Computational Linguistics and Chinese Language Processing
Vol. 9, No. 2 , August 2004, pp. 77-94

An Innovative Distributed Speech Recognition Platform for Portable, Personalized and Humanized Wireless Devices

Yin-Pin Yang[*]

Abstract

In recent years, the rapid growth of wireless communications has undoubtedly increased the need for speech recognition techniques. In wireless environments, the portability of a computationally powerful device can be realized by distributing data/information and computation resources over wireless networks. Portability can then evolve through personalization and humanization to meet people's needs. An innovative distributed speech recognition (DSR) [ETSI, 1998],[ETSI, 2000] platform, configurable DSR (C-DSR), is thus proposed here to enable various types of wireless devices to be remotely configured and to employ sophisticated recognizers on servers operated over wireless networks. For each recognition task, a configuration file, which contains information regarding types of services, types of mobile devices, speaker profiles and recognition environments, is sent from the client side with each speech utterance. Through configurability, the capabilities of configuration, personalization and humanization can be easily achieved by allowing users and advanced users to be involved in the design of unique speech interaction functions of wireless devices.

Keywords: Distributed, speech recognition, configurable, wireless, portable, personalized, humanized.

1. Introduction

In the current wireless era, cellular phones have become daily-life necessities. People carry their own handsets and make phone calls anytime, everywhere, while public payphones have

[*] Ph.D, Senior Researcher, Advanced Technology Center, Computer and Communications Research Laboratories, Industrial Technology Research Institutes
E-mail: YinPinYang@itri.org.tw TEL: 886-3-5914830 FAX: 886-3-5820098
Address: E000 CCL/ITRI Rm.712, Bldg.51, 195 Sec.4, Chung Hsing Rd., Chutung, Hsinchu 310, Taiwan.

almost disappeared. Inspired by this vast number of mobile phone users, the wireless communication industry is developing wireless data services to create more profit. Wireless devices can be treated as terminals of an unbounded information/data network – the Internet. However, the small screen sizes of mobile devices discourage users from surfing the Internet in mobile situations. Wireless data services are not as attractive as was expected, and this is one of the major reasons for the so-called "3G Bubble" [Baker, 2002][Reinhardt *et al*, 2001].

On the other hand, the handset market is still blooming. Personal, stylish and fashionable features, such as ring tones, color screen displays, covers, and so on, are all very popular, especially among teenagers. Functionally speaking, portable devices, such as PDAs, pocket/palm PCs and digital cameras, are now integrated with handsets. Many interesting applications, such as portable electronic dictionaries, map navigators, and mobile learning, can be built into mobile devices. However, these functions or services still cannot create serious business opportunities for telecom companies.

What will future appealing services for cell phones be? "Talking to a machine," or interacting with a machine, might be a candidate. That is, besides talking to human-beings through voice channels, people may like to talk to machines and access the Internet through data channels. The possibilities are unlimited. Handsets may thus evolve into personal "intimate pets" that people will use from childhood to grownup. In this scenario, speech interaction will play an important part in humanizing devices [Hiroshi et al. 2003]. However, due to the limitations of the current state-of-art speech recognition techniques, the robustness issue [Deng *et al.* 2003][Wu *et al.* 2003][Lee 1998] is always a bottleneck in commercializing speech recognition products. This imperfection reveals the importance of configurability. In the following paragraphs, the relationships among configurability, personalization, and wireless environments will be explored.

Speech Recognition and Wireless Environments

How does a speech recognition system fit into a the wireless network? In this paper, we will primarily highlight two key terms "distributed" and "configurable." The term "distributed" can be interpreted as follows: computation distributed and data distributed. As for the former, normally speech recognition functions are needed in mobile situations, and devices are usually thin and lacking in computational power. It would be much easier to design speech recognition functions if the computation involved in recognition processes would be distributed over wireless networks by means of a client-server architecture. As for the latter, speech recognition is by nature a pattern matching process, which needs to acquire utterances within a given application domain. For example, a speaker-independent (SI) continuous digit recognizer targeting the Taiwanese market needs to acquire a great large number of sample continuous digit utterances from all dialects in this market. The representation and quality of the sample utterances used for training or adaptation can seriously dominate the performance of a speech recognizer. If a wireless network is used, speech data acquisition can be done in a much more efficient and systematical way. More importantly, the acquired speech data, labeled by means of a speaker profile, recognition environment, and device/microphone type, can be kept on the server. Speech data will thus not be abandoned when particular applications or services are discontinued.

From the above, we can conclude that we need a centralized speech recognition server embedded in the wireless infrastructure. When we say “talking to a machine”, the “machine” is actually an entire wireless network. People talk to the same lifetime recognizer, and the recognizer evolves continuously. This speech recognition server can provide any types speech recognition services (computation distributed) for all classes of wireless mobile devices. These services continuously acquire speech data from all locations (data distributed), and adapt the engine performance all the time. For each recognition task, there is a “configuration file” (or, say, a tag) to record all of the information regarding types of services, speaker profiles, recognition environments, etc. We call this type of server a configurable distributed speech recognition (C-DSR) server.

