

Hangportál-szolgáltatások minőségmérése

Balog András¹, Mihajlik Péter², Fegyó Tibor¹, Tatai Péter¹

¹ Aitia International Informatikai Zrt.
1039 Budapest, Czetz János u. 48-50.

² Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és
Médiainformatikai Tanszék, Távközlési és Beszédjel-feldolgozási Laboratórium
1117 Budapest, Magyar tudósok krt. 2.

Kivonat: A cikkben beszámolunk egy hangportál tesztelő rendszer fejlesztéseiről. A hangportál szolgáltatás-minőség már a bemutatott kísérleti rendszerrel is többértűen vizsgálható a tipikus hívásközpontok, interaktív beszédválaszú rendszerek esetében. Szemléltetjük az automatizált hangportál-mérési alapelveket, a teljesség igénye nélkül vázoljuk a lehetséges alkalmazásokat, valamint bemutatjuk a mérést vezérlő script nyelvet és a mérés során generált napló állományokat. A rendszer jelenleg az analóg és a VoIP illesztési felületeken keresztül képes több csatornán párhuzamosan mérni, de az ISDN BRI/PRI vonali illesztési felület is hamarosan elérhető lesz. További fejlesztések keretein belül az intelligens beszéd-szintetizálásra és gépi beszéd felismerésre épülő tesztfunkciók is beépítésre kerülnek.

Bevezető

Nemzetközi és hazai viszonylatban is rohamosan terjednek a hívásközpontok, interaktív beszédválaszú rendszerek, hangportálok. Azonban e szolgáltatások minőségbiztosítása nem feltétlenül tart lépést a gyors elterjedéssel. A minőségbiztosítás gyakran megreked a hangfelvételek jel-zaj viszonyának biztosításánál, holott a felhasználó számára egyes egyéb jellemzők sokkal kritikusabbak lehetnek, mint pl. az ügyintézőhöz jutás átlagos ideje, szórása, a rendszer válaszában késleltetése, vonalmegszakadások stb. A valódi minőségbiztosítás hiányának eredményeképpen a felhasználók általános elégedettsége szintje ezekkel a szolgáltatásokkal kapcsolatban nem minden esetben mondható kedvezőnek.

Egy több éves projekt keretében¹ azt a célt tűztük ki, hogy olyan – a felhasználó szempontjából releváns – szolgáltatásminőségi paramétereket határozzunk és mérjünk meg, melyek egyrészt magas szinten automatizáltan vizsgálhatóak, másrészt jó tájékoztatást adnak a mért objektum főbb szolgáltatásminőségi paramétereiről – legyen szó akár egy hagyományos hívásközpontról, akár egy automatikus beszédfelismerésre épülő hangportálról.

A következőkben röviden vázoljuk a megoldandó problémát, illetve bemutatjuk a kezdeti fázisban lévő, de már alpmérésekre használható mérőrendszerünket. A továbbiakban az interaktív beszédválaszú rendszereket (IVR – Interactive Voice Response), hívásközpontokat (Call-Center), és az akár beszéd szintetizátort és/vagy beszédfelismerő motort tartalmazó hangportálokat egységesen kezeljük, és általában egyszerűen csak hangportálok néven hivatkozunk rájuk.

Miért érdemes géppel tesztelni egy hangportált?

A hangportálok tesztelésének, minőségellenőrzésnek legkézenfekvőbb módja, ha ugyanúgy, mint valós működés közben, felhívjuk, és kommunikálunk velük. Eközben a rendszerben tapasztalt hibákat, zavaró jelenségeket, szubjektív benyomásokat feljegyezzük. Mivel a hangportálok emberek számára készülnek, elengedhetetlen az ilyesfajta személyes tesztelés.

Vannak viszont a hangportálnak olyan jellemzői, esetleges hibalehetőségei is, amelyeket egyszerűen, objektíven, esetleg számszerűen lehet minősíteni, így alkalmasak gépi vizsgálatra. Sőt, az emberi megfigyelőképesség korlátai miatt nem csak hogy lehetséges, de sokkal hatékonyabb is lehet a tesztelés, mivel a gép folyamatosan oda tud figyelni, gyorsan tudja pillanatnyi tapasztalatait lejegyezni (akár olyan részletesen, amiből bonyolult utófeldolgozás során több szempontból is vonhatunk le következtetéseket), és pontosan képes mérni – például időtartamot.

