

# Standard VXML w implementacji systemu dialogowego

**Andrzej WIŚNIEWSKI**

Instytut Teleinformatyki i Automatyki WAT,  
ul. Gen. S. Kaliskiego 2, 00-908 Warszawa  
awis@ita.wat.edu.pl

**STRESZCZENIE:** Przedstawiono sposoby implementacji systemu dialogowego języka mówionego oraz ogólną charakterystykę i pojęcia standardu VXML. Omówiono realizację sterownika dialogu, interpretera języka naturalnego oraz przetwarzania wejścia audio użytkownika za pomocą znaczników języka hipertekstowego VXML.

**SŁOWA KLUCZOWE:** system dialogowy, standard VXML.

## 1. Wstęp

Liczba wdrożonych systemów dialogowych języka mówionego (Spoken Dialogue System, SDS) gwałtownie wzrosła w ciągu ostatnich lat. Wykorzystując aplikacje udostępnione na całym świecie, ludzie wyłącznie za pomocą mowy mogą uzyskać dostęp do zasobów Internetu w celu wykonania użytecznych zadań (np. uzyskać informacje, odczytać zawartość poczty elektronicznej, zaplanować podróż).

System dialogowy języka mówionego łączy technologie automatycznego rozpoznawania mowy, przetwarzania języka naturalnego, sterowania dialogiem i syntezy mowy do utworzenia efektywnego interfejsu człowiek-komputer.

Projektowanie interfejsów głosowych do niedawna było zadaniem dla ekspertów: wymagało umiejętności w zakresie technologii mowy i programowania niskopoziomowego. Wiele trudności sprawiało szybkie projektowanie i ponowne wykorzystanie zaprojektowanych systemów dialogowych. Z tych powodów interfejsy głosowe są mniej rozpowszechnione i w związku z tym mniej intuicyjne niż interfejsy graficzne. Aby ułatwić projektowanie systemów dialogowych języka mówionego i zwiększyć użyteczność interfejsów

głosowych, należy doskonalić metody ich projektowania, implementacji, ewaluacji, automatycznej adaptacji do nowych zastosowań oraz symulacji. Istotnym czynnikiem postępu w tej dziedzinie stało się opracowanie standardu języka skryptowego VoiceXML (Voice eXtensible Markup Language), który w znaczący sposób obniżył bariery w projektowaniu i wdrażaniu aplikacji głosowych oraz umożliwia tworzenie SDS przez nieprofesjonalistów.

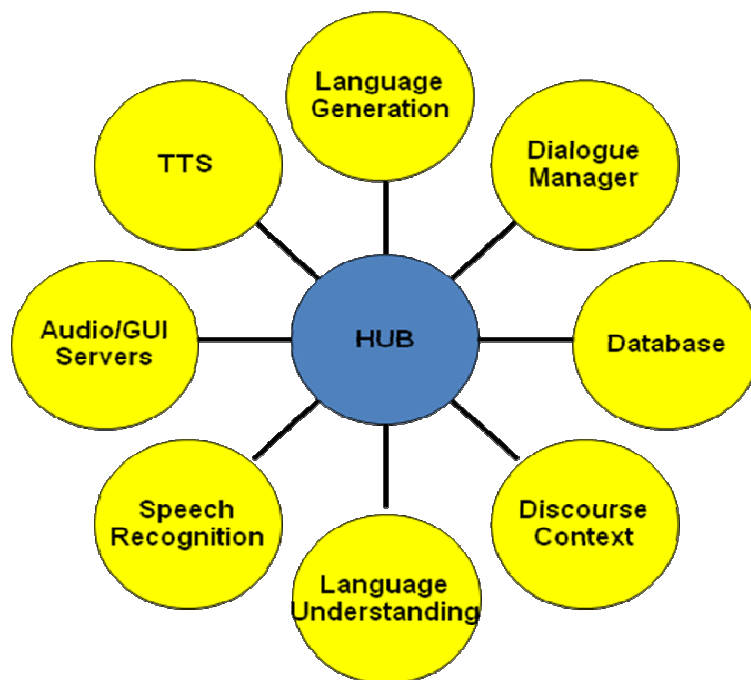
## 2. Implementacja systemu dialogowego

Implementacji systemu dialogowego języka mówionego dokonuje się obecnie albo wykorzystując środowiska programistyczne (infrastruktury) umożliwiające łączenie obiektów czy wydzielonych modułów systemu dialogowego, albo stosując jako ogólny mechanizm języki skryptowe (np. standard VXML).

