ヒューマンインタフェース 84−16
情 報 メ デ ィ ア 35−16
(1999. 8. 20)

Voice Windows: A Voice-Based Platform for the Visually Disabled

TUNEYOSHI KAMAE ,[†1] TAKAYUKI WATANABE ,[†3] TOMIO KOIDE ,[†6]
HIROHIKO HONDA ,[†4] SHINICHIROU UNO ,[†5] TOORU KURIHARA [†2]
and SAWAKO TAJIMA[†7]

We are developing a voice-based platform that enables visually disabled university students and professionals to use personal computers at higher levels and write application programs on Windows 98.

We find many softwares in market that assist the visually disabled in word-processing and browsing web pages. In university, however, students are required to advance to higher levels of computer use, including programming on standard OS's such as the Microsoft Windows and UNIX. Voice Windows is aimed to assist students and professionals in writing application softwares with standard audio user interfaces (AUI's) and graphic user interfaces (GUI's).

Voice Windows focuses on use of script languages, such as Visual Basic Script (VBScript), Visual Basic Application (VBA), and Emacs-Lisp. They are interpreted line-by-line and allow access to most of Windows 98 standard utilities. Voice Windows supplies a voicified set of I/O interfaces, utilities, and help files, all with large-format GUI's for users with low vision. Through them, visually disabled users can write VBA codes to manupulate Microsoft's Word, Excel, and Access. The human interfaces are designed to be equally friendly to the sighted and the sight-disabled, and the programs developed on Voice Windows will be suitable for use in public terminals of community centers and libraries.

1. Introduction

Personal computers have long been used by visually disabled professionals and students. In the age of the Disk Operating System (DOS), I/O operations were processed by character-based interfaces. Voicification with a synthesizer and presentation on a braille display were relatively straightforward: In fact, visually handicapped programers developed sophisiticated application programs. In the US, T. V. Raman, a visually disabled graduate student at Cornell University, developed Emacspeak* and opened a path for the disabled to develop expertise in computer science and write application programs[1]. In Japan, Masao Saito who lost sight at pre-school age wrote a widely used Japanese text reader (VDM100) that has an efficient Kana-Kanji translator and a homophonic Kanji identifier[2].

Usefulness of personalized computers had become widely recognized by the visually disabled through use of text readers such as VDM100[3]. Students learned programming on DOS and several application softwares were developed by visually disabled programmers in Japan and elsewhere. When Microsoft Windows 95 became the dominant OS for personal computers, this situation has changed. The MS-DOS on its DOS window became incompatible with most existing screen readers and visually impaired users lost their platform. Visually disabled computer users were left with little choice but to live in the DOS world if they wish to write computer programs. Many new voice-based platforms appeared including 95Reader[4] for Japanese, and visually disabled users moved to the Windows 95 world. Programming on Windows 95/98 with Visual Basic, C, and Visual C++, has remained least optimized for sighted users and only few vision-disabled programmers write application programs now.

As Internet-based applications move continuously toward more elaborate GUI's, script languages that run on the Hyper-Text Markup Language (HTML) platform have gained popularity. They include Visual Basic Script (VBScript), Java Script (JScript), Perl (Perl for

†1 Department of Physics, School of Science, University of Tokyo
†2 Department of Information Sciences, Tsukuba College of Technology
†3 Information Science, Shonan Institute of Technology
†4 Institute of Space and Astronautical Sciences
†5 Faculty of Social and Information Sciences, Nihon Fukushi University
†6 Create System Development
†7 Internet Technology Research Committee

* Emacspeak package is at `http://simon.cs.cornell.edu/Info/People/raman/emacspeak/`.

Win32*). In 1997, Microsoft has released an engine (Windows Scripting Host) which supports these script languages independently of the HTML platform[5].

Emacs-Lisp has a much longer history than VBScript and JScript as a script language, and has been widely used in the UNIX community. The language is not easy to learn but has several merits: Its interpreter allows line-by-line debugging, its source codes are made public, and international free software volunteers offer support through Internet.

In 1994, T. V. Raman built a new voice-based system, ASTER (Audio System for TEchnical Readings)[1]. ASTER is a system that formats the content of an application into styles suited for audio presentation. For example, it does not read what is presented in a tabular form but reformats so that correlation among data can be understood in the serial audio rendering. Raman extended this approach to his monumental work, Emacspeak[1].

