# Hangos jelölőnyelvek
## Jelölőnyelvek a beszéd alapú alkalmazások fejlesztésében

ABARI KÁLMÁN

*Debreceni Egyetem, Pszichológiai Intézet és Matematikai-Számítástudományi Doktori Iskola*
*abarik@delfin.unideb.hu*

***Lektorált***

***Kulcsszavak: beszéd alapú alkalmazás, szabványok, SALT, SRGS, SSML, VoiceXML***

*Az utóbbi években a beszéd alapú alkalmazások fejlesztésében az egyéni megközelítések helyét fokozatosan az ipari szabványokon alapuló stratégiák és architektúrák veszik át. Különösen igaz ez a telefonos és a multimodális alkalmazásokra, melyek fejlesztését mára majd egy tucat XML alapú jelölőnyelv segíti. A cikkben összefoglaljuk a beszéd alapú alkalmazások egyes komponenseit és azok kommunikációját leíró jelölőnyelveket.*

## 1. Bevezetés

Az elmúlt évek hatalmas technológiai fejlődése ellenére a beszéd alapú alkalmazások fejlesztése összetett feladat, hiszen olyan bonyolult technológiák integrációjára van szükség, mint például a beszédfelismerés, beszédszintézis és dialógusvezérlés. A régebbi alkalmazások elsődlegesen fejlesztőik egyéni megoldásain alapultak, habár a különböző nyílt programozási felületek (API-k) megjelenése – például SAPI (Microsoft Speech Application Program Interface), JSAPI (Java Speech API) – jelentősen csökkentette az alkalmazásfejlesztés bonyolultságát.

Az 1990-es évek végétől aztán egy igen kedvező folyamat indult el: az egyéni megközelítések helyét fokozatosan az ipari szabványokon alapuló stratégiák és architektúrák veszik át. Ennek a szabványosítási folyamatnak a legjelentősebb hajtómotorja a webes és a telefonos világ összekapcsolásának igénye volt. Az áhított cél, hogy ugyanazok a szolgáltatások, amelyeket az ügyfelek eddig hagyományosan grafikus felületről értek el, ezután telefonon keresztül, a meglévő webes infrastruktúrával együttműködve, hang alapú kérések formájában is hozzáférhetőek legyenek. Az integrációs törekvés szimmetrikus, tehát az a cél, hogy az adatbevitel grafikus és hangalapú módon egyaránt megtörténhessen. Ennek érdekében az utóbbi nyolc évben majd egy tucat jelölőnyelvet fejlesztettek ki, melyek a beszéd alapú alkalmazások egyes részeinek szabványos leírását teszik lehetővé. E cikkben ezeket a „hangos” jelölőnyelveket tekintjük át.

## 2. Testületek

A szabványok alkalmazása a beszéd alapú alkalmazások fejlesztésében – azon túl, hogy jelzik, a terület kezd nagykorúvá válni – számos előnnyel jár. Elrejtik a technológiai részleteket, biztosítják a különböző szállítóktól érkező komponensek együttműködését, kevesebb időbefektetés és kisebb erőfeszítés mellett újrafelhasználható és hordozható megoldások létrehozását támogatják. Másfelől azonban a fejlesztők korlátozva érezhetik a kreativitásukat és bosszankodhatnak, ha valamely funkciót az adott szabvány (még) nem támogatja.

Szabvány alatt a továbbiakban olyan leírást értünk, melyet valamely szabványosításért felelős testület formálisan elismert. A beszéddel kapcsolatos területen a következő szervezetek a legaktívabb:

• A **W3C (World Wide Web Consortium)** hagyományosan vezető szerepet játszik a webes technológiák kifejlesztésében, a Webben rejlő lehetőségek minél teljesebb kihasználásában. Az egyes specifikációk kidolgozása munkacsoportokban történik, melyet a W3C tagjai alkotnak. Egy többlépcsős folyamat eredménye (munkaterv, utolsó felhívás munkatervre, előzetes javaslatterv, javaslatterv, ajánlás) míg egy specifikációból W3C-ajánlás lesz, amelyre a webes társadalom és az ipar már szabványként tekint [5]. A beszéd és multimodális alkalmazások területén két munkacsoport végez fejlesztést, a Voice Browser Working Group (Hangböngésző Munkacsoport) és a Multimodal Interaction WorkGroup (Multimodális Interakció Munkacsoport).