Przykładem złożonego środowiska programistycznego jest SDS Galaxy-II, realizowany w ramach programu DARPA Communicator przez MIT Laboratory for Computer Science [3, 4]. Architektura SDS Galaxy-II (rys. 1) integruje technologie mowy oraz wydzielone moduły i umożliwia wytwarzanie systemów dialogowych języka mówionego.

Zastosowana architektura klient/serwer umożliwia użytkownikom komunikację za pomocą niewielkiego programowo klienta z systemem specjalizowanych serwerów, które realizują złożone obliczeniowo zadania, jak rozpoznawanie mowy, rozumienie języka, dostęp do bazy danych i syntezę mowy. Zasadniczym elementem systemu jest centralny hub, który steruje przepływem danych między różnymi klientami i serwerami oraz przechowuje stan i historię bieżącego dialogu.

Ten specyficzny hub uaktualnia listę aktywnych serwerów, określa host, port i zbiór operacji realizowanych przez każdy serwer oraz wykonuje zestaw programów. Każdy program składa się ze zbioru reguł, z których każda specyfikuje operację, zestaw warunków uaktywniających regułę, listę zmiennych wejściowych i wyjściowych, jak również opcjonalnie zmienne związane z historią dyskursu. Uaktywnienie reguły powoduje wysyłanie do serwera tokena ze zmiennymi wejściowymi, wykonanie operacji i zwrot tokena ze zmiennymi wyjściowymi, które są zapisywane w głównym tokenie centralnego huba.



Rys. 1. Architektura SDS Galaxy-II

Specjalizowane serwery mogą być uruchomione na tej samej maszynie lub w odległym miejscu. Interakcja huba i serwerów jest sterowana za pomocą specjalizowanego języka skryptowego poprzez standaryzowany szkieletowy protokół.

Należy również wspomnieć o innych, rozwijanych w ostatnich latach infrastrukturach: Open Agent Architecture [2], TRIPS [1], TRINDI [5].

Odmiennym sposobem implementacji systemu dialogowego języka mówionego jest zastosowanie standardu VXML do tworzenia aplikacji głosowych w postaci dokumentów vxml, a następnie wykorzystanie specjalizowanych portali głosowych do ich wykonywania.

To podejście rozwijane jest intensywnie ze względu na zastosowania komercyjne, których głównym celem jest:

- 1) umożliwienie każdemu, kto mówi i słyszy, wykorzystanie aplikacji bez uprzedniego uczenia;
- 2) umożliwienie doświadczonemu rozmówcy wykonanie zadania szybko i efektywnie.

Portale głosowe umożliwiają dostęp do informacji biznesowych (zapytania rozmówców o osoby i rzeczy, zautomatyzowane zamawianie usług za pomocą telefonu, pomoc, śledzenie zamówień, informacja lotnicza, rezerwacja

miejsc w kinach i teatrach, usługi bankowe), publicznych (pogoda, warunki drogowe, wskazówki i wypadki, wiadomości, informacje giełdowe i gospodarcze), osobistych (kalendarz, adresy, telefony, sporządzanie list, listy zakupów, liczniki kalorii) oraz pomoc użytkownikom w komunikacji z innymi ludźmi za pomocą poczty głosowej i elektronicznej. Realizują zadania usługowe i, w mniejszym stopniu, obliczeniowe.

