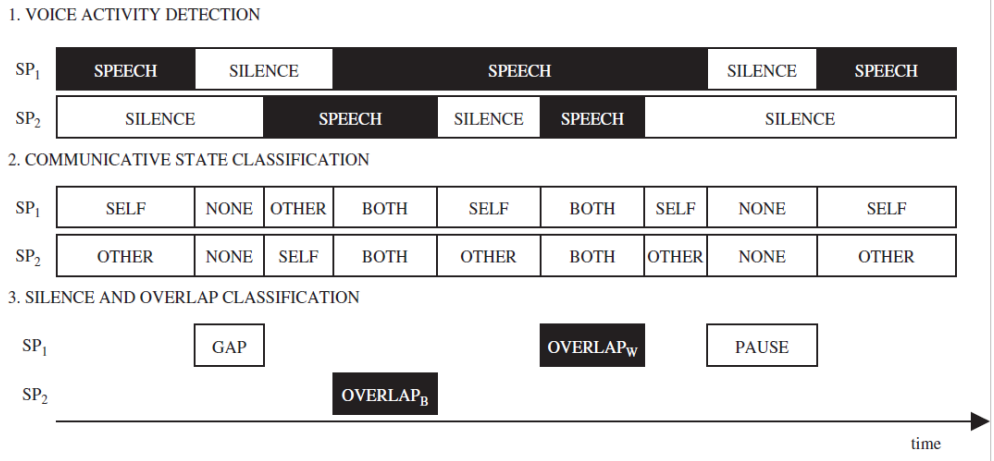
- Wtrącenia → efekt lombardzki po 300-500ms
- Złożoność gramatyk dla ASR
- TTS vs nagrania
- „Osobowość" systemu
- Dźwięki w komunikatach ? (dzwonki, sygnały, informacja po 4-8 sek., itp.) Lepiej dźwięki o bogatym spektrum niż tonowe, najlepiej o czasie trwania 50-75 ms, nie więcej niż 500-1000 ms.
- „Spłaszczenie" menu
- „Spłaszczenie" formularzy
- Komendy globalne i opcja wstecz, (opcja wyjścia?)
  - Komendy globalne i pomocnicze można podać po 1500-2500 ms ciszy od ostatniego zapytania – jako pomoc na brak odpowiedzi.
- Czy będzie dostępność żywych agentów ?
- Jeśli się da – lepiej wykorzystać prostsze rozwiązania.

Decyzje niskopoziomowe

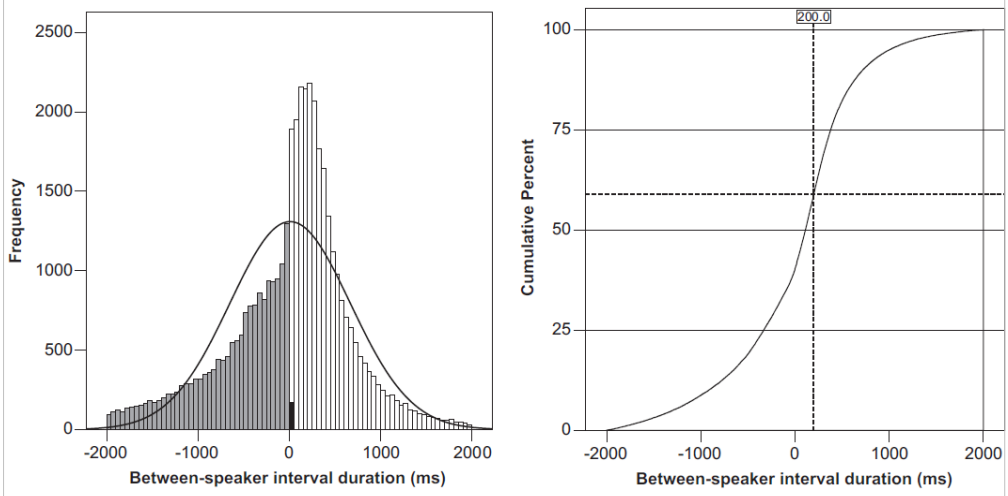
- Konkretne rozwiązania
- Fraza powitalna (konieczna)
  - NIE pytać czy ASR czy DTMF
  - NIE sprzedawać
  - NIE odsyłać do WWW
  - NIE informować, że „Twój telefon jest ważny.”
  - Nie informować, że nagrywane
    (chyba, że wymaga tego prawo)
- Zależności czasowe
  - *Noinput timeout*
    VoiceXML *default* = 7000ms, ale 5000ms też daje radę
  - Pauzy i przerwy – max. 300-500-750 ms (np. między opcjami)
  - Czas odpowiedzi w rozmowie < 1000 ms,
    naturalnie ok 300-600ms