In the following, the history of DSR developed by ETSI/Aurora will be briefly described. Then, the innovative C-DSR platform will be introduced.

Distributed Speech Recognition (DSR) developed by ETSI/Aurora

Instead of squeezing the whole recognizer into a thin device, it seems more reasonable to host recognition tasks on a server and exchange information between the client and server. However, due to the low bit-rates of speech coders (note that coders are designed for humans, not recognizers), the speech recognition performance can be significantly degraded. The DSR, proposed by ETSI Aurora, overcomes these problems by distributing the recognition process between the client and server, by using an error protected data channel to send parameterized speech features.

From DSR to Configurable DSR (C-DSR)

Aurora DSR can be seen as a speech “coder” [Digalakis *et al.* 1999] designed to enable handset users to talk to their recognizers. Besides handsets, there are many other mobile devices that need DSR services, and they all operate in different environments and in different recognition modes. Each combination or, say, configuration, needs its own “coder” to achieve better performance. Based on these needs, C-DSR was built as an integrated client-server platform which not only offers a convenient way to construct speech recognition functions on various client devices, but also provides powerful utilities/tools to assist each configuration to obtain its own coder in order to increase the overall recognition task completion rate. To achieve these goals, C-DSR maximizes the advantages of data channels and centralized servers by means of its “configurable” capability - *configurability*. The configurability can be considered from two points of views.

The C-DSR Client

From the client side viewpoint, speech recognition processing is configurable to work with: (1) various kinds of thin to heavy mobile devices, ranked according to their computational power, noting that most of devices do not have sufficient computational power to perform the feature extraction process proposed by ETSI Aurora; (2) various types of recognition environments, such as offices, homes, streets, cars, airports, etc; this information about recognition environments can help recognition engines achieve greater accuracy; (3) various types of recognition services, such as command-based, grammar-based, speaker independent/ de--pendent mixed mode, and dialogue style services; (4) various speaker profiles since speaker information can help recognizers achieve higher recognition rates[1] and are required by recognition applications, such as speaker adaptation [Lee *et al.*1999][Chen *et al.*1990], speaker verifications identification [Siohan *et al.*1998]. The C-DSR platform provides a faster and more flexible way to construct various speech recognition functions for various mobile devices used in various recognition environments. One of the major missions of C-DSR is to increase the frequency of speech recognition use in daily life.

The C-DSR Server

From the viewpoint of the centralized server, the C-DSR server collects from all of the registrant clients speech utterances or formatted speech feature arrays along with their configuration tags. The basic idea is to formalize the life cycle of a speech recognition product/task from the deployment phase, to the diagnostic phase, tuning phase, and upgrading phase. Also, similar tasks can share corresponding information and adaptation data located on the server. The C-DSR server offers the following mechanisms to fully take advantage of these categorized speech and configuration data: (1) the C-DSR server can decide which recognition engine or which acoustic HMM model to employ according to the history log; (2) the C-DSR server can balance the trade-offs among communication bandwidth, system load and recognition accuracy; (3) the categorized and organized speech database can be utilized to create diagnostic tools that can be used to tune-up recognition engines and to perform all kinds

[1] For example, if we may provide gender information from speaker profile to speech recognizer, even a first-time speaker can obtain higher recognition accuracy.

of adaptation, such as speaker adaptation, channel adaptation [Siohan *et al.*1995] and background noise adaptation [Kristjansson *et al.*2001].

In summary, C-DSR is a generic speech recognition engine, in which all of the information, or parameters, concerning application-dependent user profiles and device profiles are kept in a configuration file which is initiated at the client side. In technical terms, C-DSR is a platform, while from customer’s point of view, C-DSR is a personally owned, lifetime speech recognition engine. The C-DSR platform is embedded in the wireless network in contrast to conventional speech recognizers that are treated as input interfaces for portable devices (see Figure 1).

Overview

In the following, the architecture of C-DSR is described in section II. Then in section III, a demo system is presented to demonstrate the unique capability, that is, configurability, of C-DSR. Some conclusions are drawn in the final section.