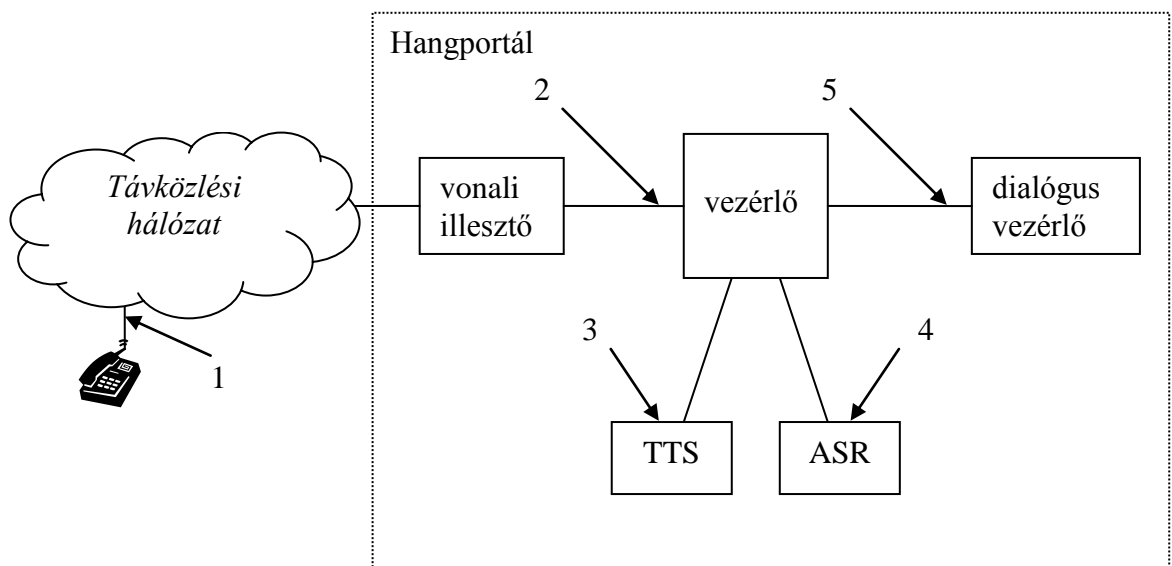
Egy másik szempont, ami miatt érdemes lehet a hangportál tesztelésének automatizálásával foglalkozni, a többcsatornás működés értékelése. Itt arról van szó, hogy azt az esetet szeretnénk vizsgálni, amikor egyszerre több felhasználó hívja a portált. Élő felhasználókkal ennek a megszervezése nehézkes. Gondot okozhat többek között a megfelelő számú embert előkeríteni, kiképezni őket a tesztelés mikéntjére, sőt esetenként a tesztelő tevékenységük időbeli összehangolására is szükség lehet, például ha azt szeretnénk, ha egy időben minden felhasználó a beszédfelismerőt használja. Megfelelő mértékű összehangolás valószínűleg nem is érhető el, a kísérlet pedig biztosan megismételhetetlen, ez nehezíti a hibák felderítését. Ezzel szemben, ha a tesztelő alkalmazásunkban megvalósított nagy számú virtuális felhasználót megfelelő programmal összehangolt működésre utasítjuk, a másodperc tört részére pontosan betartható a kívánt ütemterv, és legközelebb, ha szükség van rá, pontosan ugyanúgy, ugyanazokkal a hullámformákkal ismételtető meg a kísérlet.

¹ A Jedlik Ányos Program támogatásával létrejött „Új vizsgálati és mérési módszerek kidolgozása korszerű távközlési szolgáltatások minőségének biztosítására” című NKFP2-00015/2005 projekt keretein belül.

Hol, mit lehet mérni, tesztelni?

A tesztelés legtermészetesebb módja, ha távközlési hálózatba illesztve, fekete doboznak tekintve vizsgáljuk a hangportált. A megközelítés előnye, hogy a mérőrendszer így a felhasználóval azonos módon „látja” a portált. Hátrány, hogy nem közvetlenül, hanem a hálózattal együtt mérjük a hangportált, ez azonban kiküszöbölhető a mérésre dedikált minimális hálózat (pl. alközpont) alkalmazásával.