• Az **IETF (Internet Engineering Task Force)** célja az Internet működésének és fejlődésének előmozdítása, az egyes protokollok használatának szabályozása. A Speech Services Control (SpeechSC) munkacsoport az elosztott környezetben működő biztonságos beszédfeldolgozás szabványaiért felelős.

• A **ETSI (European Telephony Standards Institute)** célja azon szabványok kidolgozása, amelyek biztosítják, hogy a globális távközlési piac egyetlen piacként működjön. Az Aurora projekt a mobilhálózaton megvalósuló elosztott beszédfelismerés szabványosításán dolgozik.

Két további vállalati összefogáson alapuló „fórum” is meghatározó szerepet játszik ezen a területen:

• A **VoiceXML Forum** olyan nagyvállalatok összefogásából alakult ki, melyek mindegyikének korábban megvolt a saját ötlete a hang alapú webes szolgáltatásra. Ez az AT&T és a Lucent Technologies vállalatok PML specifikációja, a Motorola SpeechML-je és az IBM Vox-

ML-je volt. Mivel érdekeltek voltak az egységes hangvezérelt Web létrehozásában, közösen elkészítették a VoiceXML 1.0-s változatát, amit 2000 márciusában bemutattak a W3C-nak [9]. Azóta a fórum nem vesz részt a nyelv továbbfejlesztésében, munkája az oktatásra és a webes technológiák népszerűsítésére korlátozódik.

• A **SALT Forum**, amely a Cisco, Comverse, Intel, Microsoft, Philips és Scansoft összefogásából jött létre 2001-ben. Közösen dolgozták ki a SALT (Speech Application Language Tags) 1.0-s változatát, melyet 2002-ben bemutattak a W3C-nak [8].

## 3. Architektúrák

A Web által kínált információk hagyományos elérési módja a személyi számítógépek grafikus felülete, mely a kommunkáció során a „rámutatás" (point and click) elvet követi, néha a billentyűzetet használja adatbevitelre. A hang alapú interfész ehhez képest a mindennapos beszélgetésben megszokott, „beszélek és hallgatok" stílust követi, olyan eszközöket felhasználva, mint szóbeli utasítok, előre felvett beszéd visszajátszása, szintetizált beszéd, és szükség esetén a telefonok nyomógombjai. Irodai környezetben a vizuális felület használata a leghatékonyabb, ahol rendelkezésre áll szélessávú átviteli csatorna, nagyfelbontású képernyő, egér és billentyűzet. A hang alapú felület akkor a leghasznosabb, amikor távol vagyunk az íróasztalunktól, illetve egyes speciális felhasználói csoportoknak, mint például a látássérültek és látáskorlátozottak. Ha tehát a webes szolgáltatások univerzális elérését akarjuk biztosítani, akkor mindkét megközelítési módot, a vizuális és a hang alapú felületet is támogatnunk kell.

Négy alapvető módszert ismerünk, melyek segítségével grafikus és hang alapú felhasználói felület is biztosítható webes alkalmazásunkhoz:

– **Különállóan megtervezett grafikus- és hanginterfész,** melyek ugyanazokra az adatokra és üzleti logikára támaszkodnak, de egymástól függetlenül lettek kifejlesztve.
– **A hagyományos vizuális böngésző „meghangosítása",** mely során grafikus böngészőnk az épp megjelenített lap tartalmát fel tudja olvasni, és szóbeli utasítások segítségével oldalak közötti navigációra is képes.
– **Átkódolás (transcodig),** mely során a meglévő (X)HTML dokumentumokból automatikusan állítunk elő hang alapú interfészt.
– **Kombinált grafikus- és hanginterfész,** ahol minden egyes oldal tartalmaz a grafikus és a hang alapú felhasználói felületre is információt. Ez nem multimodális interfészt jelent, hiszen egyszerre csak az egyik modalitás használható.

A kombinált grafikus- és hanginterfészt bonyolult tervezés, implementálás és karbantartás jellemzi, a „meghangosított" vizuális böngésző és az átkódolási technika esetén pedig nehezen biztosítható a vegyes kezdeményezésű dialógusvezérlés. Ezeket a hátrányokat a grafikus felülettől függetlenül megtervezett hanginterfészek kiküszöbölik, így nem érzékenyek a vizuális interfész változására és a vezérlés jellege is tetszőlegesen megválasztható. A következő pontban ismertetendő szabványok és nyílt specifikációk az ilyen különállóan megtervezett hangalapú felületek fejlesztését támogatják, melyeket többnyire a nyomógombos bevitelt beszédfelismeréssel kombináló *telefonos alkalmazások* körében használhatjuk. Néhány szabvány *multimodális alkalmazások* létrehozását is támogatja, melyek a beszéd feldolgozásán túl az olyan hagyományos perifériák párhuzamos használatát biztosítják, mint például egér, billentyűzet és képernyő.