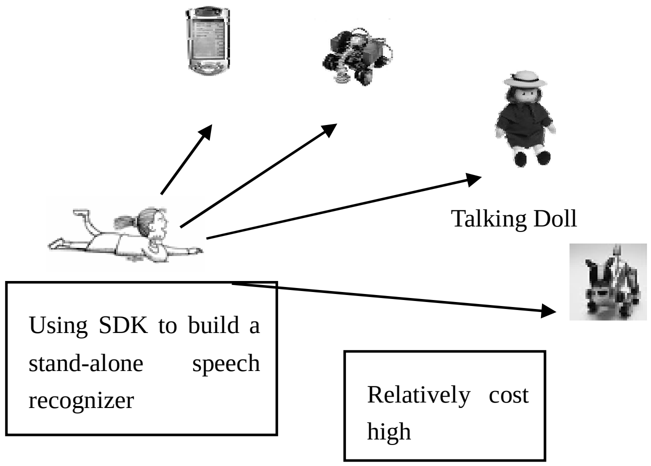
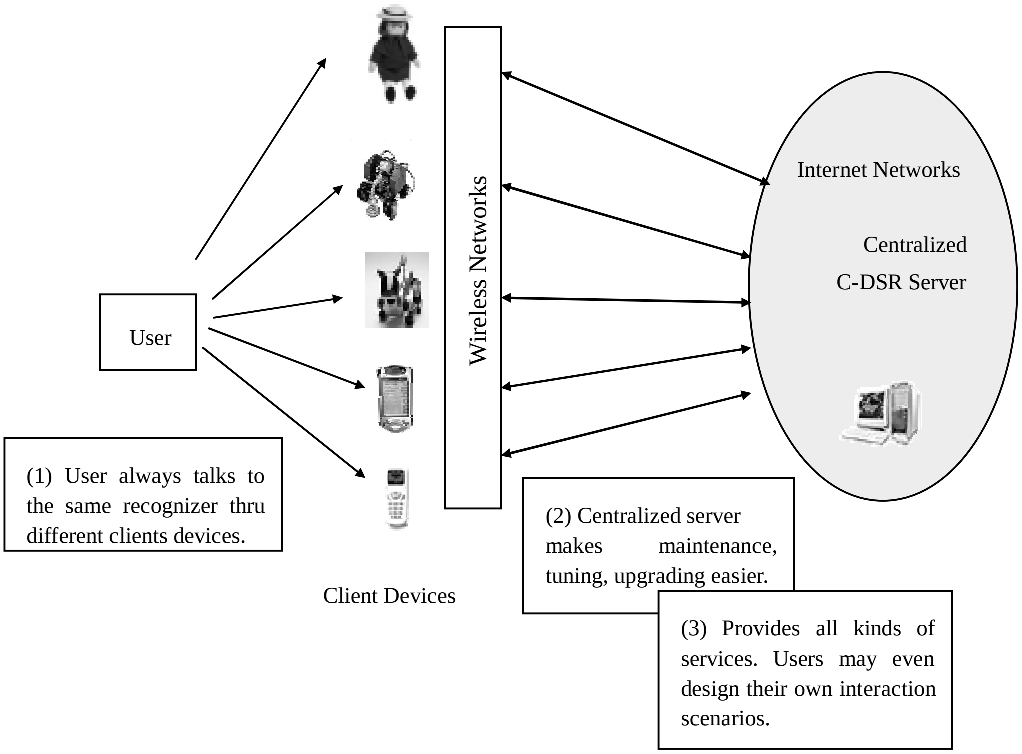


Figure 1(A). Conventionally, speech recognition is simply one of the available input interfaces for portable devices.



(1) User always talks to the same recognizer thru different clients devices.

(2) Centralized server makes maintenance, tuning, upgrading easier.

(3) Provides all kinds of services. Users may even design their own interaction scenarios.

Figure 1(B). Illustration of the innovative C-DSR platform.

2. The C-DSR Architecture

The function blocks of the C-DSR development platform are shown in Figure 2. A wireless device equipped with the C-DSR Client connects to a remote C-DSR Server using the C-DSR Protocol through a wireless network. The C-DSR Protocol sends speech data and parameters. The speech data can be in the form of PCM raw speech, ADPCM or pre-defined compressed speech features, depending on the computation power and bit-rate (communication bandwidth). The configuration file with speech data prepared by the client is, thus, transmitted by the C-DSR Protocol thru wireless networks. For now, the C-DSR Protocol is implemented on top of TCP/IP or RTP (Real Time Transit Protocol). After parsing the received protocol, the Configuration Controller (CC) decides how to configure the recognition engine (C-DSR Engine) and Dialogue System (DS) to accomplish the recognition task. The C-DSR engine and DS engine are composed of modulized components such that switches inside the engines can be shifted to corresponding components to perform the functionalities requested by the configuration. The recognition results are then logged and organized in the History Log Center

(HLC), resulting in a formatted package, or a database. The package is then passed to the Diagnostic Center (DC), where, diagnostic tools are used to tune-up the engines and provides adaptation data for various kinds of adaptation schemes, such as speaker/channel adaptation.