### 3. Historia standardu VXML [12]

W połowie lat 90 XX wieku w firmie AT & T Bell Laboratories został zainicjowany projekt badawczy „Phone Web”. Jego celem było utworzenie podobnego do HTML języka znaczników głosowych (Phone Markup Language, PML). Po podziale AT & T Bell Labs na AT & T Labs i Lucent Bell Labs obie firmy kontynuowały prace nad PML – powstały dwie wersje PML.

W 1998 Motorola Inc. opracowała język VoxML, przyjęto założenie, że użytkownik jest mobilny – stąd skupienie się na rozwoju technologii rozpoznawania mowy (a nie wybierania tonowego) w interfejsie wejściowym.

Wkrótce potem kolejne firmy ogłosiły powstanie swoich wersji języka znaczników głosowych: IBM o nazwie SpeechML, Hewlett-Packard – TalkML, Sun Mikrosystem – JSML (Java Speech Markup Language), PipeBaech – VoiceHTML, Vocalis – SpeechHTML. Cechą wspólną wszystkich wersji było istnienie mechanizmów udostępniania zasobów Internetu przez telefon, każdy z nich posiadał jednocześnie cechy indywidualne.

2 marca 1999 AT & T Labs, Lucent Bell Labs, Motorola wraz z 17 innymi firmami ogłosiły powstanie VoiceXML Forum, którego celem było zdefiniowanie standardowego języka projektowania dialogów języka mówionego oraz dostępu do zasobów Internetu przez telefon. Po roku prac, w marcu 2000 roku, ukazała się wersja VoiceXML 1.0 – wtedy VoiceXML Forum liczyło już 79 członków. W maju 2000 roku W3C (World Wide Web Consortium) przyjęło VoiceXML jako światowy standard oraz VoiceXML 1.0 jako podstawę do rozwijania dialogowego języka znaczników. Od tej pory W3C zajmuje się dalszymi pracami nad rozwojem VoiceXML, a VoiceXML Forum – edukacją i propagowaniem specyfikacji języka oraz akcjami marketingowymi mającymi na celu pozyskiwanie nowych użytkowników.

16 marca 2004 roku W3C zarekomendowało wersję VoiceXML 2.0 [9], w której usunięto błędy specyfikacji poprzedniej wersji oraz opracowano nowe standardy rozpoznawania i syntezy mowy. We wrześniu 2006 r. W3C przyjęło wersję roboczą (working draft) VoiceXML 2.1 [10].

Od roku 2007 trwają prace nad wersją roboczą VoiceXML 3.0 [11], których koniec zapowiadany jest na IV kwartał 2010 r.

#### 4. Ogólna charakterystyka standardu VXML

Celem twórców języka hipertekstowego VoiceXML (VXML) było udostępnienie aplikacjom głosowym zasobów (wydarzeń, treści) sieciowych (internetowych, intranetowych). Powstał standard, który:

- stanowi podstawę budowy interfejsu głosowego (HTML dla aplikacji telefonicznych);
- umożliwia użytkownikowi decydowanie o kierunku nawigacji przez wybór jednej z oferowanych opcji;
- przypisuje zmiennym podane wartości, takie jak: dane osobiste, słowa kluczowe, identyfikatory czy hasła;
- umożliwia określanie właściwości generowanego głosu, na przykład wysokości (częstotliwości tonu podstawowego) czy szybkości czytania.

VXML jest zgodny z XML, zatem możliwe jest, za pomocą klasycznych technik internetowych, integrowanie zawartości portali głosowych z zawartością serwisów przeznaczonych dla terminali wizualnych (WWW, WAP). Standard umożliwia realizację następujących zadań szczegółowych:

- na wejściu: rozpoznawanie sygnałów DTMF i języka mówionego, rejestrowanie (bez rozpoznawania) wejściowej wypowiedzi;
- na wyjściu: generowanie mowy syntetycznej (umożliwia określanie właściwości generowanego głosu), odtwarzanie plików dźwiękowych;
- sterowanie przebiegiem dialogu;
- reagowanie na zdarzenia w sieci telefonicznej: przeniesienie i rozłączenie rozmowy;
- przesyłanie informacji użytkownika na strony sieciowe lub inne serwery;
- uzyskiwanie informacji z Internetu i przekazywanie jej użytkownikowi;
- wykonywanie typowych funkcji programowania: operacji logicznych i arytmetycznych, operacji na tekście.