### Rövidítések

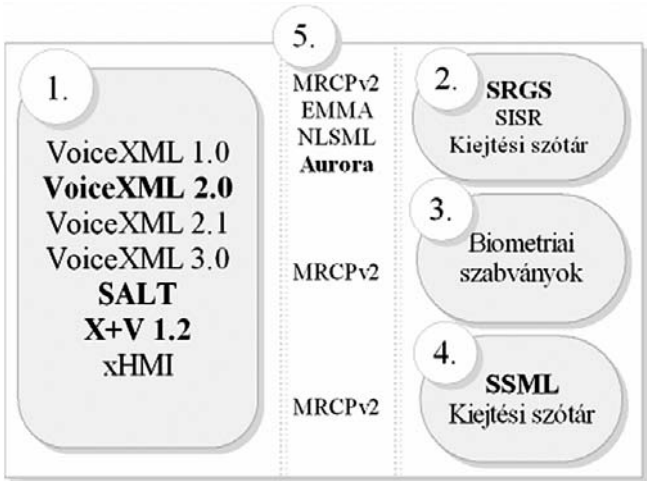
| | |
|---|---|
| **API** | Application Programming Interface |
| **ECMA** | European Computer Manufacturers Association |
| **EMMA** | Extension Multi-Modal Annotation |
| **ETSI** | European Telephony Standards Institute |
| **DSR** | Distributed Speech Recognition |
| **HTML** | Hypertext Markup Language |
| **IETF** | Internet Engineering Task Force |
| **JSAPI** | Java Speech API |
| **JSGF** | Java Speech Grammar Format |
| **JSML** | Java Speech Markup Language |
| **MRCP** | Media Resource Control Protocol |
| **NLSML** | Natural Language Semantics Markup Language |
| **SALT** | Speech Application Language Tags |
| **PML** | Phone Markup Language |
| **SAPI** | Microsoft Speech Application Program Interface |
| **SISR** | Semantic Interpretation for Speech Recognition |
| **SMIL** | Synchronized Multimedia Integration Language |
| **SRGS** | Speech Recognition Grammar Specification |
| **SSML** | Speech Synthesis Markup Language |
| **SVG** | Scalable Vector Graphics |
| **TTS** | Text-to-speech |
| **W3C** | World Wide Web Consortium |
| **VoiceXML** | Voice Extensible Markup Language |
| **X+V** | XHTML+Voice |
| **xHMI** | Extensible Human-Machine Interface |
| **XHTML** | Extensible Hypertext Markup Language |
| **XML** | Extensible Markup Language |

## 4. Szabványok és nyílt specifikációk

A beszéd alapú alkalmazások fejlesztését lehetővé tevő különböző szabványokat és nyílt specifikációkat két csoportba sorolhatjuk: az alkalmazás *leírására* hasz-

nálatos szabványok (1. ábra: 1-4) illetve az így elkészült szoftver komponensek közötti *kommunikációt* elősegítő specifikációk (5). Az alkalmazás leírásához használt nyelvek további csoportjai – tükrözve a beszéd alapú alkalmazás általános felépítését – a dialógusvezérlés (1), a bemenő és kimenő beszéd kezelése (2 és 4), valamint a beszélő azonosítás (3) funkciókat fedik le. Az *1. ábrán* kiemelve a ma használatos, teljesen kidolgozott szabványok vagy nyílt specifikációk szerepelnek, a többi fejlesztés alatt áll, kivéve a VoiceXML 1.0 és az NLSML, melyek túlhaladott szabványok.

*1. ábra Beszédtechnológiai specifikációk*
*1.– dialógusvezérlés; 2.– beszéd bemenet;*
*3.– beszélő azonosítás;*
*4.– beszéd kimenet; 5.– kommunikáció*




### 4.1. Dialógusvezérlés

A dialógusvezérlés felelős a teljes beszédfolyamat vezérléséért, a felhasználóval való kommunikációért. A dialógusvezérlés dönti el, hogy a rendszer mikor mit mondjon, illetve mikor figyelje a felhasználó szóbeli utasításait, és milyen válaszokra számítson. Ő ad utasításokat a bemenő és kimenő beszédért valamint a beszélő azonosításáért felelős komponenseknek.