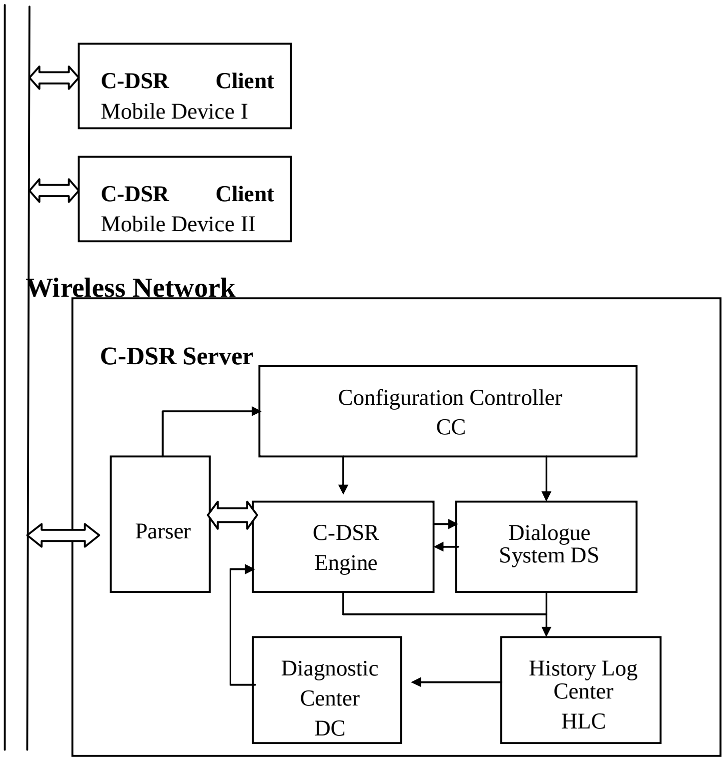


Figure 2. The function blocks of the C-DSR Platform.

A. Configuration File and Configuration Controller, CC

The configuration of the C-DSR platform is stored in a Configuration File (CF). Each field in the CF can be categorized into three attributes, r SR, r DSR, and Interactive Design-It-Yourself (I-DIY), which are explained below.

i. r SR is the difference between the perfect and a current practical state-of-art speech recognition engine. The SR engine is never perfect. However, if we can restrict the speaking styles of the users, the recognition accuracy will be higher.

ii. r DSR refers to those configurable parameters which can minimize the degradation due to wireless transmission loss.

iii. IDIY means Interactive Design-It-Yourself. We can never expect a machine to act exactly like a human being. The philosophy behind C-DSR is to make human-machine interaction simple and easy. The best way to achieve this goal is to involve users in design. Thus, we provide DIY tools to enable users to design their own ways of talking to machines.

Table 1. Configuration file

Configuration, Configurable Parameter		Attribute
SrCoder	Bit-rate: 0.1/1/4/16/64/128 Kbps	r DSR
computationPower (deviceType)	very-thin/thin/medium/heavy	
Expandable	…	
NoiseType	home/office/street/in-vehicle	r SR
microphonelType	US5/US20/US100/US200	
searchEngine	Full/Fast-mode	
voiceActivated	Yes/No	
endPointDetect	Yes/No	
speakingSpeed	Fast/medium/slow	
speakingAccent	Taiwan/china/foreigner	
Expandable	…	
vocabularySetUp	Vocabulary	I-DIY
grammarSetUp	Grammar (ABNF/cdsr_format)	
dialogueSetUp	Dialogue Script (VXML/AIML/cdsr_format)	
PersonalitySettings	Talktive/Quiet/Shy/…	
Expandable…	…	

The original design principle behind the C-DSR platform is to remotely configure the speech recognition engine on the server from the client side. It is the client device that initially prepares all of the configuration files. However, some of the configuration parameters may not be fully determined by the client device or may even be totally empty. In this case, the CC of the server should be able to append or modify these parameters by utilizing all available resources, including the historical configurations or statistics located in the server.

B. Configurable Engine

The configurable engine is the heart of the C-DSR server. As the name indicates, a configurable engine is an SR engine which is modulized and can be configured according to different requests requested from the various types of clients; Figure 3 shows the modules of the engine, which is a generalized SR engine on which state-of-art SR techniques can be employed. These typical modules (in a traditional/generalized SR engine) are listed below:

Table 2. Configurable modules in the C-DSR Engine with their parameters

Configurable Module	**Parameter**
Energy Normalization	None / FRAME_ENG_NORM
Front-end filter	FF_NONE / FF_LOW_PASS / FF_1POLE_1ZERO
Feature Extraction (if needed)	FE_NONE / FE_MFCC / FE_LPC_CEP / FE_8051_CLASS
End Point Detection	EP_NONE / EP_VFR / EP_ENG
Engine Type	EG_DIGITSTRING/EG_COMMAND/EG_KEYWORDSPOT/G_LVCSR
Mean Subtraction Computing	MS_NONE / MS_STD
HMM Adjustments	HJ_NONE / HJ_PMC
HMM Adaptations	HP_NONE / HP_ADAPT_DEV / HP_ADAPT_SPKR
Viterbi Searching	VS_FULL_PATH / VS_NBEST / VS_BEAM
Post Operations	PO_NONE / PO_STD

As shown in Table 2 above, each module has a well-defined interface and, for a particular module, several implementations are available. To each implementation, one CF name is attached, and it can be switched or configured. For instance, in the End Point Detection (EPD) module, there are three options, EPD_NONE, EPD_VFR and EPD_ENG, each representing a different algorithm used to implement the EPD function. The C-DSR platform also allows the system maintainer to adopt a new method for each module.