A tesztelő be- és kimenetei nem csak kívülről, a teljes hangportál be- és kimeneteihez csatlakoztathatjuk. A hangportálok többnyire moduláris felépítésűek, az egyes modulok kommunikálnak egymással. A modulok közötti kommunikációs csatornákat átkonfigurálva beépíthetünk tesztelő alkalmazásokat a hangportál szoftver különböző helyeire – ha a portál programozói ezt lehetővé tették. A hangportál tipikus blokkdiagramját az 1. ábra mutatja. Számozott nyilak jelölik azokat a helyeket, ahova tesztelő alkalmazás csatlakoztatásacélszerű lehet.



1. ábra: A hangportál és környezetének blokkvázlata

- Az 1-es nyíl a távközlési hálózaton keresztül analóg telefon, ISDN vagy VoIP vonalra csatlakozik, erről már volt szó.

- A 2-es nyíl a hangportál szoftverben a telefon illesztő hardver és a vezérlő szoftver közötti csatlakozást jelöli. Nagyjából egyenértékű az 1-es esettel, a lényeges különbség, hogy itt a hangjelek már digitálisak. Tesztelhetünk innen, ha nem akarunk foglalkozni a telefonvonal torzításával és zajával, vagy ha tisztán szoftveres tesztelést szeretnénk megvalósítani, telefonhálózat használata nélkül.

- A 3-as és 4-es helyeken tesztelhetjük elkülönítve a beszédszintetizátort (TTS: Text To Speech) vagy a beszédfelismerőt (ASR: Automatic Speech Recognition). Ez nem annyira a hangportál tesztelés témája, mindkét egység tudományának megvannak a saját tesztelési, hibakeresési módszerei [2], [3]. Tekintsük most ezeket a szoftveregységeket gyári fekete dobozoknak, amelyek ismert minőséggel (pl. felismerési hibarány) működnek.

- Az 5-ös helyen a dialógusvezérlőt tesztelhetjük. Mivel ez adja a szoftver hangportál mibenlétének lényegét, érdekes lehet ezt önmagában is tesztelni, ha a portált akarjuk minősíteni. A dialógusvezérlő felületén be- és kilépő információk egyrészt a dialógus menetével kapcsolatos absztrakt adatok, másrészt a kimenő szöveg előállítására vonatkozó adatok (szöveg a TTS számára vagy hivatkozás lejátszandó hangmintára), vagy bejövő beszéd ASR által értelmezett alakja, illetve bemenet lehet még a telefon billentyűzetének

megnyomása (DTMF). Mindezek formátuma lehet szabványos (Voice XML) vagy a hangportál gyártó saját adatszerkezete. Ezen a helyen azt vizsgálhatjuk, hogy helyes-e a dialógus leírása, minden értelmes vagy értelmetlen bemenetre értelmesen, helyesen reagál-e. VXML dialógus leírás esetén használhatjuk a már létező VXML debugger(ek)e)t, saját formátumhoz saját debuggert/tesztelőt kell írni. A továbbiakban a 1-es hozzáférési felületen történő mérésekkel fogunk foglalkozni.

A tesztelés szempontjai

Egy hangportál működésének minősítésére több szempontból is szükség lehet. A kézi, élő felhasználós tesztelés eredménye a felhasználók szubjektív benyomásainak és konkrét észrevételeinek összessége. Véleményt mondhatnak a felhasználók többek között

- a dialógusrendszer ergonómiájáról, áttekinthetőségéről
- a beszédfelismerő találati arányáról, a félreismerések zavaró jellegéről,
- a beszédszintetizátor érthetőségéről és élvezhetőségéről,

Bár ezek fontos szempontok, talán a legfontosabbak, mivel a felhasználók számára készül a hangportál, de igen korlátozott mértékben gépesíthetők. Lehetőség van viszont más jellegű, objektívabb paraméterek gépi minősítésére, például még az éles tesztelés előtt.