Pauses, gaps and overlaps in conversations

Mattias Heldner , Jens Edlund



Rozkład odstępów czasowych między mówcami



Dialogi

- Komunikaty mogą być dyrektywne
  - „Wybierz: rachunki, karty, lub inwestycje.”
  - np. dla małych gramatyk ASR
- Niedyrektywne
  - Zazwyczaj tylko dla systemów otwartych
  - Wykorzystanie NLP
  - Formularze (np. *semantic frames*)
- Komunikowanie o błędzie lub braku zrozumienia przez ASR NIE pomaga i zazwyczaj nie poprawia QoE.
  - Lepiej podjąć próbę jeszcze raz, bez dodatkowych operacji.
- Błędy (ASR, podobieństwa, „Tak / Nie”)
  - Dyktowanie liter („M jak Maria”)
  - Potwierdzanie treści
    - za każdym razem vs po zgromadzeniu informacji
  - Wykorzystanie *N*-best listy i progów wiarygodności

Kwestia wieku

- Ludzie młodzi → Internet
- Osoby starsze:
  - 50+ to 80% użytkowników
  - 60+ to 50% użytkowników
  - 40- to 5% użytkowników
    - » (wg Fidelity Investments)
- Sukces:
  - 65- : 60%
  - 65+ : 40%
- Powody
  - Spadek pojemności pamięci roboczej
  - Problemy słuchowe
  - Problemy manualne (DTMF)

IVR dla osób starszych

- Oczekiwania stawiane przed IVR przez osoby starsze są inne niż oczekiwania młodych.
  - Chęć naturalnej rozmowy mimo świadomości rozmowy z maszyną.
  - Udawanie naturalności powoduje więcej błędów.
- Konieczność zachowania spójności
  - Głos, sygnały, stałe komendy
  - Jasny podział DTMF/ASR
- Zachowanie niskiej prędkości działania
  - Wydłużony czas oczekiwania na reakcję (5+ sek.)
  - Przedstawianie opcji w odstępach (na wokalizację)

IVR dla osób starszych, c.d.

- Długie komunikaty i wielokrotne wybory utrudniają ukończenie zadania.
  - Zniechęcenie i połączenie z konsultantem
- Zmniejszona pojemność pamięci roboczej osób starszych powoduje trudności w komunikacji z systemem IVR.
- Długie listy opcji powodują, że osoby starsze często wybierają opcję ostatnią, a nie pożądaną.
- Osobom starszym trudno jest zareagować na popełniony błąd.

IVR dla osób starszych, c.d.

- Należy unikać stosowania żargonu lub nowomowy, języka technicznego lub specyficznego.
- Brak zrozumienia skutkuje losowym wyborem lub połączeniem z konsultantem.
- Do określenia obiektów/akcji itp. należy stosować słowa znane i potoczne.

IVR dla osób starszych, c.d.

- Osoby starsze w interakcji z systemami IVR są często nadmiernie uprzejme.
- Trudności z wtrąceniem się (barge-in)
- Nadużywanie zwrotów tj. „dziękuję”, „proszę”.
  - Należy to mieć na uwadze projektując strukturę IVR, szczególnie w przypadku stosowania ASR.
- Oczekiwanie na komunikat „Do widzenia” lub sygnał rozłączenia.
- Należy informować użytkownika o możliwości rozłączenia się w odpowiednim momencie.
  - Oszczędność kanału

IVR dla osób starszych, c.d.

