

Piattaforma vocale TEMPEST

Sommario

Sommario	1
Descrizione generale	1
Strumenti per sviluppatori software	2
Architettura della piattaforma.....	2
Struttura modulare e distribuita	2
Caratteristiche dei moduli principali	3
Phone Engine	4
Speech Portal (IVR Engine e Configuratore grafico)	7
Dialog Manager.....	10
Appendice A)	13
Modalità di integrazione Voip	13
Modulo VoIP senza alcun componente aggiuntivo	13
Modulo VoIP e Gatekeeper	13
Modulo VoIP e MediaProxy	13
Considerazioni	14
Referenze	16

Descrizione generale

TEMPEST è una piattaforma software orientata ai system e software integrator che realizzano soluzioni CTI (Computer Telephony Integration). La piattaforma è costituita da strumenti di sviluppo e moduli applicativi che coprono le esigenze tipiche dell'integratore software.

Le applicazioni telefoniche automatiche realizzate con la piattaforma Tempest possono sostituire gli operatori telefonici ed essere di ausilio agli operatori del call center.

Questo grazie all'impiego di innovative tecnologie vocali che rendono "naturale" il dialogo telefonico con l'utente.

La piattaforma Tempest unisce le tecnologie telefoniche e quelle informatiche offrendo strumenti che risolvono le complessità di integrazione tra i due "mondi".

L'interfaccia tra il mondo del software ed il mondo telefonico, fornita da Tempest, prevede diversi approcci che sono legati al tipo di applicazione ed il livello di integrazione che si vuole realizzare.

Tempest fornisce un'integrazione con il mondo telefonico che va dal semplice canale telefonico analogico collegato alla centrale pubblica ai canali digitali Isdn (BRI / PRI) sino ai collegamenti basati sulla Voice Over IP.

Strumenti per sviluppatori software

Gli strumenti offerti dalla piattaforma Tempest prevedono un approccio multi livello che parte dalle tecnologie abilitanti (speech recognition e text to speech) e va verso moduli applicativi completi. Rispondono sia alle esigenze di coloro che vogliono aggiungere funzionalità evolute alla propria piattaforma telefonica, sia di coloro che per la prima volta si affacciano al mondo dei call center e dei portali vocali.

Oltre agli strumenti che sono rivolti ai programmatori software, è presente un modulo di alto livello, basato su interfaccia grafica, orientato alla creazione di portali telefonici con funzionalità vocali. Semplici wizard guidano, passo-passo, un utente non programmatore sino alla realizzazione dell'applicazione.

Architettura della piattaforma

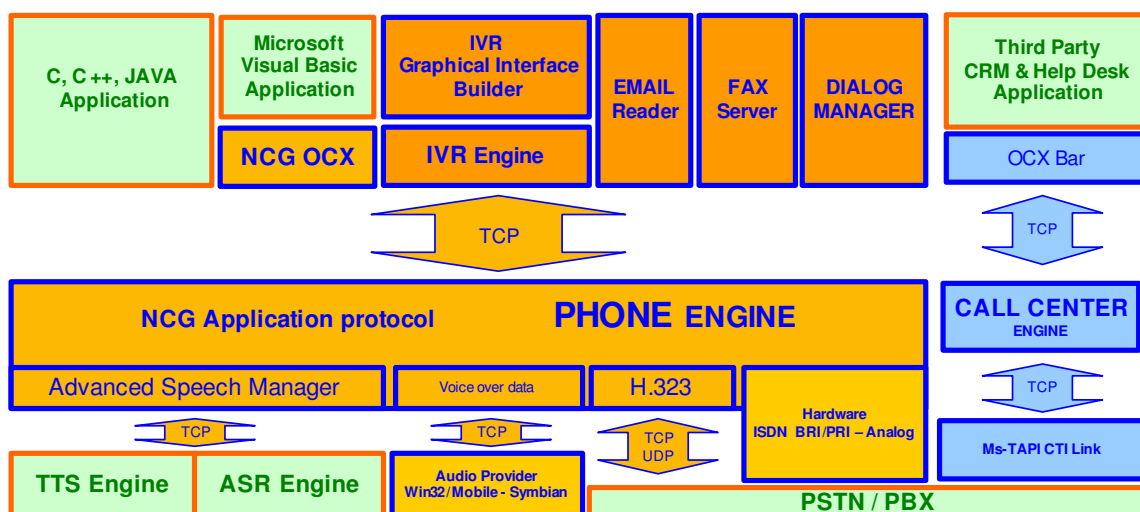
Struttura modulare e distribuita

L'architettura software è modulare e si basa sulla distribuzione dei moduli sulla rete intranet attraverso il protocollo TCP/IP. La struttura modulare prevede lo sfruttamento massimo delle piattaforme hardware e dei sistemi operativi.