Számítógép segítségével tesztelhető, vizsgálható

- a dialóguskezelő rendszer helyes működése: Valóban a megadott dialógusleírás szerint működik-e a portál?
- üzembiztonság, terhelhetőség: A hangportált futtató számítógép növekvő terhelése esetén (ami a csatlakozó felhasználók számával nagyjából arányosan nő) romlik-e, és ha igen, milyen módon a felhasználókkal való interakció minősége? Milyen mértékben nő a válaszidő? Akadozik-e a hanglejátszás? Intelligens módon, hibaüzenettel adja a felhasználó tudtára a túlzott terhelés tényét? Esetleg lefagy-e egy, több vagy az összes csatorna?
- zajérzékenység: A felhasználó környezetében vagy a telefonhálózatban fellépő zaj hatására romlik-e a beszéd- és a DTMF felismerő hibaránya? Ez valójában tisztán az említett programmodulok tesztelését jelenti, nem kell ezért az egész hangportált futtatni.
- bolondbiztosság: Hogyan kezeli az értelmetlen bemeneteket? (értelmezhetetlen bemondás, értelmezhető, de nem helyénvaló bemondás, illetve DTMF) Figyelmeztet-e, hogy hibás bemenetet kapott? Úgy tesz-e, mintha valami más, értelmes utasítást kapott volna? (Ez főleg a beszédfelismerő minősítése.) Esetleg lefagy?
- ergonómia: Ez inkább élő felhasználóval tesztelhető, de lehetnek mérhető mennyiségei is. Például a válaszidők, szünetek hossza.

A tesztelés módszerei

A dialógus helyességének tesztelését, hibakeresését célszerű a dialóguskezelő felületén vagy csak a dialógusleírás alapján végezni. A szabványos, VXML leírást lehet VXML debuggerrel vizsgálni.

A VXML eredetileg csak dialógus leíró nyelv volt, amit aztán kibővítettek egyéb, hangportálok számára hasznos adatok leírásával, például nyelvtan, beszédfelismerő, beszédszintetizátor vezérlés leírással. A szabványt a W3C Voice Browser Working Group hozta létre és fejleszti [4].

Ha a hangportál egészét kell tesztelni, ezt telefonvonalon keresztül analóg hangjellel, vagy a telefon illesztő után digitális adatokkal tehetjük meg.

A hangportál bemenetére jutó hangot többféle módon állíthatjuk elő. A DTMF előállítását egyértelmű. A hangjelet adhatjuk előre rögzített beszéd lejátszásával vagy beszéd szintetizátorral. Az utóbbi megoldás processzor igényesebb, viszont nem igényel élő beszélős felvételeket (ami hosszadalmas lehet) és rugalmasabb. A beszéd, illetve DTMF lejátszásokat időzíteni kell. Egyrészt a hangportáltól érkező válaszok szerint, másrészt több csatornán történő tesztelésnél fontos lehet a csatornák közötti szinkronizálás is. Ez abban az esetben érdekes, ha több felhasználó használja a beszéd felismerőt vagy a dialógusvezérlőt vagy bármely más komponenst.

A hangportál kimenete főként beszéd és jelző hangok (amik például azt jelzik, mikor kezdjen beszélni a felhasználó), esetleg háttér- vagy várakozási időt kitöltő zene, szignál. Ezek közül az első kettő értelmezése érdekesebb. A tesztelőbe bejövő hangjel spektrumának folyamatos kiszámítása alapja lehet a jelzőhang, beszéd, illetve csönd detekciónak. Érdekes mérni a folyamatos beszéd és a beszéd (vagy hangok) közötti szünetek hosszát. A szünet hosszából például leterheltségre lehet következtetni, sűrű beszéd-szünet váltások sorozata pedig akadozó hangelőállítást jelent, amit fontos észrevenni. Egyszerűbb esetekben elég lehet a beszéd, vagyis a mondatok hosszát mérni, ha kevésféle mondat lehetséges, esetleg csak a hosszak alapján is el lehet dönteni, melyiket mondta a hangportál. Sok lehetséges mondat, és főleg dinamikus tartalom esetén szükséges beszéd felismerő használata a tesztelőben. A felismert beszéd segítségével a dialógus menetét tudja követni a tesztelő.

A tesztelő futása, vagyis a hangportállal való telefonbeszélgetés alatt érdemes minden észlelt eseményt, mért mennyiséget, felismert hangot, illetve beszédet, és a tesztelő által végrehajtott tevékenységet naplózni. Ebből – célszerűen a beszélgetés után – egy utófeldolgozó leszűri a kívánt és lehetséges minőségi jellemzőket, hibákat, és olvasható beszámolót generál belőle.