Przy tworzeniu aplikacji głosowej, będącej implementacją systemu dialogowego języka mówionego za pomocą języka VXML, istotne są następujące cechy tego języka:

- oddzielanie kodu interakcji z użytkownikiem (dokumenty VXML) od logiki serwisowej (np. skrypty CGI) – ukrywa wiele niskopoziomowych detali i umożliwia tworzenie aplikacji głosowych za pomocą wysoko-poziomych dialogów;
- integrowanie usług głosowych i usług wykorzystujących tradycyjne dane (stosując architekturę klient-serwer);
- łączenie różnych dokumentów aplikacji za pomocą URI (Universal Resource Identifiers);

- przystosowanie formatów plików akustycznych, gramatyk i schematów URI do różnorodnych platform – od twórców platform głosowych wymaga jednego wspólnego formatu gramatyki (Speech Recognition Grammar Specification, SRGS) i plików audio (do rejestrowania i odtwarzania), jednocześnie platformy mogą udostępniać różne inne formaty gramatyk i formaty audio;
- przesyłanie danych na serwer metodami HTTP (GET lub POST) – umożliwia sposób identyfikacji danych wysyłanych na serwer;
- definiowanie zasobów i własności, które są dostępne w całej aplikacji – dokument źródłowy aplikacji, który zawiera: deklaracje zmiennych, skrypty, łączniki, gramatyki, reakcje na wydarzenia czy własności dostępne we wszystkich dokumentach tworzących aplikację;
- interpretacja semantyczna wypowiedzi (wejść) użytkownika na podstawie gramatyki.

## 5. Pojęcia VXML

Sesja zaczyna się, gdy użytkownik rozpoczyna interakcję z interpreterem VXML, trwa, gdy dokumenty VXML są ładowane i interpretowane, kończy się na żądanie użytkownika, dokumentu VXML lub kontekstu interpretera, który reaguje na zdarzenia w platformie implementacyjnej aplikacji.

Można traktować aplikację głosową, będącą zbiorem powiązanych dokumentów VXML, jako konwersacyjną maszynę o skończonej liczbie stanów. Użytkownik w dowolnej chwili czasowej jest zawsze w jednym z tych nazwanych stanów dialogu. Przejścia do następnego dialogu określane są za pomocą URI, które wyznaczają następny dokument i dialog do przeprowadzenia. Jeżeli URI nie dotyczy dokumentu, przyjmowany jest bieżący dokument. Jeżeli URI nie dotyczy dialogu, przyjmuje się pierwszy dialog w tym dokumencie. Wykonywanie jest przerywane, gdy aktualnie realizowany dialog nie specyfikuje następnego lub gdy zawiera element, który kończy rozmowę.

Dialog może przyjąć postać formularza (znacznik `<form>`) lub menu (`<menu>`). Menu umożliwia użytkownikowi wybór jednej z wielu opcji oraz przejście do innego stanu dialogu zgodnie z wyborem. Jednak głównym sposobem budowania dialogu jest formularz, który definiuje interakcję z użytkownikiem.

Formularz składa się z przynajmniej jednego pola (`<field>`), którego atrybutem jest nazwa zmiennej (jest to, wspomniany wcześniej, nazwany stan dialogu), gdzie zapisane zostanie wejście użytkownika (informacja, dane) jako łańcuch tekstowy. Każde pole może specyfikować podpowiedź (`<prompt>`),

która precyzuje treść oczekiwanej odpowiedzi (wejścia) użytkownika i jej spodziewaną postać (wypowiedź słowną lub wybór opcji za pomocą przycisku na klawiaturze telefonu – DTMF).