I vantaggi di tale modularità sono molteplici:

- Costi di implementazione
- Scalabilità
- Load-Balance per il bilanciamento dei carichi applicativi
- Fault Tolerant per la disponibilità continuità dei servizi erogati

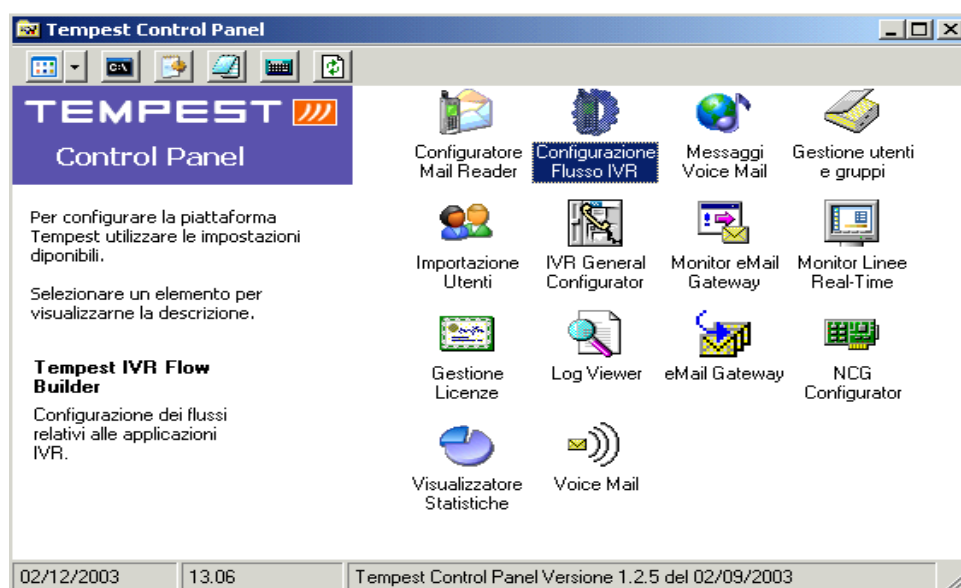
Di seguito viene illustrato per mezzo di uno schema la disposizione dei moduli e l'interoperabilità di questi per realizzare sistemi complessi e distribuiti su più calcolatori.



Caratteristiche dei moduli principali

Ogni modulo/applicazione è fruibile da un apposito strumento denominato Tempest Control Panel dove l'utente ha accesso alle applicazioni descritte e ad ulteriori strumenti per il monitoring real-time o configuratori grafici.

Segue un esempio del pannello di controllo della piattaforma Tempest:



Phone Engine

È il cuore della piattaforma Tempest e realizza l'interfaccia tra il software applicativo e le tecnologie telefoniche e di speech (speech recognition e text to speech).

Colloquia con il software applicativo attraverso un linguaggio di alto livello(NCG) via protocollo Tcp/IP. Il linguaggio/protocollo NCG (Network Command Gateway) è composto da una serie di comandi che vanno impartiti verso il Phone Engine ed una serie di eventi che il Phone Engine invia all'applicazione.

Le tipologie di comando ed eventi con una descrizione di massima sono:

Comandi di tipo applicazione:

- *Registrazione del modulo software sul Phone Engine*
- *Abilitazione / inabilitazione a ricevere eventi telefonici*

Comandi di tipo linea:

- *Risposta ad una chiamata*
- *Abbandono chiamata*
- *Chiamata telefonica*
- *Trasferimento della chiamata ad un'altra linea*
- *Trasferimento del controllo della chiamata ad un'altra applicazione*
- *Switch tra due linee telefoniche*

Comandi di tipo I/O:

- *Lettura di file testo tramite text to speech multi lingua con possibilità di caratterizzazione della voce*
- *Riconoscimento vocale di comandi e frasi contenute in dizionari predefiniti (speech grammar)*
- *Lettura e contemporaneo riconoscimento vocale con supporto del "barge-in" che consente l'interruzione della lettura del messaggio quando l'utente pronuncia una frase, un nome o un qualsiasi comando vocale (senza l'attesa del "beep" di fine messaggio).*
- *Lettura e registrazione di messaggi di tipo audio (wave files)*
- *Lettura di cifre, numeri, ore e date (senza l'uso del text to speech)*
- *Acquisizione, emissione e gestione del buffer di toni DTMF*
- *Analisi dell'attività vocale presente sulla linea*
- *Ricezione ed invio di documenti via fax*

Eventi:

- *Comando eseguito*
- *Chiamata telefonica entrante*
- *Passaggio di controllo chiamata da altra applicazione*
- *Abbandono chiamata / cornetta abbassata*
- *Errore*

Il modulo Phone Engine è l'applicazione che fornisce una interfaccia di alto livello che consente allo sviluppatore software di concentrarsi esclusivamente sulle funzionalità applicative.

Inizialmente il Phone Engine è stato progettato e realizzato per l'interfacciamento con le linee telefoniche tradizionali analogiche e digitali (Isdn Bri e Pri) verso le centrali pubbliche o Pbx. L'interfacciamento avviene tramite schede telefoniche di diversi vendor:

- NMS Communications (Analog - Bri - Pri - FAX)
- Dialogic (Analog - Bri)
- Donjin (Analog)

Successivamente è stata realizzata una nuova versione del Phone Engine orientato all'interfacciamento tramite protocollo IP utilizzando una interfaccia propria con trasporto Tcp ed una standard con trasporto misto Tcp/Udp basata sullo standard H.323/H.450 per il call control e G.711 per codifica della voce.