A dialógusvezérlőknek többféle megközelítése létezik, de a napjainkban használt szabványosnak tekinthető megoldások a webes paradigmát követik. Azaz a webszerver jóldefiniált jelölőnyelven írt lapokat küld a böngészőnek, ha az kéri, amiket aztán a böngésző értelmez és végrehajt. A legfontosabb dialógusvezérlő jelölőnyelvek: VoiceXML, SALT, X+V, xHMI.

#### VoiceXML

A legrégebbi és a legtöbbet hivatkozott szabványos dialógus leíró formanyelv a VoiceXML (rövidebben VXML), aminek az 1.0-ás változatát még 2000-ben a VoiceXML Forum definiálta. Ebből a változatból indult a W3C Hangböngésző Munkacsoportja és készítette el a mára ajánlássá vált VoiceXML 2.0-t (2004. március). A cikk írásának idején a VoiceXML 2.1 „felhívás utolsó munkatervre" fázisban van, a 3.0-ás változatnak pedig az előkészítése folyik.

Egy VoiceXML alkalmazás általában több dokumentum együttese, ezek Web-szerveren tárolódnak, vagy szerver oldali szkriptek generálják őket. A VoiceXML böngésző dokumentumokat tölt le, értelmezi őket, majd inputot kér a felhasználótól és figyeli a választ. Azt az időtartamot, míg a felhasználó kapcsolatban van a VoiceXML böngészővel, ügymenetnek (session) nevezzük. Egy ügymenet során a hangböngésző általában több VoiceXML dokumentumot futtat. Egyidőben két VoiceXML dokumentum lehet aktív, az egyik a gyökér dokumentum (root document), mely az alkalmazásban mindig aktív, a másik a gyermekdokumentum, ami az alkalmazás egy részletét tartalmazza. Az aktív gyermekdokumentum az alkalmazás működése során mindig cserélődik.

Két elsődleges vezérlő van a VoiceXML-ben: a menü (menu) és az űrlap (form). A menü általában egy prompt lejátszást és a felhasználó szóbeli utasításának figyelését jelenti. Amikor felhasználó normál beszéd segítségével kiválasztja, hogy merre akar továbbmenni az alkalmazásban, akkor arról dönt, hogy melyik dokumentum töltődjön le és vegye át a gyermekdokumentum szerepét. Az űrlap mezőket (field) tartalmaz, melyek szóbeli közléseink alapján értéket kapnak. A mezők „kitöltését" hangos üzenetek (block) megszólaltatásával segíthetjük, és mezők kitöltöttségét is tudjuk ellenőrizni (filled). Az Űrlap Értelmező Algoritmus (Form Interpretation Algorithm, FIA) felelős a soron következő mező kiválasztásáért, a mezők kitöltését pedig nyelvtanok (grammar) felügyelik. A kitöltési algoritmus normális működését események (event) és az azokat lekezelő programrészek (event handler) futásai szakítják meg időlegesen.

Az *1. példa* egy prompt lejátszással kezdődik (4-6. sor), majd a felhasználó szóbeli választásának megfelelően (7-12. sor), az adott űrlapra lépve (14-16. vagy 17-19. sor), az alkalmazás prompt lejátszással (15. vagy 18. sor) nyugtázza döntésünket:

*1. Példa*
*Egy VoiceXML menü*

```
1  <?xml version = "1.0"?>
2  <vxml version = "2.0">
3  <menu id="travel">
4  <prompt>
5    Do you want to travel by rail, or boat?
6  </prompt>
7  <choice next="#train">
8    rail
9  </choice>
10 <choice next="#boat">
11   boat
12 </choice>
13 </menu>
14 <form id="train">
15  <block> You have selected to travel by rail.</block>
16 </form>
17 <form id="boat">
18  <block> You have selected to travel by boat.</block>
19 </form>
20 </vxml>
```