Każde pole (<field>) może specyfikować gramatykę (<grammar>), która definiuje dopuszczalne wejścia dla tego pola. Po wprowadzeniu przez użytkownika danych do pól formularza, za pomocą znacznika <field> można określić dalszy przebieg sterowania dialogiem (np. przejście do innego formularza lub do następnego dokumentu).

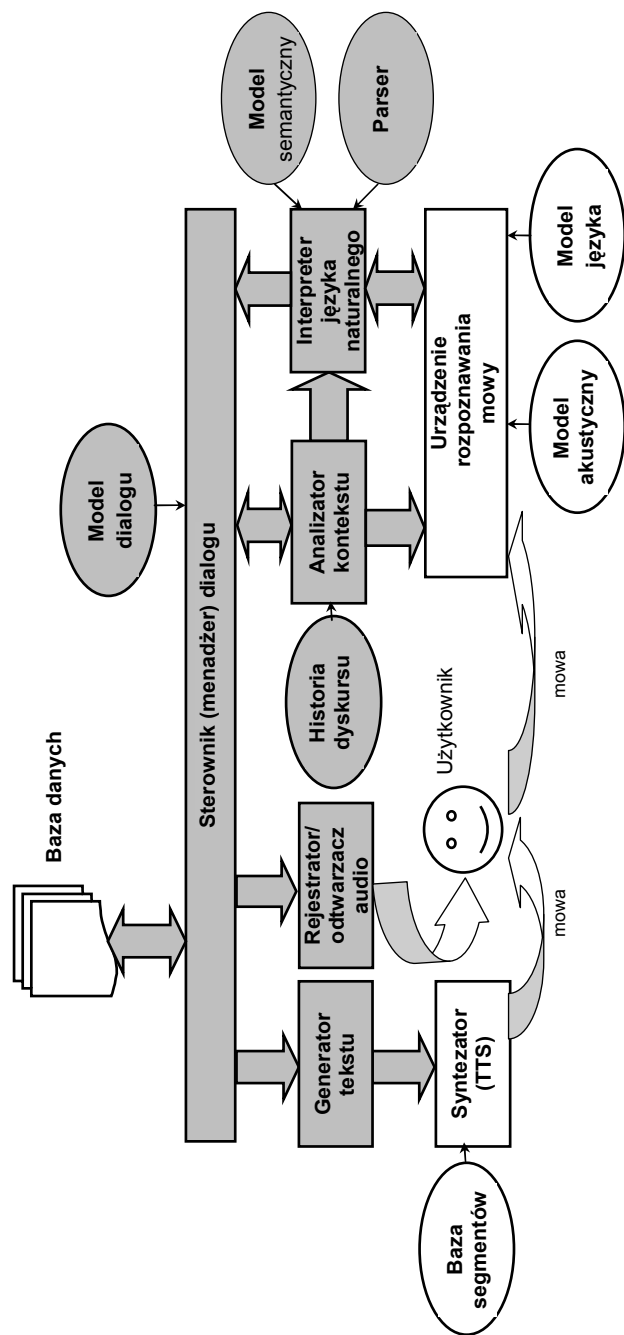
Do tworzenia biblioteki dialogów dla wspólnych zadań (mogą być wykorzystane przez różne dokumenty tej samej aplikacji) stosuje się subdialogi (<subdialog>). Subdialog jest jak wywołanie funkcji, umożliwia wywołanie nowego dialogu i powrót do pierwotnego z zachowaniem informacji lokalnej – zmienne, gramatyki, informacje o stanie są przechowywane i dostępne po powrocie do wywołującego dokumentu.

## **6. Znaczniki VXML w implementacji systemu dialogowego języka mówionego**

W [6, 7] omówiony został system dialogowy języka mówionego jako interfejs głosowy systemu komputerowego, umożliwiający dialog (interakcję) między użytkownikiem-człowiekiem i sterownikiem (menadżerem) systemu dialogowego, wykorzystujący do tego mowę. Architekturę systemu dialogowego pokazano na rys. 2 [6], kolorem szarym oznaczone zostały podsystemy przetwarzania mowy i języka, których implementację umożliwia standard VXML.