A kísérleti hangportál tesztelő motor

A kísérleti hangportál tesztelő motor egy speciális, „inverz” hangportálnak is felfogható, mivel ugyanazon blokkokból, hasonlóan épül fel (lásd 1. ábra), de a normál hangportál dialógus ellentett műveleteit végzi. Többlettudása egy általános hangportálhoz képest, hogy egyrészt képes hívást kezdeményezni több csatornán keresztül, másrészt speciális elemzésekre képes az audio bemenet alapján. Továbbá a hangportál tesztelő speciális vezérlő script-tel és naplózási képességekkel rendelkezik.

Az általunk kidolgozott hangportál tesztelő motor egy ipari PC-alapú hardveren fut. A mérést voltaképpen szoftver végzi, a távközlési hálózathoz PCI-buszos kártyán, illetve USB csatolású eszközzel csatlakozunk. A kísérleti változat többcsatornás analóg és VoIP (Voice Over IP) vonali illesztési felülettel is rendelkezik, az ISDN BRI/PRI illesztés fejlesztése folyamatban van. A mérőrendszer az ISDN-PRI hálózathoz speciális illesztőkártyával [1] csatlakoztatható, BRI szinten általános (Eicon, AVM) kártyával. Az analóg csatolást szintén speciális USB kártyával valósítottuk meg, mely P-Tel néven kerül hamarosan kereskedelmi forgalomba.

A kísérleti hangportál tesztelő motor használata során elsősorban a vezérlő script előállítását, testre szabását a feladat, ezért a következőkben röviden ismertetjük ennek a felépítését.

A hangportál tesztet vezérlő script

Ez a speciális script nyelv alacsony szintű, flexibilis szoftveres illesztő felületként is felfogható, melyet a későbbiekben célszerű magas szintű (pl. Voice XML) leírásból automatikusan előállítani.

A script egy szöveges fájl, mely több blokkra tagolódik. A legelső, konfigurációs rész nem feltétlenül kell, hogy szerepeljen. Ezen kívül minden rész egy-egy program szálnak felel meg, amelyek párhuzamosan hajtódnak majd végre, futásuk egy időpontban indul.

Létezik pontosan egy vezérlő szál, szerepe a több csatornán futó különféle vagy akár azonos tesztelő folyamatok vezérlése, ütemezése, szinkronizálása a többi szálnak küldött üzenetek által. A hangportál tesztelő futása akkor ér véget, ha a vezérlő szál elérkezik programjának végéhez.

A vezérlő szálon kívül lennie kell legalább egy csatornához tartozó szálnak. Ez a

Channel <logikai azonosító> <vonali interfész típus> <fizikai azonosító>

sorral kezdődik. A vonali interfész típus és a fizikai azonosító egy fizikai telefonvonalat határoz meg, minden, az adott szálnak található telefonos műveletet végző utasítás erre a vonalra vonatkozik. Vonali interfész típus lehetséges értékei: **USB** (azaz analóg intrerfész), **ISDNBRI**, **ISDNPRI**, **VoIP**. A fizikai azonosító értelmezése a vonali interfész típustól függően különböző. A csatornákra való hivatkozás minden esetben (pl. üzenet küldés, vétel) a logikai azonosítóval történik.

Egy szál programját leíró blokk

Minden sor egy utasítás. A sor első szava a kulcsszó, az utána következő(k) a paramétere(i), vagy ha az első szó kettősponttal kezdődik, akkor az címke, a második a kulcsszó, és utána a paramétere(i).

A szál programját a tesztelő motor a telefonos műveletek közben soronként értelmezi. Ezen kívül egyszer végigolvassa a telefonálás megkezdése előtt, ekkor értelmezi az eseményfigyelés beállító deklarációkat. A szálak címkék és eseményfigyelések szempontjából teljesen különállóak. A szálak végrehajtása egymástól teljesen függetlenül történik, kivéve a szálnak üzenő utasításokat és a száltól jövő üzenet elkapó eseményfigyelést.

Utasítások

Az utasítások voltaképpen elemi hangportál műveletek, vezérléseket jelentenek. A legfontosabbak:

Wait <időtartam (ms)>,

SetChTelNr <telefonszám vagy egyéb hívás cél azonosító>

Call,

Hangup,

Play <fájlnév>

Dial

End

Goto <címke>

IfEvent <GoTo/GoSub> <ugrás címkéje> <esemény típus> <esemény paraméterek>

EnableEvent <azonosító címke>

DisableEvent <azonosító címke>

Stb.