The present work has been motivated by T. V. Raman's Emacsspeak. One author (T.K.) began exploring possiblity of installing Emacspeak to Mule[6], the multi-lingual version of Emacs, and found found Mule for Win32 (Meadow) by H. Miyashita[7]. Test programs were written in Emacs-Lisp[8] to voicify with Toshiba's text-to-speech (TTS) engine **. Sample programs were also written for WSH in Visual Basic 6.0 and VBScript[8]. These efforts grew to the present Voice Windows project.

Several key features needed to be a useful system became known through the above prototype testing. The present Voice Windows include following components:

(1) Multitasking, multimodal, and multilingual Text-to-Speech System
(2) WSH-based platform: Voice Windows Scripting Host
    (a) Voicifying I/O and utility interfaces that can be called from VBScript and VBA.
    (b) VBScript and VBA sample programs.
    (c) Voicified script language editors.
    (d) Voicified dictionary reader.
(3) Emacs-based Platform: Voice Meadow
    (a) Voicified editor (Mule).
    (b) Voicified Unix utility programs.
    (c) Voicified dictionary reader.

The present paper describes overall project and basic features. Many components are still to be implemented at the time of manuscript writing. Their trial version will be leased in the fall of 1999.

2. Multitasking, multimodal, and multilingual Text-to-Speech System

Unix and Windows are multitasking OS's and their Text-to-Speech (TTS) systems are required to mix concurrent voice outputs from multiple tasks. Such concurrency is crucial in the following cases: Key echo and system messages while listening to a text; Dialog with a Japanese translator during the text input; Sharing the audio device with background music. Several audio chips (eg. Yamaha OPL3_SAx) installed in Windows 98 PC's have their WDM (Windows Driver Model) audio driver. With the WDM driver, multiple audio streams are mixed at the kernel-mode. We note that WDM drivers are not available for some audio chips now.

The TTS system of Voice Windows consists of a TTS server and TTS clients. Voicifying programs send messages to the common mailslot of the TTS server. Each message contains ID of the program, commands for the server and TTS engines, and texts to be spoken. This TTS system is described in a separate publication by Watanabe et al.[9].

The TTS engines are assumed to have the following features: Functions such as TTS pause, resume, and stop; Multiple TTS instances; Intext commands; User dictionary. Users are likely to work in the multilingual environment (eg. Japanese and English). The TTS system of Voice Windows accomodates a Japanese and an English TTS engine. When English texts are mixed with Japanese texts, the two engines coordinate in the synchronous mode. In other cases the two engines work in the asynchronous mode. Microsoft provides Speech SDK for Microsoft Win32 applications including an English TTS engine. Toshiba's Japanese TTS engine follows MS's Speech SDK specifications *** and hence meets the Voice Windows requirement. The present version of Voice Windows presuposes these two TTS engines to be installed.

* Consult `http://www.perl.com/Info/software.html` to obtain Perl for Win32.

** See http://www5.toshiba.co.jp/jdirect/sale/mimi.htm

*** See http://msdn.microsoft.com/msdownload/speechsdk.suite/speechsdk.suite.htm

Table 1 VW Adaptation of VBScript I/O Functions

VW Function (VBScript Func.)	Arg.	Comments
VceInputBox (InputBox)	message input	voicified returned
VceMsgBox (MsgBox)	message timeout	voicified avoid hangup
VceEcho (Echo)	message timeout	voicified avoid hangup
VcePopup (Popup) (WshShell Obj.)	message buttons timeout	voicified selections avoid hangup

We have developed a DirectSound TTS engine. This allows us not only to mix sounds but also to position sound sources in the 3-D space. At the time of manuscript submission, the engine is being updated to comply with MS's Speech SDK specifications.

3. WSH-Based Platform: Basic Components

VBScript has 3 basic I/O functions and Windows Scripting Host (WSH) has one object for basic I/O: Audio User Interfaces for these 4 functions are available (**Table 3**). The GUI of VceInputBox is shown in **Fig. 1**: Menu bars allow to optimize the voice speed, the voice kind, the color combination, the assistance levels, and others. These interfaces are written in Visual Basic 6.0 and made available as standard executables. Special features include transfer of arguments and returned values between user routines and AUIs in Visual Basic; synchronization of AUIs and GUIs; notification and control of highlight among input windows.

For young students who may wish to write animating programs, Microsoft's Agent characters are also voicified **Fig. ??**. They can be called from script programs and attract attension. We are developing new Japanese characters. Sound effects add a new dimension to programs. Dialog boxes of Voice Windows have built-in sounds that users can activate.