Automatyczne rozpoznawanie mowy (realizowane przez urządzenie rozpoznawania mowy, wykorzystujące modele akustyczne rozpoznawanych jednostek fonetycznych i model języka) oraz synteza sygnału mowy (realizowana za pomocą syntezy TTS, wykorzystującego bazę segmentów do konkatencji sygnału akustycznego) to elementy platformy implementacyjnej i są dostarczane przez portal głosowy. Standard VXML jedynie w ograniczonym zakresie dostarcza narzędzi sterowania zarówno rozpoznawaniem jak i syntezą sygnału mowy.

Sterownik dialogu ma zwykle dostęp do zewnętrznego źródła informacji – bazy danych.



Rys. 2. Architektura systemu dialogowego języka mówionego



## 6.1. Sterownik dialogu

Zadania sterownika dialogu są związane z przebiegiem dialogu i istotnie zależą od jego stadium i stanu:

- gromadzi informacje i dane od użytkownika (w początkowym stadium dialogu);
- wyjaśnia niejasności spowodowane błędami rozpoznawania i brakiem pełnej informacji o realizowanym zadaniu;
- generuje zapytania bazodanowe – proponuje dodatkowe ograniczenia, gdy objętość informacji jest za duża (aby zmniejszyć liczbę opcji);
- gdy potrzeba negocjuje z użytkownikiem, po otrzymaniu informacji z bazy danych sugeruje redukcję informacji;
- generuje tekst odpowiedzi.

U podstaw sterowania dialogiem leży modelowanie dialogu. Technologia wytwarzania modelu dialogu jest jedną z najmniej dopracowanych w zakresie technologii języka naturalnego. Modelowanie dialogu może oznaczać albo planowanie działań i rozwiązywanie problemów interakcji, albo przygotowanie systemu do każdej transakcji (odpowiedź słowna, graficzna i za pomocą tablicy, jak również żądanie wyjaśnień).

Standard VXML umożliwia realizację modelu dialogu rozumianego jako przeprowadzenie wszystkich wcześniej przewidzianych transakcji. Informację lub daną pozyskaną od użytkownika w pojedynczej transakcji gromadzi się w zmiennej o nazwie przypisanej polu formularza `<field>`. Porządkiem interpretacji dialogu zbudowanego z sekwencji transakcji można sterować za pomocą znaczników `<if>` - `<then>` - `<else>`, `<goto>`, `<foreach>` (jak w innych językach wysokopoziomowych).

Dodatkowo język VXML dostarcza jeszcze jednego sposobu sterowania porządkiem interpretacji dialogu: wszystkie obiekty formularza mogą posiadać atrybuty „cond” i „expr”. Interpreter wykonuje wyłącznie te obiekty formularza, których:

- wartość atrybutu „expr” jest niezdefiniowana (zastosowanie „expr” do nadania wartości początkowej obiektowi formularza spowoduje zignorowanie tego obiektu przez interpreter);
- atrybut „cond”, zwany warunkiem ochrony, ma wartość „true”.

Ten sposób sterowania dialogiem jest szczególnie przydatny, gdy sterownik dialogu wyjaśnia niejasności spowodowane błędami rozpoznawania i uzupełnia braki informacji o realizowanym zadaniu.

Aplikacje głosowe prawie zawsze wymagają połączenia z serwerami. Najczęściej jest to przesłanie zapytania bazodanowego. Standard VXML zapewnia dwa sposoby przesyłania danych z dokumentu VXML do serwera:

- użycie znacznika `<submit>`, wówczas domyślnie wysyłane są do określonego miejsca URL wartości formularza za pomocą metody GET (można zastosować metodę POST – do określania metody służy atrybut „method”), dodatkowo za pomocą atrybutu „namelist” można kontrolować przesyłane pola formularza;
- dodanie danych źródłowych do URI.