A *2. példában* az induló prompt lejátszás (5-8. sor) az űrlapon szereplő egyetlen mező (4-19. sor) kitöltésére szólít fel, amit az adott nyelvtannak (9-18. sor) megfelelően (értéke csak „march", „april", vagy „may" lehet) kell elvégeznünk. A sikeres kitöltés nyugtázását (20-22. sor) a *monthofyear* változó használata jelentősen leegyszerűsíti.

```
1  <?xml version = "1.0"?>
2  <vxml version = "2.0">
3  <form id="checkmonth">
4   <field name="monthofyear">
5    <prompt>
6     Please say the name of any month
7     from march to may.
8    </prompt>
9    <grammar type="application/srgs+xml"
10      root="monthofyear">
11    <rule id="monthofyear" scope="public">
12     <one-of>
13      <item>march <tag>march</tag></item>"
14      <item>april <tag>april</tag></item>"
15      <item>may <tag>may</tag></item>"
16     </one-of>
17    </rule>
18   </grammar>
19  </field>
20   <block>
21    You have chosen <values expr="monthofyear" />
22   </block>
23  </form>
24  </vxml>
```

*2. Példa*
*Egy VoiceXML űrlap*

A VoiceXML támogatja továbbá aldialógusok (subdialog) használatát gyakran ismétlődő részek kényelmes felhasználására, változók létrehozását, melyekkel például az aldialógusokat paraméterezhetjük, és az ECMAScript-et, mellyel procedurális feldolgozást végezhetünk.

#### SALT

A Speech Application Language Tags (SALT), amit a SALT Forum 2001-ben tett közzé, multimodális és telefonos alkalmazások fejlesztését is támogatja.

A SALT nyílt specifikáció néhány XML jelölő együttese, melyeket olyan gazdanyelvekbe ágyazhatunk, mint az XHTML, SVG, SMIL.

A legfontosabb jelölők a következők:

*<prompt>* előre felvett vagy szintetizált beszéd lejátszásáért felelős,
*<listen>* a felhasználó szóbeli utasításait figyeli,
*<grammar>* a felhasználó lehetséges közléseiben szereplő szavakat, kifejezéseket írja le,
*<dtmf>* a telefonos alkalmazások számára nyomógombos bevitelt ír elő,
*<record>* hangfelvételt tesz lehetővé,
*<bind>* a felhasználótól származó, felismert közléseket integrálja az üzleti logikával.

A SALT nem rendelkezik vezérlésátadó funkciókkal, azokról a gazdanyelvnek kell gondoskodnia. A *3. példa* egy üdvözlő prompt lejátszással kezdődik (6-9. sor), majd ha az befejeződött (*oncomplete* jellemző), újabb prompt lejátszás (10-12. sor) és a felhasználó figyelése (13-16. sor) következik. A jelszó megadása után a *<bind>* elem hatására az *iptPIN* beviteli mező kitöltésre kerül (15. sor).

```
1  <html xmlns:salt="http://www.saltforum.org/02/SALT">
2    <body onload="sayWelcome.Start()">
3     <form id="PIN" action="checkPIN.html">
4           <input id="iptPIN" type="text" />
5     </form>
6     <salt:prompt id="sayWelcome" oncomplete=
7     "askPIN.Start(); recoPIN.Start()">
8        Welcome to my speech recognition application.
9     </salt:prompt>
10    <salt:prompt id="askPIN">
11       Please say your password.
12    </salt:prompt>
13    <salt:listen id="recoPIN" onreco="PIN.submit()">
14          <salt:grammar src="PINdigits.grxml" />
15          <salt:bind targetElement="iptPIN" />
16    </salt:listen>
17   </body>
18 </html>
```

*3. Példa*

#### X+V

Az XHTML+Voice (X+V) az IBM és az Opera Software által kifejlesztett jelölő nyelv, a VoiceXML mellett az XHTML grafikus képességét használja multimodális alkalmazások fejlesztésére. A SALT-hoz hasonlóan ez a specifikáció is „hangos" jelölőket ágyaz a meglévő XHTML kódba, de nem vezet be újakat, hanem a VoiceXML 2.0 szabványban szereplőket használja. A *<sync>* jelölő segítségével köthetjük a felismert beszédet XHTML változókhoz. Az X+V alkalmazás végrehajtását a VoiceXML űrlapvezérlő (FIA) algoritmusa is szabályozhatja, de a gazdanyelv is gondoskodhat a vezérlésről.

Az X+V és a SALT is nyílt specifikáció és nem hivatalos szabvány, de valószínű, hogy a nyelv néhány eleme bekerül a W3C jövőbeni szabványaiba.