Visual Basic has 6 Common Dialog Boxes accessible to users. They are voicified with functionality listed in **Table 3**. Single function file access AUI's are available: they are designed to avoid possible misoperation.

VBScript makes many Objects accessible to users: their functionarity overlaps with Voice Windows functions. Windows Scripting Host also offers many more Objects through WScript.exe and WSHon.OCX to allow users access to Environment variables, Special Folders, and to create Shortcuts[5]. Voice Windows offers simpler alternative with AUI's (**Table 3**).

Table 2 Voice Windows Common-Daialog Boxes

Name	Arguments	Returned
VceFileOpenDlg	def. setting	path/attrib.
VceFileSaveDlg	def. setting	path/attrib.
VcePrinterDlg	def. setting	save setting
VceFontDlg	def. setting	name/type/size
VceColorDlg	def. setting	fore/back colors
VceHelpDlg	help file context	msg/pointers

Table 3 VW Adaptation of Selected WshShell Objects

VW Func. (WshShell Object)	Comments
VceEnvAll (Environment)	all environment items process or system
VceEnvItem (.Item)	selected environment item item specified
VceSpecFoldAll (SpecialFolders)	all special folders
VceSpecFoldItem (.Item)	selected special folders item specified
VceCreateShortcut (Shortcut)	creation of a shortcut file, icon loc., hotkey

Voice Windows keeps log of all access to its AUI's in a standard log file. All AUI's have their own argument type tables (declarations) and their argument transfer. These files are save for trouble shooting and also recalling its history.

4. Mule-Based Platform: Voice Meadow

Meadow is a multi-lingual version of Emacs (Mule)[6] that runs on Windows95/98[7]. Meadow serves as a universal platform for various GNU softwares including email utilities, Web browsers, dictionary viewers, data-base systems, text formatters, and text editors[9]. Since Meadow runs on Windows95/98, the TTS engines and AUIs developed for the WSH-based platform can be used.

Basic commands for text reading and status monitoring are voicified (**Table 4**). These programs are written in Emacs-Lisp and similar applications can be written by visually disabled computer users.

Another example is voicified version of a popular email software MEW. The code is straightforward as shown in **Table 4**. Visually disabled users can optimize by modifying or adding codes.

5. Future Prospects

Many more works are needed to make Voice

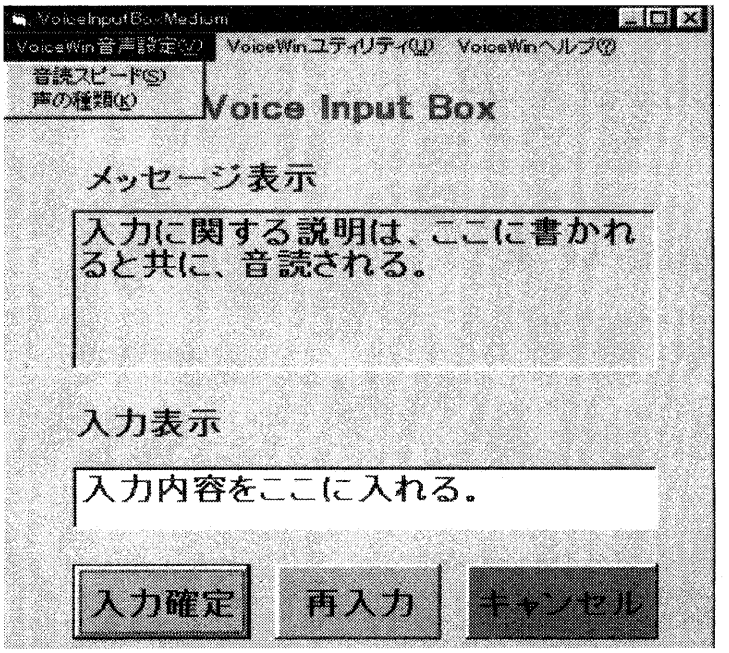

Fig. 1 VceInputBox window for user input.