- Większa chęć stosowania DTMF niż ASR przez osoby starsze. (przyzwyczajenie)
- Konieczność zapewnienia możliwości wykorzystania DTMF do podania istotnych danych: PIN, login, itp.
- Nawigacja za pomocą DTMF wspomaga interakcję poprzez wykorzystanie pamięci proceduralnej (pamięć ruchu).
- ASR pomaga jednak osobom z problemami ruchowymi lub używającymi smartfonów lub małych telefonów komórkowych (małe lub trudno osiągalne klawisze).

IVR dla osób starszych, c.d.

- Pogorszenie słuchu osób starszych najczęściej jest bardziej dotkliwe w zakresie wysokich częstotliwości. (>4kHz)
- Wskazane jest stosowanie komunikatów wypowiadanych głosem o niższym brzmieniu (*f*0, *f*1-*f*3, harmoniczne).
- Niskie częstotliwości są mniej wrażliwe na zjawisko maskowania psychoakustycznego.

Testowanie

- Weryfikacja założeń systemu (*System Verification Tests*)
- Testy akceptacji przez użytkowników (*Customer Aacceptance Tests*)
- Testowanie gramatyk
  - Zasięg językowy *vs*
  - Jakość rozpoznawania ASR (tunning, dodatkowe tagi, TAK/NIE, fonetyka) *vs*
  - Naturaloność i efektywność →
  - Zawężenie lub poszerzenie gramatyk/modelu języka
- Definiowanie zadań testowych użytkownikom powinno odbywać się z możliwie małym narzuceniem informacji językowej.
  Np. za pomocą obrazów:
  100 USD = ? zł
- Testy ilościowe (np. statystyczne) prowadzone na losowej reprezentatywnej grupie problemów (gramatyk, przypadków, wartości, itp.).
- Przynajmniej 6 – 12 użytkowników z 50% parytetem płci.
- Przynajmniej 100 różnych fraz testujących.

Testowanie

- Pomiary % zadań ukończonych prawidłowo z punktu widzenia użytkownika (wykonanie przelewu, zmiana adresu, itp.)

- Common Industry Format for Usability Test Raports (ANSI 2001, ISO 2006)
  - Efektywność – wskaźnik % ukończonych zadań
  - Wydajność - średni czas ukończenia zadania + odch. std.
  - Średnia opinia (QoE) użytkowników mierzona np. w skali MOS, po ukończonym zadaniu lub całej sesji testowej.

Testowanie
Badanie opinii użytkowników (MOS, %)

- Satysfakcja (MOS):
  - „Czy obsługa automatyczna była satysfakcjonująca?”
- Postrzegana łatwość użycia (MOS):
  - „Czy aplikacja była łatwa w użyciu?”
- Postrzegana jakość (MOS-X):
  - „Czy głos był zrozumiały i przyjemny?”
- Sukces w pierwszym podejściu (% → 1 / 0):
  - „Czy wykonałeś swoje zadanie?” TAK/NIE
- Czas **do** rozpoczęcia zadania (sek.), im mniejszy tym lepszy. (wstępy, reklamy, przerwy, „ogarnięcie” komunikatu)
- Obiektywna liczba (%) rzeczywiście ukończonych zadań.
- Czas ukończenia zadania (sek.), im mniej tym lepiej.
- Wskaźnik (%) odrzuconych i przerwanych rozmów.

- Badania są drogie, więc trzeba je dobrze zaprojektować. Minimalne ale wystarczające (istotność statystyczna i przedziały ufności).

Rozmiar próby potrzebny do wykrycia problemu (Czy mamy szanse go wykryć?)

Skumulowane prawdopodobieństwo przynajmniej jednokrotnego wykrycia błędu.

P(problemu)	*0.5*	*0.75*	*0.85*	*0.9*	*0.95*	*0.99*
0.01	68	136	186	225	289	418
0.05	14	27	37	44	57	82
0.10	7	13	18	22	28	40
0.15	5	9	12	14	18	26
0.25	3	5	7	8	11	15
0.50	1	2	3	4	5	7
0.90	1	1	1	1	2	2

Analizy wdrożeń

- Sens wdrażania pilotów.
- Analiza raportów i logów z działania systemu.
  - Wskaźnik % opuszczonych połączeń.
  - Wskaźnik % połączeń przekierowanych do agenta.
  - Wskaźnik % ukończonych zadań bez wielokrotnych prób (przekierowań, oddzwaniania itp.).
- Wskaźniki liczone ze względu węzły końcowe IVR.
- Monitorowanie rozmów
  - Na żywo
  - Losowej próby:
    - Rozkład powodu nawiązania rozmowy.
    - Analiza ścieżki rozmowy.
    - Wskaźnik prawidłowych połączeń z agentami.
    - Zaoszczędzony dzięki automatyce czas Agenta liczony na jedno połączenia (sek., min.).