#### xHMI

Az Extensible Human-Machine Interface (xHMI) a Nuance (régebben Scansoft) által az utóbbi időben meghirdetett nyílt specifikáció, ami kompatibilis a VoiceXML és SALT formanyelvekkel, de a dialógus magasabb szintű vezérlését definiálja. Az xHMI lehetővé teszi a dialógusok közös, nyílt formában történő leírását, mely független a későbbi felhasználás módjától és az alkalmazott technológiától.

### 4.2. Beszéd bemenet

A beszéd bemenet azokat a funkciókat jelenti, amelyek lehetővé teszik, hogy a felhasználó beszéljen a rendszerhez, a rendszer megértse ezeket a közléseket és megfelelően reagáljon rájuk. A beszéd elemzése a beszédfelismerő feladata. Maga a beszédfelismerés nem standardizált, de szinte minden kereskedelmi beszédfelismerő nyelvtanon alapul, vagy legalábbis a felismerendő egységek formális definícióján.

A W3C Hangböngésző Munkacsoportja a Speech Recognition Grammar Specification (SRGS) jelölőnyelvet definiálta nyelvtanok létrehozására.

#### SRGS

Az SRGS 2004 óta W3C-ajánlás, nincs konkrét terv a következő verziójára, de ez változhat, ha a piaci szereplők újabb funkciók megvalósításának igényével lépnek fel. Az SRGS két változatban érhető el: XML és ABNF (Augmented Backus-Naur Format). Az ABNF tömörebb, az ember számára jobban olvasható, az XML alapú pedig a gép számára könnyebben feldolgozható. Mivel a nyelvtan definíciója a beszédalapú alkalmazások fejlesztésének legnehezebb része, a szabvány létrejöttének rendkívül nagy jelentősége van az egyéni megoldások használatával szemben. A 2. példa 9-18. sorában egy egyszerű, XML formájú inline („helyben kifejtett") nyelvtanra láthatunk példát.

### SISR

Az SRGS kiegészítése a Semantic Interpretation for Speech Recognition (SISR) a W3C új specifikációja. A SISR úgy terjeszti ki az SRGS-t, hogy meghatározhatjuk milyen értékkel térjen vissza a nyelvtan, amikor egy felhasználói közlést felismer. Például bizonyos szituációban az „igen”, „jó”, „oké”, „ja”, „aha” közlések felismeréséhez egységesen azok jelentését az „igen” értéket tudjuk rendelni. A 2. példa 13-15. sorában a szemantikus információ jelölésére használatos *<tag>* elemre láthatunk egy példát.

A SISR „előzetes javaslatterv” állapotban van, a technikai részletek kidolgozottak, de még végső felülvizsgálatra és implementációkra van szükség az ajánlássá válásához.

## 4.3. Beszéd kimenet

A beszéd kimenet a rendszer által kimondott beszédre vonatkozik. A beszéd kimenet alapulhat szöveg-beszéd átalakítón (Text-to-Speech, TTS) vagy előre felvett beszéd lejátszásán.

### SSML

A szöveg-beszéd átalakító bemenete lehet egyszerű szöveg, de gyakran kívánatos jelöltté tenni a szöveget, hogy a beszéd nyelvét, sebességét, a hangsúlyt, a hangerőt, a hangmagasságot, a beszélőt és egyéb tényezőket szabályozhassuk a generált beszédben. Az SSML (Speech Synthesis Markup Languages) biztosítja ezt a lehetőséget. Az SSML egy W3C-ajánlás, amit a W3C Hangböngésző Munkacsoportja fejlesztett ki. Az SSML támogatása követelmény a VoiceXML és a SALT platform számára is.

### Kiejtési szótár

A kiejtési szótár (pronunciation lexicon) létrehozása a W3C újabb kezdeményezése, melynek célja, hogy szabványosítsák a szokatlan szavak kiejtését, mind a beszédfelismerő, mind a TTS rendszerek számára. A munka „utolsó felhívás munkatervre” fázisba lépett 2006 januárjában.

## 4.4. Beszélő azonosítás

A beszélő azonosítás azokat a technológiákat jelenti, amelyek eldöntik, ki a beszélő. Habár jelenleg kimondottan beszélő személy azonosítására nincs szabvány, a biometria néhány szabványa segítségünkre lehet. A BioAPI általános programfejlesztési felület biometriai alkalmazások fejlesztésére ANSI és ISO szabvány.