Intermediate data between modules are also generated to provide “symptoms” useful for diagnostic purposes. These symptoms include:

§ speech segmentation boundaries,

§ the Viterbi resulting path,

§ likelihood trajectories along the time axis on the resulting path,

§ a histogram of observations (feature vectors) for a particular Gaussian mixture,

§ a histogram of the observing likelihood of a particular HMM state

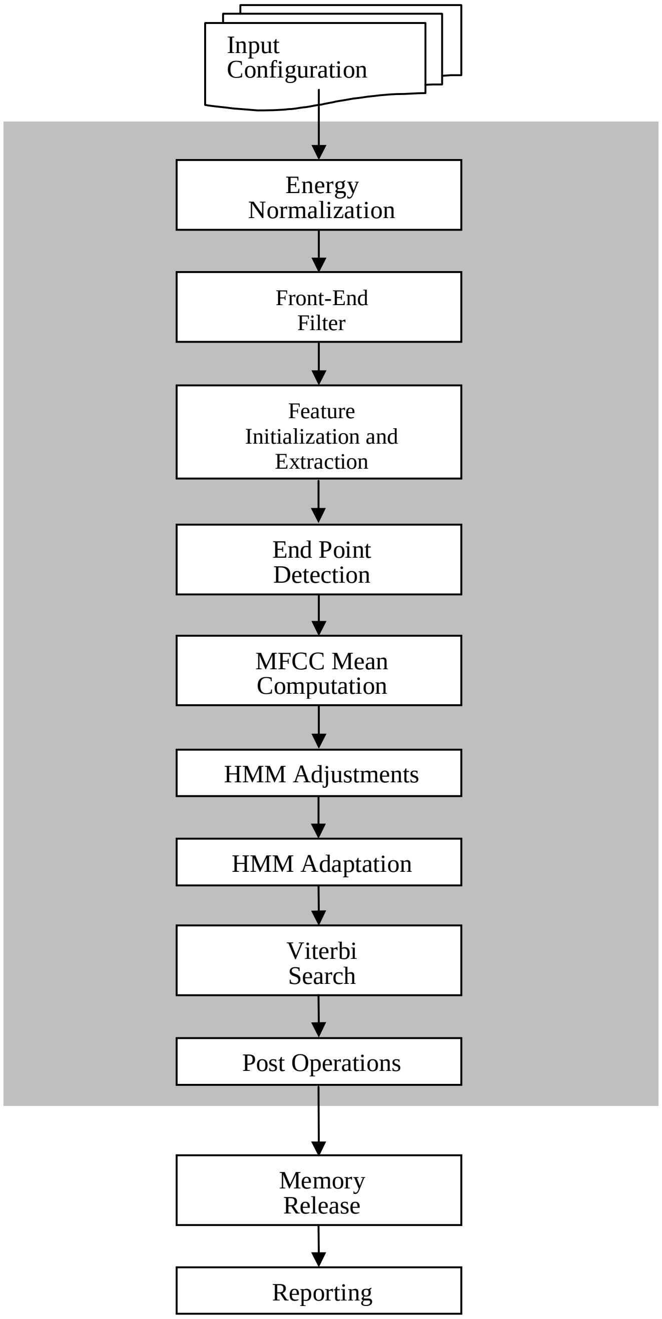


Figure 3. Modules of the C-DSR engine.

C. The Dialogue System, DS

A generic DS is, firstly, responsible for preparing a grammar, including vocabulary needed for the next speech recognition process. Then, the grammar with the incoming utterance is fed to the recognizer. The recognition result, a recognized key word, is then sent back to the DS. The DS then updates the “dialog status” records and determines the grammar for the next recognition.

Currently, we support AIML (Artificial Intelligence Markup Language, www.alicebot.org) and the simplified VoiceXML format used to describe dialogue scripts (see Figure 4).

D. The Diagnostic Center, DC

As described earlier, the so-called symptoms are intermediate data obtained while the engine is running and sent to the DC. The main purpose of the DC is to analyze and diagnose these symptoms in order to make suggestions. Thus, the C-DSR server is faced with various types of services, various environment configurations, and various types of client devices and speakers, so we want to make the engine core to be as generalized as possible. Currently, all of the diagnostics are done manually, which means that the DC only display to users or C-DSR server maintainers. We plan to make the DC automatically in the next generation of C-DSR.