L'appendice "A" descrive le modalità di interfacciamento IP con diverse centrali.

L'integrazione delle tecnologie vocali avviene in modalità diversa tra le due versioni di Phone Engine.

Phone Engine per schede telefoniche

La versione con interfacciamento "telefonico" colloquia via protocollo Tcp con i moduli risorsa di Speech Recognition e Text to Speech. Le due risorse possono essere installate su calcolatori differenti. Ciò per le installazioni basate su flussi primari consentendo di concentrare più canali possibili sulla macchina di front-end lasciando il carico computazionale legato alle risorse di speech alle due macchine dedicate.

Le risorse supportate sono le seguenti:

Speech Recognition:

- Itc Irst SPINET
- Loquendo ASR
- Nuance ASR

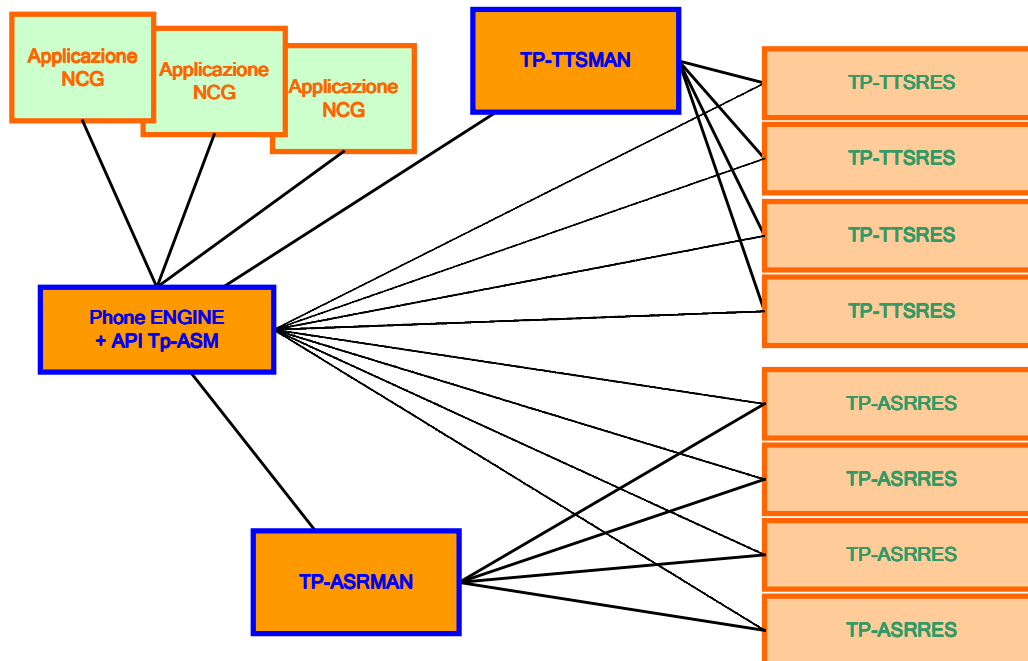
Text to speech:

- Loquendo TTS
- Nuance REALSPEAK

Phone Engine IP H.323

La versione con interfacciamento "IP" è più modulare ed integra le tecnologie di speech attraverso i moduli distribuiti Advanced Speech Manager (TP-AsmTts e TP-AsmAsr) che negoziano, per conto del Phone Engine, le richieste vocali asr e tts con i moduli risorsa (TP-AsrRes e TPTtsRes) che possono essere distribuiti su diversi calcolatori ed utilizzare motori asr e tts di diversi vendor contemporaneamente. Ciò consente di configurare architetture fault tollerant e con funzionalità di load balancing.

Struttura dell'interfacciamento tra Phone Engine e moduli Manager e Risorsa:



Phone Engine IP “proprietario”

Nel Phone Engine versione “IP” di tipo proprietario (denominato Voipick) sono stati aggiunti comandi ed eventi NCG per la trasmissione di dati tra front-end ed applicazione che vengono utilizzati nello sviluppo di interfacce vocali per dispositivi mobili quali i Pda (ad esempio la comunicazione di un codice a barre letto da un dispositivo portatile in un applicazione di Voice Picking).