Generowanie tekstu pytań wyjaśniających oraz tekstu odpowiedzi wykonuje się w języku VXML za pomocą znacznika `<prompt>`.

## 6.2. Interpreter języka naturalnego

Rozumienie języka w systemie dialogowym realizowane jest przez interpreter języka naturalnego, który może do tego celu wykorzystywać, w zależności od stopnia złożoności, parser do analizy syntaktycznej oraz model semantyczny w celu określenia znaczenia wypowiedzi.

Komunikacja werbalna między ludźmi, która jest procesem dwukierunkowym dotyczącym aktywnych uczestników, nazywa się dyskursem. Zdolność analizowania dyskursu (prowadzonego przez analizator kontekstu, który używa do tego formalnie zapisanej historii dyskursu) umożliwia systemowi dialogowemu zrozumienie wypowiedzi w kontekście poprzednich interakcji. Jest to ważne zarówno dla zrozumienia języka naturalnego, jak również dla rozpoznawania mowy. Możliwość dziedziczenia informacji z poprzednich wypowiedzi jest szczególnie pomocna w obliczu błędów rozpoznawania. Na przykład, użytkownik może zadać złożone, wymagające kilku atrybutów (wyróżników) pytanie – urządzenie rozpoznawania może nie rozpoznać pojedynczego słowa w tej wypowiedzi, np. numeru lotu lub czasu przylotu w pytaniu o cenę biletu lotniczego. Jeśli istnieje model kontekstowy, system najpierw wygeneruje pytanie uzupełniające, użytkownik wypowie krótką frazę korekcyjną, a następnie system będzie potrafił wstawić do wypowiedzi tylko źle zrozumiane słowo, zapobiegając konieczności powtarzania całej wypowiedzi i zmniejszając w ten sposób ryzyko kolejnych błędów rozpoznawania i rozumienia.

Głównym narzędziem interpretacji języka naturalnego w standardzie VXML jest gramatyka, definiowana za pomocą znacznika `<grammar>`. Gramatyka może być predefiniowana, specyfikowana w dialogu, zewnętrzna, definiowana w dokumencie i oznaczona jako globalna, oraz definiowana w dokumencie źródłowym i wtedy jest aktywna w całej aplikacji (we wszystkich jej dokumentach). Gramatyka dostarcza reguł interpretacji semantycznej wejścia

użytkownika. Jeżeli istnieje gramatyka na poziomie formularza (a nie na poziomie pola), może być zastosowana do wypełnienia kilku pól na podstawie jednej wypowiedzi (przemienna inicjatywa).

Mankamentem gramatyki w najprostszej postaci jest fakt, że rozpoznana fraza w całości zostanie umieszczona w zmiennej wejściowej (polu). Aby gramatyka przypisywała zmiennej wejściowej tylko jedno ze słów frazy (słowo kluczowe), należy zastosować slot – jest to główny sposób interpretacji semantycznej wejścia (wypowiedzi) użytkownika.

W aplikacjach głosowych implementujących dialog o przemiennej inicjatywie [7], gdzie użytkownik i system dialogowy zamieniają się w określaniu następnej czynności, można uaktywnić gramatykę (tj. postawić ją w stan nasłuchiwania), aby była dostępna we wszystkich dokumentach aplikacji. Wtedy gdy użytkownik, będąc w innym dialogu tej aplikacji, użyje słowa zdefiniowanego w uaktywnionej gramatyce, sterowanie dialogiem automatycznie zostanie przekazane do dialogu z tą uaktywnioną gramatyką – np. zostanie wypełnione pole w odpowiednim formularzu. Przemienna inicjatywa powiększa elastyczność i siłę aplikacji głosowych. Dialog o przemiennej inicjatywie w standardzie VXML realizowany jest za pomocą specjalnego formularza. Formularz o przemiennej inicjatywie ma specyficzną strukturę – gramatyka zdefiniowana jest dla całego formularza, umożliwia rozpoznawanie różnorodnych wejść użytkownika oraz pozwala na dostarczanie wartości do więcej niż jednego pola podczas pojedynczego wejścia użytkownika. Jest to cecha języka naturalnego w dialogu człowiek – człowiek implementowana przez standard VXML.