E. The History Log Center (HLC)

The HLC is responsible for collecting and logging all of the corresponding information for each recognition service. The information collected includes speech utterances, formatted feature arrays, configuration files and the intermediate data, that is, symptoms and recognition results, and is saved to a corresponding user directory according to the user registration ID. The HLC serves as a database manager, whose job functions includes: (i) maintaining the database, if necessary, and creating a mechanism to eliminate garbage data; (ii) building data links to prepare adaptation data for various types of adaptation algorithms, for speaker or channel adaptation; (iii) preparing intermediate data for the DC to diagnose, so that the DC can provide data for the C-DSR engine to perform to tune algorithms and improve recognition accuracy.

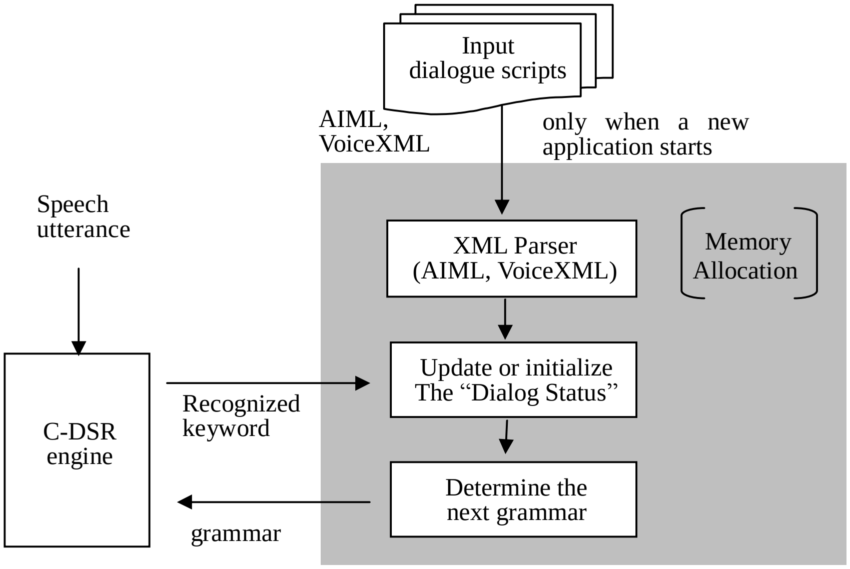

Input
dialogue scripts
AIML,
VoiceXML
only when a new application starts
Speech utterance
XML Parser (AIML, VoiceXML)
Memory Allocation
Update or initialize The "Dialog Status"
C-DSR engine
Recognized keyword
Determine the next grammar
grammar

Figure 4. Diagram of the Dialogue System, DS.

3. C-DSR Demo System

In a laboratory, several speech recognition applications may be emulated by a PDA serving as a client on the C-DSR platform. These recognition applications are usually realized on stand-alone portable devices or server-based dialogue systems. Now, using the proposed C-DSR solutions, thin client devices can take advantage of powerful, wireless servers to perform sophisticated speech recognition functions (see Figure 5), including the following:

§ car agent – retrieving map/travel/hotel information through GPRS network in a car;

§ a personal inquiry system – a portable device which can retrieve stock/weather information anywhere through a GPRS network;

§ general-purpose remote control – in a WLAN 802.11b environment, a remote control which can be used to control a TV, stereo, air-conditioner, etc., through infrared rays by using natural language commands;

§ Sim-Librarian – a portable device, which, when a person walks into a library, can be used to ask for directions or for the location of a book the person is searching for.