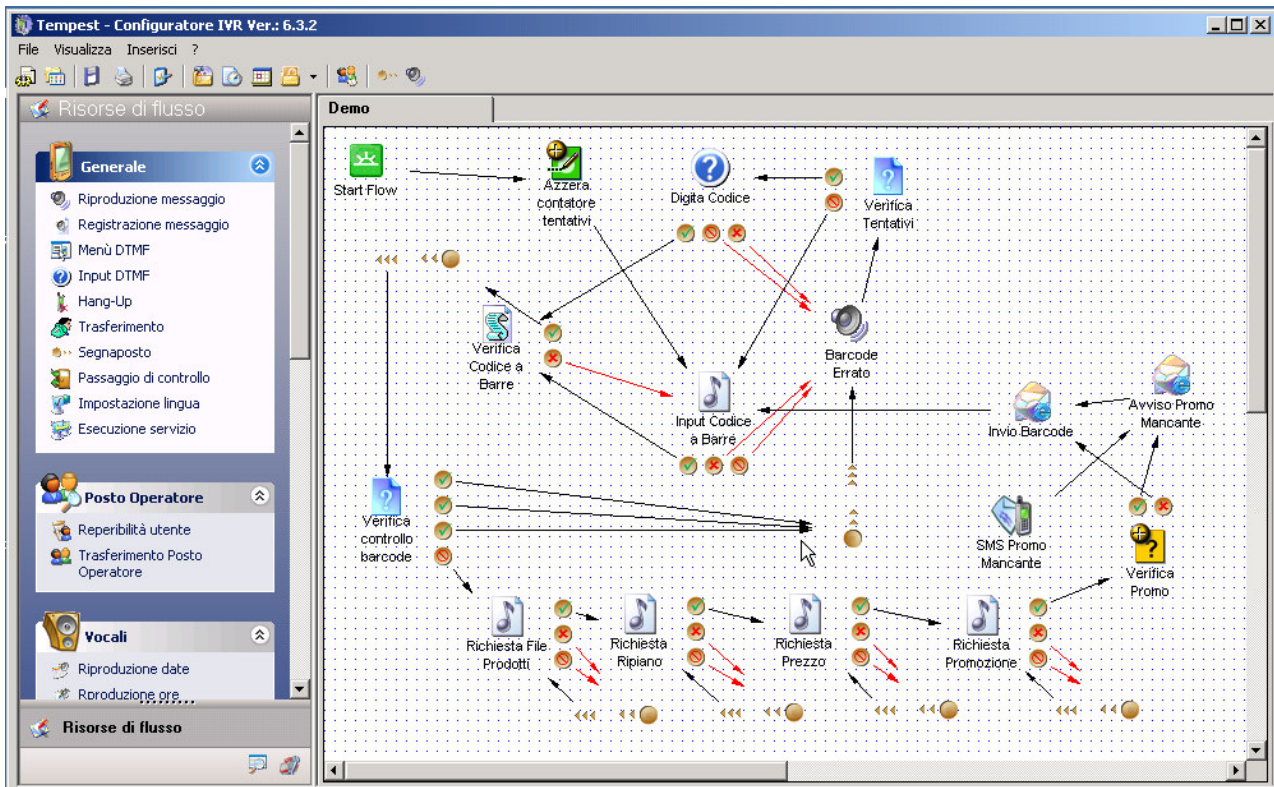
Sono state realizzate interfacce di tipo Audio Provider per diversi sistemi operativi quali:

- Windows 32 bit
- Windows Mobile
- Symbian (in fase di ultimazione)

Tali interfacce prevedono l'utilizzo del “Push to talk” e consentono lo sviluppo di applicazioni vocali e multimodali su dispositivi portatili quali i Pda e gli Smartphone.

Speech Portal (IVR Engine e Configuratore grafico)

Il modulo IVR Engine è il motore applicativo di più alto livello, interpreta i flussi vocali “disegnati” tramite una applicazione basata su interfaccia grafica.



Le funzionalità sono molteplici e coprono diverse esigenze con diversi livelli di complessità dipendenti dalla tipologia di servizio che si vuole realizzare.

Può essere configurato da semplice posto operatore automatico che instrada gli utenti attraverso alberi informativi, trasferire le chiamate ad interni di un centralino o a numeri di cellulari, oppure essere integrato con il sistema informativo aziendale (tramite pagine XML, scripting o chiamate SQL) per realizzare portali vocali telefonici che sfruttano le tecnologie di speech recognition e text to speech.

La navigazione da parte dell'utente del sistema può avvenire tramite selezioni vocali (anche multi lingua) o per mezzo della tastiera del telefono (DTMF).

I messaggi vocali possono essere pre-registrati (WAV files) o generati da sistema di sintesi vocale (text to speech). Anche questa modalità può essere in multi-lingua.

La modalità di interazione con le scelte vocali (menu vocali) prevede il supporto delle funzionalità di “barge-in” per l'interruzione dei messaggi del sistema nel momento in cui l'utente pronuncia una frase, un nome o un qualsiasi comando vocale senza attendere il “beep” di fine messaggio.

Il sistema si integra inoltre con i più comuni sistemi di messaging quali email, Fax, SMS, per la generazione di architetture per l'accesso e la notifica multicanale e unified messaging.

Esegue in contemporanea una pluralità di servizi per i quali la chiamata viene instradata grazie alla definizione di regole di routing basate ad esempio su:

- Numero chiamato (DID)
- Numero chiamante (CID)
- Canale fisico chiamato
- Orari
- Altri...

Ad ogni servizio telefonico è possibile associare la generazione automatica delle statistiche dettagliate sull'utilizzo

Servizi basati sulle selezioni vocali

La navigazione vocale all'interno avviene tramite la definizione di oggetti definiti "Selezioni Vocali".