A CBEFF (Common Biometric Exchange File Format) biometriai adatok leírására szolgáló szabványos adatstruktúra, az XCBF pedig ennek XML alapú verziója. A VoiceXML 3.0 több más újítás mellett a beszélő azonosítás beépítését is ígéri.

*1. Táblázat*
*Beszédtechnológiai specifikációk ([1] alapján)*

| Név | Technológia/cél | Felelős szervezet | Állapot | Alternatívák |
|---|---|---|---|---|
| **Dialógus szervezés** | | | | |
| VoiceXML 1.0 | Dialógus szervezés | VoiceXML Forum | 1999-ben hozták nyilvánosságra, mára a VoiceXML 2.0 vette át a helyét | Egyéni megoldások |
| VoiceXML 2.0 | Dialógus szervezés | W3C VBWG | W3C-ajánlás, 2004 | Egyéni megoldások, SALT |
| VoiceXML 2.1 | Dialógus szervezés | W3C VBWG | 2004 óta felhívás utolsó munkatervre fázisban van | Egyéni megoldások, SALT |
| VoiceXML 3.0 | Dialógus szervezés | W3C VBWG | Követelmények gyűjtése | - |
| SALT | Dialógus szervezés | SALT Forum | 2002-ben hozták nyilvánosságra, azóta nyílt specifikáció | Egyéni megoldások, VoiceXML |
| | Mutimodális interakció | | | Egyéni megoldások, X+V |
| X+V | Mutimodális interakció | Az IBM, az Opera Software és a Motorola összefogása | 2001 óta nyílt specifikáció | Egyéni megoldások, SALT |
| xHMI | Dialógus szervezés és multimodális interakció | Nuance és partnerei | Bejelentették, de még nem hozták nyilvánosságra | Egyéni megoldások |
| **Bemenő beszéd** | | | | |
| SRGS | Nyelvtanok definiálása beszédfelismerésre | W3C VBWG | 2004 óta W3C-ajánlás | JSGF, SAPI, Egyéni megoldások |
| SISR | Szemantikus értékek beszédfelismerőhöz | W3C VBWG | Előzetes javaslatterv, 2006 | Egyéni megoldások, JSGF jelölők, SAPI szemantikus jelölők |
| Pronunciation Lexicon | Kiejtés reprezentálása | W3C VBWG | Utolsó felhívás munkatervre, 2006 | Egyéni megoldások |
| **Kimenő beszéd** | | | | |
| SSML | Szöveg kiejtési módjának leírása | W3C VBWG | 2004 óta W3C-ajánlás | JSML, SABLE, Egyéni megoldások |
| Pronunciation Lexicon | Kiejtés reprezentálása | W3C VBWG | Utolsó felhívás munkatervre, 2006 | Egyéni megoldások |
| **Kommunikáció** | | | | |
| EMMA | A felhasználói input reprezentálásának formátuma | W3C MIWG | Munkaterv 2004 óta | NLSML |
| NLSML | A felhasználói input reprezentálásának formátuma | W3C VBWG | Munkaterv 2000 óta | EMMA |
| MRCP v2 | Szétosztott beszédfunkciók | IETF SpeechSc munkacsoportja | 2005 decemberében publikálták az utolsó munkatervet | - |
| DSR-Aurora | Elosztott beszédfelismerési feladatok | ETSI Aurora | Az 1.1.3-as verziója 2003-ban jelent meg | - |

### 4.5. Kommunikáció

A beszéd alapú alkalmazás legfontosabb részeinek leírásán túl, néhány további szabvány az elkészült komponensek kommunikációját biztosítja. A szabványosított kommunikációs protokollok abban az esetben különösen fontosak, ha a különböző rendszerkomponenseket a hálózat erőforrásain szétosztjuk, vagy ha az egyes rendszerkomponensek különböző szállítótól érkeznek.

#### EMMA

A W3C Multimodális Interakció Munkacsoportja jelenleg is fejleszti az Extensible Multi-Modal Annotation (EMMA) specifikációt, amely a felhasználótól érkező input szabványos leírása. A bemenet forrása tetszőleges lehet: beszéd, kézírás, látás stb. A beszéd alapú alkalmazások esetében a beszédfelismerők így szabványos szövegekkel térhetnek vissza, ami nagyban segíti ezen komponensek integrációját. Az EMMA hamarosan „utolsó felhívás munkatervre" fázisba kerül.