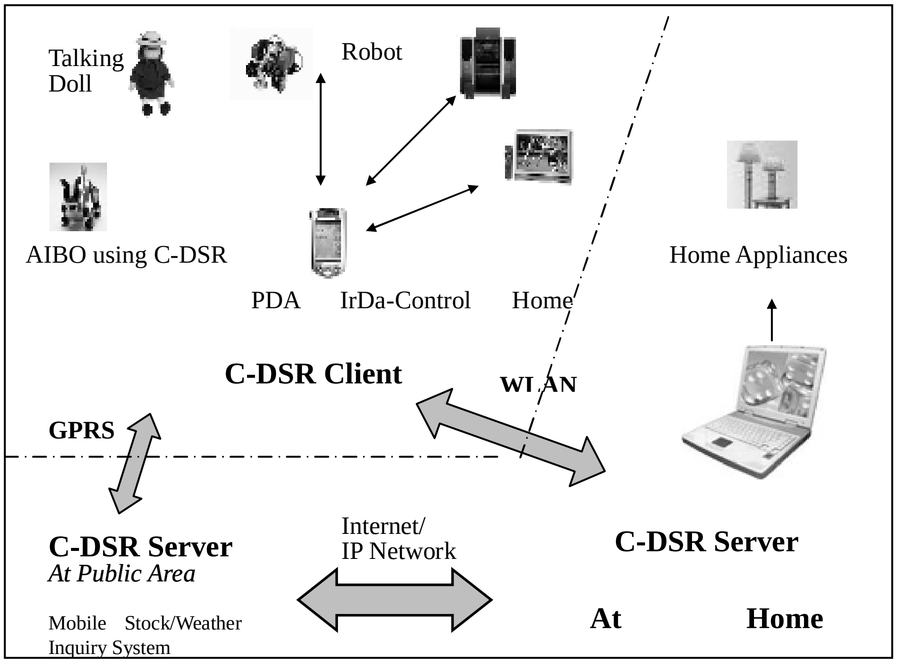


Figure 5. Illustration of the C-DSR implementation.

Each application uses its own configuration, specified according to (1) the device type: from thin to heavy, 8-bit 8051 class, DSP, PDA class; (2) the recognition style: command-based, natural, or dialogue; (3) the recognition environment: in a car, home, or open space. Two configuration files are presented below to illustrate how configuration files are used to realize a speech recognition application. Note that, normally, there are two types of speech recognition applications: voice command-based and dialogue style.

[Example 1] Voice-controlled home appliances

[Scenario] The user may use his wireless portable device with the installed C-DSR client to control a TV, lamp, or other home appliances within a WLAN environment.

[Configuration Settings]

1. Speech Feature Compression Format: this may be PCM, 8051-class, LPC-based Cepstrum, or MFCC (Mel-Frequency Cepstral Coefficients), depending on the computational cost and communication bandwidth (bit-rates) of the client device.
2. Environmental noise: this can be Quiet or Noisy. If this is skipped, the C-DSR Server will make a decision according to the history log.

3. Speaking speed: the speaking speed of a normal person is around five words per second. The user can determine his range of speaking speed, for instance, from three words per second to six words per second. If this is skipped, the C-DSR server will use default values.
4. Gender/Age/Accent: gender, age and accent information are very helpful for improving recognition performance. The C-DSR Client will retrieve and pass these pieces of information from the user/speaker profile to C-DSR Server for reference purposes. If this is skipped, the C-DSR Server will employ default models.
5. The Number of Results: the user may configure the number of recognition results, say N. The C-DSR Client will then display the first most likely candidates to the user.
6. Recognition Style: this can be Command-based or Dialogue. The grammar format for the Command-based style uses the ABNF format shown in the following:

```
#ABNF 1.0
$prefiller= 請 | 麻煩 | 你 | 我要
$action1= 開{open} | 關{close}
$keyword1= ( 燈{light} | 電燈{light} | 風扇{fan} | 電
風扇{fan} | 電視{tv} | 收音機{radio} )
```

[The Setup at the C-DSR Server]

When the configuration file and speech data are received from the client, the C-DSR Server performs recognition tasks according to the configuration. In the grammar example shown above, exactly one keyword from the $action1 group (open/close) and $keyword1 (light/fan/tv/radio) group will be generated. The Action Center on the C-DSR Server will perform a corresponding action, such as “turn on light.”

[Example No.2] Tourist Information Guide of Yu-Shan National Park

[Scenario] Users may use their own PDAs or smart phones to access this service when entering the area covered by the WLAN.

[Configuration Settings]

As in the previous case, we only need to change the field RecognitionStyle from Command-based to Dialogue-ProvidedByServer.

[The Setup at the C-DSR Server]

This example shows a dialogue system for a tourist guide. The content was provided by

Yu-Shan National Park. The C-DSR Platform provides several Dialogue Scripts. Here, we use VoiceXML as an example.

```
[CDSR_VXML]
<form_id="”tourinfo_agent”>
        <field name="hello">
           <prompt>您好，旅遊導覽精靈在此爲您服務</prompt>
           <grammar src="howRU.gram"    type="application/grammar+xml"/>
         </field>
        <field name="caragent">
           <prompt>很好，謝謝</prompt>
           <prompt>馬馬呼呼啦，謝謝</prompt>
           <prompt>可以啦，謝謝</prompt>
           <grammar src="tourinfo.gram" type="application"/>
           <filled>
           <if cond="caragent == 'resource'">
                <prompt>森林型態以柳杉，天然闊葉林樟樹，台灣杜鵑爲主</prompt>
                <prompt>動物型態有松鼠，穿山甲，台灣獼猴，台灣野兔等</prompt>
                <prompt>鳥類型態有綠繡眼，小鶯，巨嘴鴉，五色鳥等</prompt>
                <clear namelist="tourinfo.gram"/>
            </if>
            <if cond="caragent == 'facilities'">
                <prompt>遊客中心：內設餐飲部，會議室，多媒體簡報室及生態教育展示館</prompt>
                <prompt>餐廳：除可同時供一百人用餐外，並可作爲大型會議室，教室使用</prompt>
                <prompt>行政管理中心：爲本區工作人員處理行政事務的辦公地點</prompt>
                <clear namelist="tourinfo.gram"/>
            </if>
            <if cond="caragent == 'spot'">
                 <prompt>化石區：此生痕化石是大約三萬年前蝦，蟹類進行築穴工事時所遺留而成
                            </prompt>
                <prompt>造林紀念石：爲前新竹山林管理所大溪分所，在民國四十四年爲紀念
                            東眼山造林工作實績而建</prompt>
                 <prompt>親子峰：在林道終點上方，有大小雙峰，猶如慈母帶著小孩，故名親子峰
                            </prompt>
                <clear namelist="tourinfo.gram"/>
            </if>
            </filled>
         </field>
        <block>
            <submit next="theTop.vxml" namelist="city state"/>
            <prompt>好吧，掰</prompt>
            <prompt>好吧，下次再聊囉，掰</prompt>
        </block>
</form>
```