Le selezioni vocali vengono create attraverso l'interfaccia grafica di Tempest Speech Portal e prevedono l'utilizzo di semplici "liste di comandi e vocaboli" e grammatiche "complesse" abbinabili tra di loro tramite "espressioni regolari".

Un portale vocale basato su semplici interazioni vocali (comandi o piccole frasi) può essere realizzato semplicemente utilizzando le selezioni vocali basate sulle liste di comandi definibili con la sola interfaccia grafica.

Un portale vocale basato invece su complesse interazioni vocali (es. frasi composte dalle quali estrarre i concetti per definire la strategia di dialogo) viene realizzato sfruttando al massimo le potenzialità dei motori di riconoscimento scrivendo manualmente le grammatiche su "files di grammatica". Tali "files di grammatica" verranno richiamati all'interno dell'oggetto "Selezione vocale". L'interpretazione dell'output del motore di riconoscimento potrà, ad esempio, essere "passato" ad un oggetto del tipo "VbScript" con il quale effettuare il "parsing" estraendone i dati/concetti da utilizzarsi per la prosecuzione del flusso applicativo.

Realizzazione di un servizio IVR di tipo Speech Portal

Un servizio Speech Portal realizza un sistema automatico di interazione con un utente, reagendo e rispondendo alle operazioni e alle richieste dell'utente stesso: per realizzare l'interazione, lo Speech Portal, dispone di una serie di funzionalità (es. interrogazioni a database, esecuzione di processi esterni, esecuzione di codice predisposto all'interno dell'engine, riproduzione di messaggi vocali). La definizione di un servizio comporta la definizione di un flusso operativo, composto da una sequenza di oggetti ("foglie" o "speech object") opportunamente predisposta per rispondere alle specifiche di queste funzionalità.

Il configuratore Speech Portal permette di costruire graficamente questo flusso: ogni funzionalità viene rappresentata da un'icona ('foglia') e quindi il flusso è costituito da

una serie di icone che indicano le funzionalità utilizzate opportunamente interconnesse tra loro.

Naturalmente tutte le funzionalità richiedono la definizione di alcuni parametri: tutte queste informazioni trovano ovviamente riscontro in un database che contiene la definizione di informazioni quali ad esempio:

- Servizi: contiene la descrizione dei servizi definiti all'interno del database (ricordiamo, infatti, che in un database, è possibile definire più servizi e quindi più flussi).
- Flussi: contiene tutte le foglie e gli speech object definite all'interno dei flussi disegnati nel configuratore.
- Tipo Risorse: contiene l'elenco completo delle foglie disponibili per la definizione di un flusso.
- TipoEventi: contiene l'elenco di tutti i tipi di evento definiti all'interno del sistema.
- Eventi: contiene l'elenco degli eventi definiti all'interno dei servizi configurati.

All'atto della definizione di un servizio, è possibile raccogliere alcune foglie in una macro, che può essere considerata come un sottoflusso del flusso principale: concettualmente questo processo è molto simile alla definizione di funzioni all'interno del codice di un programma, per raccogliere parti di codice che possono essere ripetute più volte nel corso dell'elaborazione o che realizzano una particolare operazione che si vuole mettere in evidenza. Nel caso del configuratore "Speech Portal", la raccolta di alcune foglie in macro può anche essere utile per semplificare il disegno grafico di un flusso.

Altri elementi grafici sono relativi alla definizione delle festività (Master Festivities – Festivities), di particolari periodi della settimana (Master Schedules – Schedules), es. weekend, ore lavorative, ecc. dei messaggi registrati da riprodurre in varie fasi di un servizio (Classes – TTSVoices – Encoding – Messages) e delle variabili di sessione (Variables) utilizzate all'interno di un servizio, di un flusso o di una macro.

Ad ogni elemento del flusso può essere associata una "label" di statistica tramite la quale sarà possibile realizzare analisi statistiche sulle modalità di utilizzo di ogni singolo servizio.

La gestione delle statistiche è implementata da un apposito manager dove il gestore dei servizi è in grado di eseguire query mirate, creare modelli statistici e salvarli per elaborazioni successive o particolarmente complesse.

Elementi dinamici. L'accesso a fonti dati esterni e le variabili di sessione

Il configuratore garantisce funzionalità di integrazione verso il back-end grazie alla possibilità di inserire come elementi di flusso "foglie" per l'accesso a dati esterni, la definizione di "variabili di sessione" e costrutti tipici di un linguaggio di sviluppo per la loro valutazione come ad esempio "if then else", "switch" ecc..

La natura e la tecnica di accesso può essere approssiata secondo specifiche esigenze a partire da semplici query SQL (Select, Insert, Update, Delete, ecc.), interfacciamento e

scambio di strutture dati complesse XML via protocollo "http", interfacciamento via Socket TCP fino alla scrittura per utenti esperti di vere e proprie procedure in linguaggio VbScript.

Le variabili di sessione potranno essere valutate come parametro per le modalità di accesso sopra descritte.