Voice XML 2.1 - wstęp

- *Voice Extensible Markup Language*
- Standard W3C: http://www.w3.org/Voice/
- Pierwsze założenie: dodanie modalności głosowej do sieci Web – rozszerzenie funkcjonalności HTML
- Wykorzystywany przez- i współpracujący z:
  - *Speech Grammar Recognition Specification* (SRGS)
  - ABNF - *Augmented* BNF (Backus-Naur *Form*)
  - *Semantic Interpretation for Speech Recognition* (SISR)
  - *Pronunciation Lexicon Specification* (PLS)
  - *Speech Synthesis Markup Language* (SSML)
  - *Call Control* (CCXML)
  - *State Chart XML* (SCXML)
- Pierwsze prace: At&T 1995 rok (Phone Markup Language) !

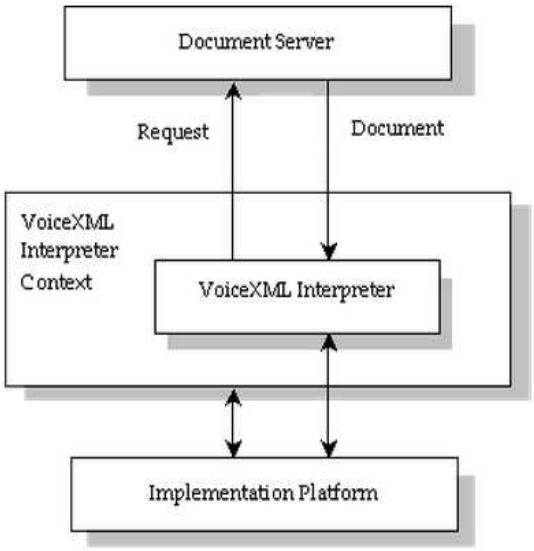
Voice XML

- Zadanie: opis dialogu, uwzględniający:
  - Mowę syntetyczną (TTS)
  - Mowę naturalną zdigitalizowaną (np. WAV)
  - Rozpoznawanie DTMF
  - Rozpoznawanie wymawianych kodów (cyfr/znaków) DTMF
  - Nagrywanie mowy
  - Obsługę połączeń telefonicznych, przekierowań i innych.
  - Obsługę *call-flaw*.