4. Conclusions

Speaking of wireless mobile devices, conventionally, speech recognition is considered to be one of the available input methods for these devices. In this paper, we have presented the client-server C-DSR platform which is a centralized speech recognition server embedded in the wireless infrastructure. By using C-DSR, people talk to the same lifetime speech recognition system. Speech data and the corresponding configuration, which keeps all the records about recognition environments, device information, dialogue scripts, recognition results, and so on, will not be abandoned when particular applications or services are discontinued. The speech recognition server provides many types of services for all classes of wireless mobile devices, and these services continuously acquire speech data from all locations, and adapt the engine performance all the time.

Personalization and humanization are essential. We have seen many successful products come on the market. A humanized device does not have to be “intelligent.” As long as it “looks” intelligent and people find it interesting, we do not really need to make such a machine act exactly like a human being. People like to have their own ways to interact with their own personal devices/pets. Perhaps the Design-It-Yourself approach, getting people involved in the design process, is one good solution, and the “configurability” of C-DSR can surely provide such a platform to meet these needs.

References

ETSI Doc. No. ES 201 108, Ref. RES/STQ-00018, STQ Aurora, “DSR Front End”.

ETSI ES, Version 0.1.1, Ref. DES/STQ-00030, STQ Aurora, “Front-End Extension for Tonal Language Recognition and Speech Reconstruction”.

Baker, S. , “A Tale of A Bubble, ”Business Week Magazine, International Cover Story, June 3, 2002.

Reinhardt, A. , W. Echikson, K. Carlisle, P. Schmidt, ”Who Needs 3G Anyway?” Business Week Magazine, International – European Business, March 26, 2001.

Yamaguchi, H. , K. Suzuki, C. V. Ramamoorthy, “The Humanization, Personalization and Authentication Issues in the Design of Interactive Service System,“ 2003 Society for Design and Process Science, www.sdpsnet.org.

Deng, L. , J. Droppo, and A. Acero, “Recursive Estimation of Nonstationary Noise Using Iterative Stochastic Approximation for Robust Speech Recognition,“ in IEEE Transactions on Speech and Audio Processing. Volume: 11 Issue: 6 , Nov 2003.

Wu, J. , J. Droppo, L. Deng and A. Acero, “A Noise-Robust ASR Front-End Using Wiener Filter Constructed from MMSE Estimation of Clean Speech and Noise,“ in Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding. Virgin Islands, Dec, 2003.

Lee, C. H. , “On stochastic feature and model compensation approaches to robust speech recognition,” Speech Communication, 25:29-47, 1998.

Digalakis, V. , L. Neumeyer and M. Perakakis, “Quantization of Cepstral Parameters for Speecg Recognition Over the World Wide Web,” IEEE Journal on Selected Areas in Communications, Jan. 1999, volume 17, pp 82-90.

Lee, C. H. , C. H. Lin, and B. H. Juang, “A study on speaker adaptation of continuous density HMM parameters,” Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pages 145-148, Albuquerque, New Mexico, April 1990. ICASSP'90.

Chen, K. T. and Hsin-min Wang, “Eigenspace-based Maximum A Posteriori Linear Regression for Rapid Speaker Adaptation,” in Proc. IEEE Int. Conf. Acoustics, Speech, Signal processing (ICASSP'2001), Salt Lake City, USA, May 2001.

Siohan, O. , A. E. Rosenberg and S. Parthasarathy, “Speaker identification using minimum classification error training,” In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, Washington, USA, May 1998.

Siohan, O. , Y. Gong, and J. P. Haton, “Channel adaptation using linear regression for continuous noisy speech recognition,” IEEE Workshop on Automatic Speech Recognition, Snowbird, Utah, USA, December 1995.

Kristjansson, T. , B. Frey, L. Deng and A. Acero. “Towards Non-Stationary Model-Based Noise Adaptation for Large Vocabulary,” in Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing. Salt Lake City, Utah, May, 2001.