Dialog Manager

E' il gestore delle applicazioni telefoniche realizzate per il motore di dialogo adattativo ad iniziativa mista realizzato dall'Irc-Irst di Trento (ora Fondazione Kessler) ed integrato nella piattaforma Tempest.

Un'applicazione di dialogo è costituita essenzialmente da un'insieme di grammatiche, e da una descrizione di dialogo. Entrambe devono essere caricate da un sistema costituito da diversi moduli, nel quale la componente principale è un processo - l'interprete di dialogo - che carica una descrizione e la esegue. La descrizione richiede via via l'esecuzione di alcuni comandi, che consentono all'interprete di dialogo di comunicare con l'esterno in maniera sincrona. I comandi più importanti sono:

- la sintesi di un messaggio
- il riconoscimento con una certa grammatica
- una richiesta ad un database esterno
- operazioni telefoniche varie (on-hook, off-hook)

Descrizione di dialogo

Un'applicazione di dialogo è caratterizzata dall'insieme dei contesti e dei singoli concetti che permettono l'introduzione dei dati, e dall'insieme delle procedure che descrivono le azioni da intraprendere nei vari casi. Le informazioni relative a contesti/concetti definiscono non solo i tipi di dati che possono essere introdotti, ma anche il linguaggio che può essere accettato in ogni stato del dialogo.

Questo documento definisce il formato dei file di descrizione. Alcuni di essi sono pensati come indipendenti dall'applicazione (ad esempio i moduli che rappresentano la strategia di dialogo, i moduli che processano alcuni tipi di dati ragionevolmente stabili come date, ore, etc.), altri sono invece strettamente dipendenti dall'applicazione (definizione dei concetti, azioni da intraprendere in caso di riempimento di contesti, etc).

Informazioni generali

Alcuni file di descrizione definiscono i contesti, i concetti e le procedure che verranno utilizzati nel corso di un'applicazione di dialogo. A partire dai file di descrizione verrà generato un file perl desc.pl, incluso dal modulo di dialogo, contenente una serie di variabili che rappresentano tutta la struttura. Per modificare una descrizione è

necessario prima editare almeno uno dei file di descrizione, quindi richiamare il programma perl BuildDescr.pl (o un qualche suo parente - vedi Tool2Dial), che ricostruisce il file desc.pl.

I file sono descrizioni in formato ascii, che devono rispettare una certa sintassi. Il parser BuildDescr.pl non è particolarmente sofisticato, per cui è opportuno porre una certa attenzione nella scrittura del file. In particolare, è opportuno conoscere le regole generali che seguono.

Commenti: sono ammessi sia i commenti tipo C /* (come quelli che delimitano questa parentesi) */ sia i commenti tipo shell (è commento tutto quello che sta tra un cancelletto ed il newline seguente). Non è salutare usarli nidificati, il comportamento è indefinito.

commento tipo shell

Ordine dei campi: le varie definizioni non devono più seguire strettamente l'ordine prestabilito (in una versione precedente, ad esempio, per prime erano definite le variabili scalari, poi le variabili array, poi le variabili associative, etc.; questo ordine non poteva essere modificato). Questo vincolo rimane solo per i campi all'interno dei contesti e delle procedure.

Uso delle virgolette: i valori di alcuni campi possono avere come delimitatori o le virgolette (") o l'apostrofo (') o nulla (separatori saranno blank, tab, newline). Quando si usa un tipo di separatore, gli altri separatori possono essere contenuti nell'espressione. L'unica avvertenza è che per i campi multipli (ad esempio descrizione di procedure) il separatore deve essere lo stesso (" o ' o blank) per tutti i campi, che vengono letti assieme. Nota: Si consiglia l'uso delle virgolette per delimitare tutti i campi non composti da una sola parola.

Uso dell'accento: siccome il carattere di accento (') viene utilizzato nella composizione di un file Perl, se si utilizza tale carattere (ad esempio in una label, o in una stringa) è necessario proteggerlo con un backslash "\". Ad esempio, la stringa "l'orario" deve essere scritta come "\l'orario" e verrà riscritta in Perl come "\'orario".

Messaggi di errore: in caso di errato parsing dei file, viene generato un messaggio che cerca di descrivere il contesto in cui l'errore è stato rilevato; la generazione del file desc.pl viene interrotta.

I file di descrizione

Il file che costituiscono una descrizione di dialogo (tutti i files -eccetto qualche residuo di emacs- contenuti nelle dir Descrizione/ e Trial/ e aventi spesso nomi come app.txt, azioni.txt e processi.txt) contengono alcuni elementi, la cui sintassi verrà definita più avanti:

- definizione variabili globali scalari
- definizione variabili globali array
- definizione variabili globali associative
- definizione contesti-concetti
- definizione procedure

Pur non essendo necessario seguire lo schema seguente, può essere utile considerare la seguente strutturazione dei file di descrizione.

Le definizioni appena introdotte sono logicamente distribuite sui tre file `app.txt`, `azioni.txt` e `processi.txt`, che costituiscono la descrizione o parte di essa. Le variabili globali possono essere definite, seguendo le regole di una buona programmazione, in ognuno dei tre file.