Ezekon kívül számos script vezérlési utasítás teszi lehetővé összetett tesztelési feladatok megoldását.

Esemény típusok

A scriptben számos esemény típust definiáltunk, melyek a hatékony kódgenerálást teszik lehetővé. Pár példa:

SpeechStart

SpeechEqu <hossz ms-ban> <tűrés ms-ban>

SilenceEqu <hossz ms-ban> <tűrés ms-ban>

Beep <frekvencia Herzben>

Timer

MsgReceived <küldő szál azonosítója/**any**> <üzenet adat/**any**>

DTMF

HangUp

A script végrehajtásának menete

Mielőtt az interpreterre kerülne a sor, a hangportálesztelő program végignézi a cfg fájlt, nyilvántartásba veszi a címkéket és az eseményfigyelés definíciókat. Ez egyrészt azzal jár, hogy egy ugróutasítás cél címkéje ugyanúgy lehet sorrendben az ugró utasítás előtt, mint utána. Másrészt az eseményfigyelés definíciók már a futás előtt feldolgozásra kerülnek, futási időben engedélyezni és tiltani lehet őket. Lehet még az ugrásuk célját is változtatni, ez elsősorban az áttekinthetőséget és a hatékonyságot növeli, mivel így kevesebb eseményfigyelést kell definiálni.

Ezután következik a csatornához tartozó telefon illesztő egységek megnyitása, majd ha az összes egység megnyitása megtörtént, egyszerre elkezdődik a szálak futása (mind a vezérlő, mind a csatornához tartozó szálaké).

Egy szálon belül az utasítások végrehajtása sorban egymás után történik. Ugrás, illetve szubrutinhívás bekövetkezhet explicit módon, vagy egy engedélyezett esemény bekövetkezéskor.

Ha egy csatorna szála elér egy end utasítást, befejezi a futását, de a többi szál tovább fut. Ha a vezérlő szál elér egy end utasítást, befejeződik a program futása attól függetlenül, hogy a többi szál végzett-e.

Naplózás

A hangportál tesztelő kimenetei a napló állományok. Elsődleges az összesített illetve csatornánkénti szöveges állomány, melyet a felhasználó akár közvetlenül elemezhet (pl. Perllal), vagy a későbbiekben az eszköz által készített elemzést is választhatja. Továbbá rendelkezésre állnak a tesztelés közben rögzített audio állományok is, melyek különösen nem várt események elemzésekor lehetnek hasznosak.

A szöveges log fájlok

A naplózás alapegysége a sor. Egy napló bejegyzés egy fájlban pontosan egy sorban van tárolva. A program ezeket a naplósorokat keletkezési idő szerint rendezve tárolja, egyszer ömlesztve minden szálból érkező naplósort, és szálanként külön válogatva is. Az ömlesztett log fájl neve: „Text_Log_All_” + létrehozásának időpontja (év, hónap, nap, óra, perc,

másodperc) aláhúzás karakterekkel elválasztva + „.txt”. A szálankénti log fájlok nevei: „Text_Log_” + szál neve + létrehozásának időpontja (év, hónap, nap, óra, perc, másodperc) aláhúzás karakterekkel elválasztva + „.txt”. A szál neve lehet „Main”, „Ctrl” és „Ch” + a csatornához tartozó USB eszköz fizikai azonosítója.