VoiceXML – przykład zapytania

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
   http://www.w3.org/TR/voicexml20/vxml.xsd"
   version="2.0">
  <form>
  <field name="drink">
     <prompt>Czy chcesz kawę, herbatę, mleko albo nic?</prompt>
     <grammar src="drink.grxml" type="application/srgs+xml"/>
  </field>
  <block>
     <submit next="http://www.drink.example.com/drink2.asp"/>
  </block>
 </form>
</vxml>
```

Architektura aplikacji VoiceXML



Zadania Voice XML

- Usługa głosowa rozumiana jako sekwencja interakcji użytkownika z systemem.
- Dialogi udostępnione przez osobny (od serwera obsługującego XML) system (np. bazę wiedzy firmy).
- W efekcie dialog, to wymiana plików VXML zawierających instrukcje i treści.
- Wiele instrukcji w jednym pliku (oszczędność transakcji)
- Jest nośnikiem scenariusza, nie technologią implementacji (np. rozgranicza interakcję użytkownika od technologii dostępu, np. skryptów CGI, WebService itp.)
- Przenośność systemu, niezależność od platformy
- Dobrze sprawdza się dla prostych dialogów jak i skomplikowanych struktur dialogowych.

Aplikacje i dialogi w VoiceXML

- Realizowane przez dokument lub zbiór dokumentów VXML (ten zbiór to właśnie aplikacja)
  - Dokument *root* → nadrzędny
  - i pozostałe dokumenty
- Definiują automat stanów skończonych
- W każdym momencie dialogu użytkownik/system jest w określonym stanie tego automatu. (dot. miejsca jak i wartości)
- Z każdego stanu konieczne jest przejście do kolejnego stanu (dokumentu i dialogu w nim).
- Przejścia zdefiniowane za pomocą URI (*Universal Resource Identifiers*)

Rodzaje dialogów

- Sesja (session) – cała sekwencja interakcji uzytkownika z systemem
- Formularze (*forms*)
  - Formularze służą do wypełnienia zdefiniowanych pól wartościami na podstawie działań użytkownika
  - Wartości wynikają ze zdefiniowanych gramatyk (np. SRGS),
  - Możliwe wypełnienie wielu pól w jednej wypowiedzi (ramki semantyczne)
- Menu
  - Służą do wyboru zdefiniowanego zakresu opcji.
- Pod-dialogi (*subdialog*) – służą do lokalnego zagnieżdżania dialogu. Np. w celu potwierdzenia zgromadzonych w formularzu danych, uproszczenia struktury, itp.

Elementy struktury VXML

- Niektóre elementy (tagi) struktury VXML
  - <audio> - odtwórz klip audio z komunikatem
  - <**choice**> - zdefiniuj menu
  - <assign>, <clear> - przypisz/usuń wartość zmiennej
  - <disconnect>, <exit> - rozłącz sesję, wyjdź
  - <if>, <else>, <elseif>
  - <field>, <option> - definiuje pole wejściowe, opcję w nim
  - <**form**> - utwórz formularz do zbierania danych
  - <goto> - idź do innego dialogu
  - <grammar> - zdefiniuj wykorzystywaną gramatykę SRGS albo DTMF
  - <**menu**> - utwórz menu wyboru alternatywnych opcji
  - <noinput>, <nomatch> - zdefiniuj akcje dla zdarzeń
  - <**prompt**> - odtwórz komunikat (TTS, WAV)
  - <return> - wróć z poddialogu
  - <submit> - wyślij wartości do serwera VXML
  - <transfer> - przekieruj użytkownika
  - <value> - wstaw wartość wyrażenia w komunikacie
  - ...

Przykład dokumentu VXML źródło: W3C VXML 2.0 specif.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
   http://www.w3.org/TR/voicexml20/vxml.xsd"
   version="2.0">
 <meta name="author" content="John Doe"/>
 <meta name="maintainer" content="hello-support@hi.example.com"/>
 <var name="hi" expr="'Hello World!'"/>
 <form>
  <block>
     <value expr="hi"/>
     <goto next="#say_goodbye"/>
  </block>
 </form>
 <form id="say_goodbye">
  <block>
     Goodbye!
  </block>
 </form>
</vxml>
```

Przykład: *root* & *leaf*

leaf – ładowany pierwszy. On wskazuje, że trzeba załadować też *roota*.

```
<vxml ...>
 <var name="bye" expr="'Ciao'"/>
 <link next="operator_xfer.vxml">
   <grammar type="application/srgs+xml" root="root" version="1.0">
     <rule id="root" scope="public">operator</rule>
  </grammar>
 </link>  </vxml>

<vxml ...>
 <form id="say_goodbye">
  <field name="answer">
     <grammar type="application/srgs+xml" src="/grammars/boolean.grxml"/>
     <prompt>Shall we say <value expr="application.bye"/>?</prompt>
     <filled>
       <if cond="answer">
        <exit/>
       </if>
       <clear namelist="answer"/>
     </filled>
  </field>
 </form> </vxml>
```