Table 4 Emacs-Lisp Codes to Read Mode-Line

```
;;; voice-mode-line: モードラインを読む(高速版)
; minor-mode, japanese codes
;
(defun voice-mode-line ()
"Read mode-line."
(interactive)
; (view-mode) にする必要はない。
 (setq string1
   (concat """Eng:Eibun=0"" ""Spd=200""バッファ名"
         (buffer-name) " モード名" mode-name))
 (setq string2
   (concat """Eng:Eibun=1"" カーソルの位置 "
     (what-line) " コラム" (current-column)))
 (if (or
  (string= mw32-ime-mode-line-state-indicator "[—]")
  (string= mw32-ime-mode-line-state-indicator "[あ]"))
   (setq string3 """Vce=Gender=Female""日本語全角モード
"))
 (if (or
  (string= mw32-ime-mode-line-state-indicator "[O]")
  (string= mw32-ime-mode-line-state-indicator "[A]"))
  (setq string3 """Vce=Gender=Male""直接半角モード"))
 (if (string= mw32-ime-mode-line-state-indicator "[-]")
  (setq string3 """Vce=Gender=Male""直接半角モード"))
  (setq string (concat string1 string2 string3))
;;  (w32-set-clipboard-data string))
  (process-send-string "ttsserv" string))
(global-set-key "M-s"C-m" 'voice-mode-line)
```

Windows friendly with visually disabled users. Our intension is however not to present a completed package but rather a set of tools with which disabled users can develop a system they wanted, just as they wrote their own MS-DOS

Table 5 Emacs-Lisp Codes in Mew

```
(defconst voice-mew-version "voice-mew.el version 0.01")
(require 'mew)
(require 'voice-etc)
;
; voice-read-message: read message v2
;
; must use the information of Content-Type: Text/Plain;
; charset=iso-2022-jp
;
(defun voice-mew-read-message()
"Read message of mew"
(interactive)
(let (beg end original eol-point flag'h string'flag)
  (setq original (point))

  (save-excursion
    (setq string-header "")
    (setq string-tmp "")
    (sit-for 0 10)  ; reflesh screen

; search "Subject" header
    (goto-char (point-min))
    (if (re-search-forward "^Subject:" nil t)
      (progn (end-of-line)
        (setq eol-point (point))
          (setq string-header (concat
            """X=S1"" ""Eng:Eibun=1""
              ""Eng=Exkigo=0"" ""Eng:Kutou=0""
                (buffer-substring (match-end 0) eol-point)))
    (sleep-for 0 20)
    (process-send-string "TTSclient" """X=S3"" Subject")
    (setq flag'h t)
    (sleep-for 0 20)
    (process-send-string "TTSclient" string-header)))
;   (sleep-for 1)
```