I contesti e i concetti vengono normalmente definiti in `app.txt`. Si può pensare ad un contesto di dialogo come ad un vero e proprio frammento di dialogo a sè stante, solo occasionalmente collegato agli altri contesti che costituiscono l'intera descrizione. I collegamenti tra i diversi contesti di dialogo avvengono tramite un meccanismo descritto più avanti e chiamato salto di contesto (che può essere top-down o bottom-up). Ogni contesto è costituito da un insieme di unità atomiche chiamate concetti. L'esecuzione di un dialogo prevede l'attraversamento dei concetti del contesto attivo, ripetutamente, e l'attivazione di uno di essi.

Durante l'attivazione di un concetto vengono richieste delle azioni esterne, che normalmente portano l'utente a dire una frase che permette al dialogo di "riempire" il concetto attivo. Il file `processi.txt` contiene le procedure che permettono di manipolare la stringa riconosciuta durante il riempimento di un concetto, ad esempio per normalizzare un numero, una data o altro (es: cento venti sette --> 127).

In uscita da un contesto, vengono normalmente eseguite delle azioni (file `azioni.txt`) che portano all'attivazione di un altro contesto, talvolta eseguendo un'elaborazione dei concetti raccolti nella fase precedente, magari richiedendo un accesso ad un database esterno, o preparando dei dati che dovranno essere gestiti nel prossimo contesto di dialogo (ad esempio una lista di elementi tra cui scegliere).

Questo meccanismo consente la creazione di applicazioni di dialogo estremamente sofisticate, anche perché tutti i campi dei contesti - concetti sono a tutti gli effetti delle variabili, i cui valore può essere modificato run-time in base a quanto succede durante il dialogo stesso.

Appendice A)

In questa appendice vengono descritte le modalità di connessione del modulo Phone Engine VoIP di Tempest con tutte le centrali PBX finora testate e sulle specifiche del protocollo H.323 da esse supportate.

Modalità di integrazione Voip

Per il momento il modulo VoIP di Tempest è stato integrato con le seguenti soluzioni:

- Modulo VoIP (senza alcun componente aggiuntivo)
- Modulo VoIP e Gatekeeper
- Modulo VoIP e MediaProxy

Modulo VoIP senza alcun componente aggiuntivo

Il protocollo di segnalazione VoIP implementato nel modulo è l'H.323. Tale protocollo gestisce i trasferimenti di chiamata sia ciechi che con consultazione mediante il protocollo H.450. Questa tipologia di funzionamento è utilizzabile dunque con tutte quelle centrali PBX che supportano il protocollo H.323 con le funzionalità avanzate dell'H.450.

Un vantaggio dell'impiego di questa soluzione è il fatto che l'accentramento dei flussi voce delle chiamate trasferite è demandato alla centrale PBX.

Modulo VoIP e Gatekeeper

Alcune centrali PBX supportano i protocolli H.323 e H.450 solo nell'interazione con un Gatekeeper esterno, per tale motivo è stata realizzata una soluzione con un Gatekeeper opensource installato sullo stesso server Windows sul quale si registrano sia il modulo VoIP che la centrale PBX.

Come la soluzione precedente anche questa ha il vantaggio di demandare l'accentramento dei flussi voce delle chiamate trasferite alla centrale PBX.

Modulo VoIP e MediaProxy

Questa soluzione nasce per garantire l'interazione con tutte quelle centrali PBX che supportano esclusivamente la funzionalità di chiamata del protocollo H.323 e dunque non sono in grado di effettuare trasferimenti di chiamata attraverso il protocollo H.450.

Il MediaProxy è un servizio che a livello di segnalazione si interpone tra la centrale PBX ed il modulo VoIP gestendo i trasferimenti di chiamata e accentrando i flussi voce delle chiamate trasferite.

MediaProxy è un'applicazione Linux che supporta sia il protocollo H.323 che SIP. Tale componente consente dunque alla piattaforma Tempest di interagire anche con le centrali PBX che parlano SIP.

Il MediaProxy può essere installato su un server esterno a quello della piattaforma Tempest oppure su una macchina Linux emulata su VMWare installata sullo stesso server di Tempest. Quest'ultima soluzione è consigliata per le aziende con poche utenze.

A fronte del vantaggio di poter interagire con la maggior parte delle centrali PBX (SIP/H.323) presenti sul mercato, tale soluzione presenta lo svantaggio di demandare l'accentramento dei flussi voce delle chiamate trasferite ad un componente esterno alla centrale PBX ossia al MediaProxy, è dunque necessario dimensionare il server che ospita tale componente in funzione del carico di chiamate gestito dal cliente.

Considerazioni

Lo studio di queste tre diverse soluzioni di integrazione nasce dal fatto che la maggior parte delle centrali PBX i cui vendor dicono di supportare il protocollo H.323 in realtà non lo supportano pienamente o meglio garantiscono esclusivamente la chiamata base mentre per quanto riguarda le funzionalità di segnalazione più evolute come il trasferimento di chiamata spesso impongono l'utilizzo del loro protocollo proprietario (vedi CISCO e Alcatel) e dunque la progettazione software basata sulle TAPI specifiche di quella centrale.