Szemléltetésül egy egyszerű naplóállomány-részlet:

```
[Jul 05 12:42:18.280] [Ch03] [LL000] Opened text log file: *Text_Log_Ch03_2006_Jul_05_12_42_18.txt
[Jul 05 12:42:19.296] [Ch03] [LL020] USB device opened successfully.
[Jul 05 12:42:19.796] [Ch03] [LL021] Opened audio log file: *AudioLog_Ch03_2006_Jul_05_12_42_19.wav
[Jul 05 12:42:19.796] [Ch03] [LL022] Program execution started.
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *2, command: *:e_kicseng ifevent goto kicseng ring
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *3, command: *:e_foglalt ifevent goto foglalt busy
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *4, command: *:e_felvettek ifevent goto felvettek answered
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *5, command: *:e_lerak ifevent goto lerak hangup
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *6, command: *:e_jelvan ifevent goto jelvan anysignal
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *8, command: *:e_sp_akettos ifevent goto sp_akettos spechequ 1...
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *9, command: *:e_sipszo_elotte ifevent goto sipszo_elotte beep 440
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *13, command: *setchtelnr 17
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *16, command: *let i 0
[Jul 05 12:42:19.796] [Ch03] [LL035] Creadted variable: *i
[Jul 05 12:42:19.796] [Ch03] [LL036] Variable *i new walue: *0
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *18, command: *ifgt i 2 cv
[Jul 05 12:42:19.796] [Ch03] [LL023] Executing line *20, command: *call
[Jul 05 12:42:20.061] [Ch03] [LL015] *0.26 s break detected.
[Jul 05 12:42:20.296] [Ch03] [LL023] Executing line *23, command: *wait 1500
[Jul 05 12:42:20.327] [Ch03] [LL016] *0.04 s speech/sound detected.
[Jul 05 12:42:20.327] [Ch03] [LL015] *0.28 s break detected.
[Jul 05 12:42:21.655] [Ch03] [LL017] Dialing completed.
```

A naplósorok felépítése a következő:

A sor első szögletes zárójele: az üzenet érkezésének időpontja: hónap nap óra:perc:másodperc.ezredmásodperc. Ez az adat néhány (1-2) tizedmásodperc pontosságú. Szögletes zárójel, amiben "Ch" és egy szám, „Ctrl”, vagy „Main” van: azt mondja meg, hogy melyik szálban történt a sorban naplózott esemény. A „Ctrl” a vezérlő szálát jelenti, a „Main” pedig a főprogramot, ami a szálak (vezérlő és csatornás) létrehozása előtt, illetve megszüntetése után fut.

A következő szögletes zárójelben az üzenet azonosítója látható. „LL” (Azt jelenti, log line.) és egy háromjegyű szám. Ezt követi az üzenet szöveges leírása. Közben *-gal kezdődő részek is lehetnek. Ezek a naplósor konkrét szituációra vonatkozó adatai. Például egy naplósor, ami azt mondja, hogy bekövetkezett egy esemény, a csillag után közli a bekövetkezett esemény azonosítóját. Az üzenet azonosító és a csillag a gépi feldolgozás egyszerűsítése végett van.

Zárójelen kívül: maga az üzenet. A megtörtént esemény rövid leírása. Minden utasítás végrehajtása is naplózásra kerül. Ez a „[LL023] Executing line” üzenet. A feldolgozás alatti programsor sorszáma 0-ról indul, minden szálban a „Control” vagy „Channel” sor utántól. A „Command:” szó után maga a programsor látható.

Hang logok

Minden csatornához létrehozunk egy-egy fájlt, ami folyamatosan minden, a csatorna vonalán hallható hangot rögzít. A szöveges log fájlban a szálak futásának megkezdése előtt olvasható, hogy mi a csatornához tartozó hang log fájl neve.

Formátuma ugyanaz, mint a lejátszható wav fájloké. (8kHz, 16bités minták)

A fájlok nevei: „AudioLog_Ch” + USB eszköz fizikai azonosítója + „_” + létrehozásának időpontja (év, hónap, nap, óra, perc, másodperc) aláhúzás karakterekkel elválasztva + „.wav”

Alkalmazások

A bemutatott hangportál teszternek már a kísérleti verziójával is számos összetett és hasznos minőségmérési feladat elvégezhető. Jól alkalmazható olyan hangportálok esetén, melyeknek a vizsgált tartalma nem függ a hívószámtól, illetve helyétől, csak a hívás tartalmától. Ilyen a legtöbb ügyfélszolgálati hívasközpont akár bankoknál, távközlési szolgáltatóknál, vagy például önkormányzatoknál. Néhány példa:

- Terheléses teszt: hány csatornát képes a rendszer lekezelni egyidejűleg, hogyan romlik a válaszidő a terhelés függvényében, stb.
- Adott menüstruktúra ellenőrzése, automatikus teljes bejárással.
- Adott telefonszámok rendszerezett automatikus felhívása, és így a különféle hívasközpontok, beszédválaszú rendszere rendelkezésre állásának hosszú távú monitorozása.
- Hívasközpontoknál az ügyintézőhöz jutás idejének mérése, monitorozása
- A jel-zaj viszony mérés alapján a hangminőség mérése.
- Stb.