#### MRCP

A Media Resource Communication Protocol (MRCP) az IETF fejlesztése. Célja, hogy leválassza a beszédfunkciókat (beszédfelismerés, beszédszintézis és beszélő azonosítás) a saját platformjukról úgy, hogy közben szabványos kommunikációs protokollt ír elő az együttműködésükre. Az MRCP v2 a Natural Language Semantics Markup Language (NLSML) szabványt használja – az EMMA elődjét – a felhasználói input reprezentálására.

#### DSR – Aurora

Az ETSI által definiált Aurora nevű szabvány a beszéd-felismerési funkciókat szétosztja helyi és távoli folyamatokra. Sok esetben előnyösebb, ha lokálisan is végzünk némi beszéd-felismerési feladatot és csak a köztes eredményt továbbítjuk a szerver felé. Például csökkenthetjük a beszédfelismerés hibáját, mivel kevesebb az esély, hogy zaj kerül a beszédjelbe, illetve kisebb sávszélességgel is megelégedhetünk, mivel nem a teljes beszédjel kerül át a szerverre. Ezt a technológiát főképp mobil alkalmazásokban használják.

## 5. Összefoglalás

A beszéd területén használt szabványos jelölőnyelvek lefedik a dialógusvezérlés, a beszéd be- és kimenet, valamint a komponensek közötti kommunikáció területét. Alkalmazásuktól eszközeink jobb együttműködését, megbízhatóbb technológiai hátteret, gyorsabb, hatékonyabb fejlesztési folyamatot várunk. Természetesen önmagában a szabványok használata nem biztosítja a jó beszéd alapú alkalmazás létrehozását. De ha alkalmazásuk megfelelő fejlesztési tapasztalattal párosul, és figyelembe vesszük az adott felhasználási terület egyéni adottságait, kivívhatjuk a felhasználók elégedettségét.

A jelölőnyelvek dinamikus fejlődése várhatóan tovább folyik a következő években, a W3C két említett munkacsoportjának a működését 2007-ig újra meghosszabbították. Az egyes nyelvek sikerét sok tényező befolyásolja, de az, hogy mennyire találnak támogatásra az egyes fejlesztőkörnyezetekben, illetve, hogy mennyire nyitottak a nemzetköziesítésre, mindenképp a legmeghatározóbbak. A magyar kutatókra, fejlesztőkre vár, hogy ezen, a természetéből adódóan rendkívül nyelvfüggő területen, a szabványok „honosítását" elvégezzék.

A folyamat elkezdődött. 2002-2003-ban a BME Távközlési és Médiainformatika Tanszékén elkészült az első magyar nyelvű VoiceXML böngésző (a felhasznált komponensek részletezése [6] és [7]-ben található). Az MTA SZTAKI Elosztott Rendszerek Osztálya pedig részt vett az EU által támogatott PublicVoiceXML-projektben, melynek célja az első ingyenes és nyílt forráskódú hangböngésző megvalósítása volt [2].

### Irodalom


[1] Dahl, Deborah A.:
Guide to Speech Standards.
Speech Technology Magazine, March/April 2005.

[2] Déri András, Fülöp Csaba, Micsik András:
Telefonos szolgáltatások VoiceXML alapon,
NetworkShop 2003 konferencia,
2003. április 14-17., Pécs

[3] Larson, James A.:
VoiceXML:
Introduction to developing speech applications.
Prentice Hall 2003.

[4] Larson, James A.:
State of Speech Standards.
Speech Technology Magazine, July/August 2003.

[5] Kovács, L., Vásárhelyi, Nóra:
Webhez kapcsolódó szabványosítás Magyarországon.
http://nws.iif.hu/ncd2004/docs/ehu/072.pdf

[6] Olaszy, G., Németh G., Olaszi, P., Kiss, G., Gordos, G.:
„PROFIVOX – A Hungarian Professional TTS System
for Telecommunications Applications",
International Journal of Speech Technology,
Vol. 3, Numbers 3/4, December 2000, pp.201–216.

[7] Szarvas, M., Fegyó, T., Mihajlik, P., Tatai, P.:
Automatic Recognition of Hungarian: Theory & Practice,
Int. Journal of Speech Technology,
Vol. 3, Numbers 3/4, December 2000, pp.237–251.

[8] SALT Forum,
http://www.saltforum.org/

[9] VoiceXML Forum,
http://www.voicexml.org/