Laddove dunque non sia possibile l'impiego del protocollo H.323/H.450 punto punto o attraverso la presenza di un gatekeeper, si consiglia l'utilizzo della soluzione con il MediaProxy che con un server opportunamente dimensionato garantisce di soddisfare le esigenze di aziende piccole e grandi.

La tabella che segue riassume i risultati dei test funzionali effettuati con alcune centrali Voip. Come si può notare, le funzionalità complete di trasferimento di chiamata (cieco e supervisionato con e senza consultazione) sono supportate in maniera diversa tra le diverse centrali testate. La soluzione con MediaProxy garantisce tali funzionalità su tutte le centrali.

	<i>Cisco VoiceGateway (CCME)</i>	<i>Cisco CalManager</i>	<i>Promelit Ipecs</i>	<i>Avaya IpOffice</i>	<i>Alcatel OmniPcx</i>	<i>Asterisk</i>
Tempest Voip (CallControl H.323)	Solo trasferimento cieco	NO	NO	✓	NO	Solo trasferimento cieco
Tempest Voip + Gatekeeper (CallControl H.323)	Solo trasferimento cieco	NO	✓	✓	NO	Solo trasferimento cieco
Tempest Voip + MediaProxy (CallControl: Sip, H.323, IAX, SCCP)	✓	✓	✓	✓	✓	✓

Aziende che utilizzano la piattaforma Tempest

Aurora Assicurazioni (Mi)	Posto operatore automatico basato su speech recognition
Comune di Genova	Posto operatore automatico per l'instradamento delle chiamate verso circa 5.000 interni tra uffici e personale dipendente
ENEA (Rm)	Posto operatore automatico.
Irc-Irst (Tn)	Diversi progetti europei per applicazioni telefoniche vocali di dialogo adattativo ad iniziativa mista
Olio Carli (Im)	Sistema di acquisizione telefonico degli ordini CarliTel completamente automatico e sistema di acquisizione ordini semi-automatico basato su casella vocale con trascrizione manuale via browser web.
ASL4 Chiavari (GE)	Chiamate automatiche per conferma visite/appuntamenti.
Sitel (Mi)	Piattaforma telefonica per l'erogazione di servizi di quotazione titoli per diverse banche tra le quali: <ul style="list-style-type: none">- Societ� Generale- Credit Lyonnais- JP Morgan- ABN Amro
Lombardini (Bg)	Sistema di VoiceMail integrato con la Collaboration Suite di Oracle
Abla (Mi)	Piattaforma telefonica per l'erogazione di servizi di portale vocale
Carta Si (Mi)	Servizio telefonico vocale automatico per la ricarica delle carte telefoniche dei gestori di telefonia mobile integrato con il middleware Genesys.
Il Secolo XIX (Ge)	Portale vocale per le informazioni di cronaca, meteo, oroscopo, farmacie di turno
Comune di Firenze (Fi)	Posto operatore automatico per l'instradamento delle chiamate verso circa 2.700 interni tra uffici e personale dipendente
American Standard (Mi)	Risponditore telefonico integrato con erp Sap per il tracking delle consegne effettuate dai trasportatori.
Voicevolution (Rm)	Piattaforma telefonica su 120 canali per l'erogazione di servizi di portale vocale su modello ASP tra i quali: <ul style="list-style-type: none">- Posto operatore automatico- Campagne automatiche di outbound- Sondaggi ed interviste automatiche- Acquisizione ordini- Ricerca punti vendita
Arclinea (Vi)	Servizio di ricerca telefonica automatica del punto vendita pi� vicino Posto operatore automatico
Europ Assistance (Mi)	Sistema telefonico per la gestione completamente automatica del servizio "salva chiavi"
Sirti (Roma)	Servizio di ricerca su telefono cellulare dei tecnici reperibili per guasti sulla rete autostradale del circuito Isoradio.
Comune di Vado Lig. (Sv)	Sistema IVR
Cooperativa Edile (Sv)	Posto operatore automatico.

Labso Telecom (Mi) Posto operatore automatico.

Aziende che utilizzano il call center Tempest

- Cassa Forense (Roma)** Contact center integrato con la centrale telefonica Avaya ed il software crm per fornire informazioni automatiche e tramite operatore
- Aurora Assicurazioni (Mi)** Call center integrato con la centrale telefonica Tenovis
- Kpmg (Mi)** Call center integrato con la centrale telefonica Voip Panasonic e software legacy crm ed helpdesk
- Gas Energia (To)** Call center integrato con la centrale telefonica Panasonic e software legacy crm ed helpdesk su host AS/400
- Cogei (Bo)** Call center integrato con centrale Philips
- Package Sistemi (AI)** Call center integrato con centrale telefonica Tenovis e software help desk
- RadioTaxi 4Mori (CA)** Call center integrato con centrale telefonica Panasonic