Az automatikus hangportál-tesztelés korábban ismertetett előnyei (megismételhetőség, események aprólékos naplózása, stb.) mellett a költséghatékonyság is megemlítendő, hiszen egy sokcsatornás automatizált mérés előkészítése néhány munkaórával megvalósítható, valamint az esetleges hiba felderítés is felgyorsul a részletes naplódatoknak köszönhetően.

Továbbfejlesztési irányok

Az intelligens – automatikus beszéd előállító és felismerő – eszközök alkalmazása kezdetektől fogva terveink között szerepelt, integrációjuk folyamatban van. A beszéd szintetizátor használatának jelentősége abban áll, hogy az akár funkcionális, akár tájékoztatási céllal a tesztelő által „bemondott” beszédrészleteket nem kell minden egyes alkalommal felvenni, megvágni és beilleszteni, hanem pár másodperc alatt szöveges formában lehet rögzíteni. Ezáltal jelentős munka és idő spórolható meg.

Hasonlóan kényelmi funkciókat is szolgál a beszéd felismerés alkalmazása, azonban emellett a tesztfunkciók érdemi bővülését is maga után vonja. Ugyanis – hasonlóan az emberhez – az automatikus beszéd felismerés révén a gép értelmezhet is az elhangzottakat, és annak megfelelően folytatódhat a tesztelés. A gépi beszéd felismerés további funkciói – mint például annak megállapítása, hogy elhangzott-e egy adott szó/kifejezés, vagy sem – újabb és újabb tesztfunkciók hozzáadását teszik lehetővé, így közelítve az emberi operátor képességeit.

Részben függetlenül az eddigiektől nagyon fontosnak tartjuk a szabványos hang-alapú dialógus leíró nyelvek támogatását, ezért közeli terveink között szerepel egy olyan speciális Voice XML értelmező kifejlesztése, amely nem a VXML-ben leírt alkalmazást, hanem annak inverz, tesztalkalmazását valósítja meg (fordítja a korábban ismertetett hangportál-tesztelő script nyelvre). Így bármely támogatott Voice XML szabvány szerint megírt hangportál alkalmazás azonnal, szinte emberi erő ráfordítása nélkül több csatornán tesztelhetővé válik.

Összefoglalás

A cikkben beszámoltunk az Aitia International Zrt-nél a BME Távközlési és Médiainformatikai Tanszékével együttműködésben készített hangportál tesztelő rendszer fejlesztéseiről. A hangportál szolgáltatás-minőség már a bemutatott kísérleti rendszerrel is többrétűen vizsgálható a tipikus hívasközpontok, interaktív beszédválaszú rendszerek esetében. Szemléltettük az automatizált mérés fontosságát, teljesség igénye nélkül vázoltunk

néhány lehetséges alkalmazást, valamint a rendelkezésre álló keretek között bemutattuk a mérést vezérlő script nyelvet és a mérést során generált napló állományokat. A rendszer jelenleg az analóg és VoIP/sip illesztési felületeken keresztül képes több csatornán mérni, de az ISDN BRI/PRI vonali illesztési felület is hamarosan elérhető lesz. További másfél éves fejlesztés kereteiben az intelligens beszéd szintetizálásra és gépi beszédfelismerésre épülő tesztfunkciók is beépítésre kerülnek.

Irodalomjegyzék

- [1] Tatai, P., Marosi, Gy., Osváth, L.: A Flexible Approach to Mobile Telephone Traffic Mass Measurements and Analysis, Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, Budapest, May 21-23, 2001, pp. 1281-1287
- [2] Fegyó, T., Mihajlik, P., Szarvas, M., Tatai, P., Tatai, G.: Intelligent Voice Enabled Call Center for Hungarian, *Proceedings of Eurospeech 2003, 8th European Conference on Speech Communication and Technology*, pp. 1905-1908, Geneva, Switzerland, Sept 1-5, 2003.
- [3] Olaszgy Gábor, Gordos Géza, Németh Géza: A gépi beszédkeltés korszerű módszerei Alkalmazott nyelvtudomány III/1. 2003 13-28.
- [4] <http://www.vxml.org/>