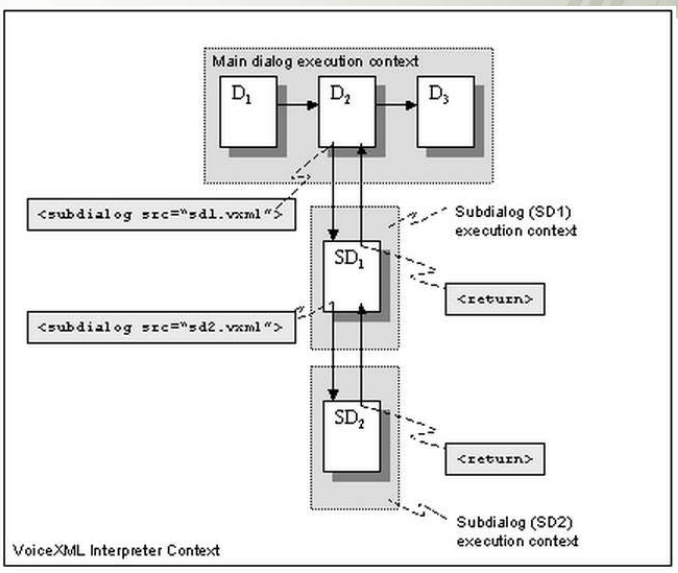
Subdialog – przykład (*root* - app.vxml)

```
<vxml ...>
  <form id="billing_adjustment">
    <var name="account_number"/>
    <var name="home_phone"/>
    <subdialog name="accountinfo" src="acct_info.vxml#basic">
      <filled>        <!-- Note the variable defined by "accountinfo" is
          returned as an ECMAScript object and it contains two
          properties defined by the variables specified in the
          "return" element of the subdialog. -->
        <assign name="account_number" expr="accountinfo.acctnum"/>
        <assign name="home_phone" expr="accountinfo.acctphone"/>
      </filled>
    </subdialog>
    <field name="adjustment_amount">
     <grammar type="application/srgs+xml" src="/grammars/currency.grxml"/>
      <prompt>
        What is the value of your account adjustment?
      </prompt>
      <filled>
        <submit  next="/cgi-bin/updateaccount"/>
      </filled>
    </field>   </form> </vxml>
```

Subdialog – przykład - acct_info.vxml

```
<vxml ...>
  <form id="basic">
    <field name="acctnum">
      <grammar type="application/srgs+xml" src="/grammars/digits.grxml"/>
      <prompt> What is your account number? </prompt>
    </field>
    <field name="acctphone">
      <grammar type="application/srgs+xml"
                          src="/grammars/phone_numbers.grxml"/>
      <prompt> What is your home telephone number? </prompt>
      <filled>
        <!-- The values obtained by the two fields are supplied
          to the calling dialog by the "return" element. -->
        <return namelist="acctnum acctphone"/>
      </filled>
    </field>
  </form>
</vxml>
```

Subdialogi – struktura wywołania



Opcje standardowe do sterowania ASR za pomocą VXML

- *ConfidenceLevel* (0 - 1)
- *Sensitivity* (0 - 1)
- *SpeedVsAccuracy* (0 - 1)
- *CompleteTimeout* (sek., typowo 0.3 – 1 s)
- *IncompleteTimout* (sek)
- *MaxSpeechTimeout* (sek)
- *BargeIn*
- …

- MRCP : http://tools.ietf.org/html/rfc4463
- JSAPI :
  http://jcp.org/en/jsr/detail?id=113
  http://docs.oracle.com/cd/E17802_01/products/products/java-media/speech/forDevelopers/jsapi-doc/

Opcje standardowe do sterowania ASR za pomocą VXML

- *ConfidenceLevel* (0 - 1)
- *Sensitivity* (0 - 1)
- *SpeedVsAccuracy* (0 - 1)
- *CompleteTimeout* (sek., typowo 0.3 – 1 s)
- *IncompleteTimout* (sek)
- *MaxSpeechTimeout* (sek)
- *BargeIn*
- …

- MRCP : http://tools.ietf.org/html/rfc4463
- JSAPI :
  http://jcp.org/en/jsr/detail?id=113
  http://docs.oracle.com/cd/E17802_01/products/products/java-media/speech/forDevelopers/jsapi-doc/

MRCP
Media Resource Control Protocol

- Protokół do kontroli strumieniowania audio/video w środowisku sieciowym
- Zastosowanie do:
  - ASR, TTS, FAX, DTMF, VAD, itd..
- Definiuje
  - Zapytania, odpowiedzi, zdarzenia w komunikacji klient-serwer
- Przesyłany za pomocą innego protokołu, np.. RT(S)P, MIME, JSON, …
-