### 6.3. Rejestracja i odtworzenie wejścia użytkownika

Można nagrać i przechowywać wejście audio użytkownika (dla poczty głosowej lub podobnych aplikacji, aby zarejestrować wypowiedź użytkownika w celu późniejszego odsłuchania). Standard VXML dostarcza znaczników:

- <record>, który przyjmuje wejście od użytkownika, nie rozpoznaje go, rejestruje jako plik dźwiękowy;
- <audio>, który odtwarza nagraną wiadomość dźwiękową.

Można zagnieździć podpowiedź <prompt> w <audio> – wtedy nastąpi odtworzenie wskazanego pliku dźwiękowego lub, gdy tego pliku nie znajdzie się, urządzenie TTS dokona syntezy tekstu podpowiedzi.

## 7. Uwagi końcowe

Zaprojektowanie użytecznej aplikacji VXML jest wciąż bardziej sztuką niż inżynierią. Podręczniki języka nie proponują standardowych sposobów tworzenia (wyrażania) podpowiedzi (<prompt>), gramatyk opisujących zawartość wypowiedzi rozmówcy w reakcji na podpowiedź, ani reakcji systemu na niewłaściwą odpowiedź rozmówcy.

Standard VoiceXML uwzględnia i spełnia wymagania większości usług typu IVR (Interactiv Voice Responce), jednak dialogi i usługi z większymi wymaganiami lepiej realizować za pomocą dedykowanych aplikacji.

## Literatura

- [1] ALLEN J., BYRON D., DZIKOVSKA M., FERGUSON G., GALESCU L., STENT A., *Towards Conversational Human-Computer Interaction*, AI Magazine 22 (4), 2001, str. 27-37.
- [2] MARTIN D.L., CHEYER A.J., MORAN D.B., *The Open Agent Architecture: A Framework for Building Distributed Software Systems*, Applied Artificial Intelligence, Vol. 13, January-March, 1999, str. 91-128.
- [3] POLIFRONI J., SENEFF S., *GALAXY-II as an Architecture for Spoken Dialogue Evaluation*, Proc. Second International Conference on Language Resources and Evaluation (LREC), Athens, Greece, May 31-June 2, 2000, str. 725-730.
- [4] SENEFF S., HURLEY E., LAU R., PAO C., SCHMID P., ZUE V., *GALAXY-II: A Reference Architecture for Conversational System Development*, Proc. ICSLP 98, Sydney, Australia, November 1998, Vol. 3, str. 931-934.
- [5] TRAUM D., LARSSON S., *The Information State Approach to Dialogue Management. Current and New Directions in Discourse and Dialogue*, Kluwer Academic Publishers, 2003.
- [6] WIŚNIEWSKI A., *System dialogowy języka mówionego – przegląd problemów*, Biuletyn Instytutu Automatyki i Robotyki WAT, 24, 2007, str. 97-122.
- [7] WIŚNIEWSKI A., *Dialog o przemiennej inicjatywie w systemie dialogowym języka mówionego*, Biuletyn Instytutu Automatyki i Robotyki WAT, 27, 2009, str. 97-113.
- [8] <http://www.voicexml.org/voicexml-tutorials>
- [9] <http://www.w3.org/TR/voicexml20/>
- [10] <http://www.w3.org/TR/voicexml21/>
- [11] <http://www.w3.org/TR/voicexml30/>
- [12] <http://www.voicexml.org/>

## **VoiceXML standard in implementation of a spoken dialogue system**

ABSTRACT: In this paper, methods of implementation of a spoken language dialogue system, general characteristics and concepts of VXML standard are presented. Implementations of a dialogue manager, a natural language interpreter, and an audio input processing with the use of hypertext markup language VXML are discussed.

KEYWORDS: dialogue system, VoiceXML.

*Praca wpłynęła do redakcji: 29.12.2010*