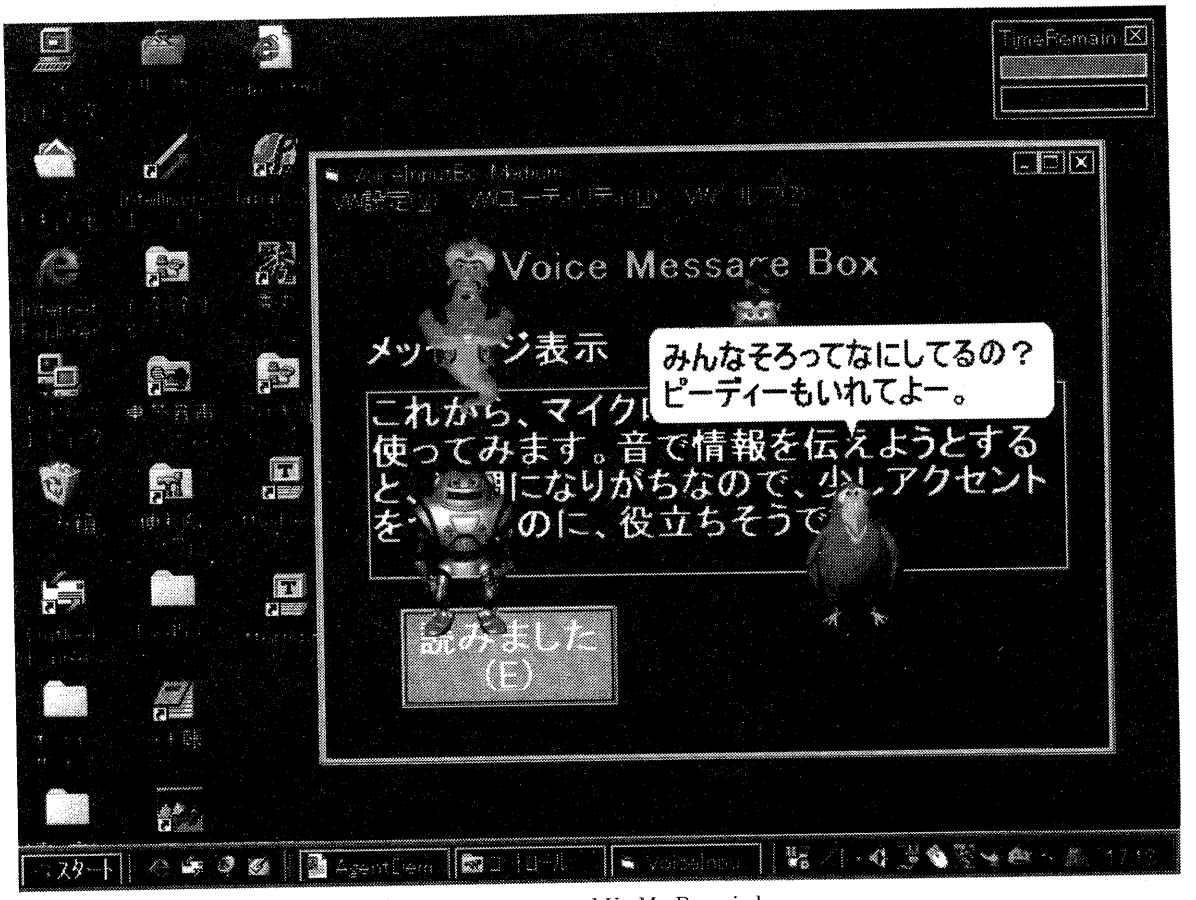

Fig. 2 Voicified MS Agents and VceMsgBox window

programs.

Windows Scripting Host is based on the Microsoft's object model and its full use may require some systematic training. So is Emacs-Lisp. The present set of Voice Windows components is small and easy to understand code-by-code. It is our hope that several visually disable programmers join the Voice Windows project, add codes while learning the system and languages, and build up sytems best for many specific uses.

Acknowledgments This work is supported by ITRC (Internet Technical Research Committee) and Mirai Kaitaku project by Japan Promotion of Science . We learned much about voice-based programs from discussions with Mr. Saito of Access Technology Co., Ltd., Mr. Kawamura of Japanese Society of Rehabilitation of Persons with Disabilities, Mr. Matuura, Mr. Momosaki, Mr. Tanaka, and Mr. Kamaya of Toshiba Co., Ltd., Mr. Kikka of Microsoft Co., Ltd., and Mr. Kimura of Fujitsu Co., Ltd. T.K. wish to thank Mr. Nakamae and Dr. Matsuzaki of University of Tokyo for assistance given to him in the early phase of this project, and members of the Meadow developers group for answering many questions about Emacs-Lisp and Meadow.

References

1) Raman, T. V.: *Audio System for Technical Readings*, PhD Thesis, Cornell University (1994). See also

Raman, T. V.: *Auditory User Interfaces – Toward the Speaking Computer–* , Kluwer Academic Publishers (1997).
2) Saito, M.: Software Production for the Visually Impaired, *Journal of IPSJ*, Vol. 36, No. 12, pp. 1116–1121 (1995) (in Japanese).
3) Kurihara, T.: The Present Status and the Future of Voice-Terminals for the Visually impaired, *CCUT-TR-89-01, Computer Centre, University of Tokyo* (1990) (in Japanese).
4) Windows95 screen reader by System Solution Center Tochigi Inc. See http://www.ssct.co.jp
5) Windows Scripting Host by Microsoft. See http://msdn.microsoft.com/scripting/windowshost/. WSH will also be supported in Windows 2000.
6) Multilingual Enhancement for GNU Emacs developed by Ken'ichi Handa Mikiko Nishikimi, Satoru Tomura and Naoto Takahashi. See http://www.etl.go.jp/ mule/
7) Meadow, Multilingual enhancement to gnu Emacs with ADvantages Over Windows, was written by Miyashita Hisashi and edited by Takeyori Hara. See http://tancho.scphys.kyoto-u.ac.jp/and ftp://ftp.etl.go.jp/pub/mule/Windows/
8) Kamae, T.: The present status of voice-based computer use, *3rd ITRC General Meeting*, June 24–26, 1998, (in Japanese)
(see `http://www.itrc.net/sympo/itrc_meet3/`).
Kamae, T.: A prototype for Voice-Meadow, *4th ITRC General Meeting*, November 26–28, 1998, (in Japanese)
(see `http://www.itrc.net/sympo/itrc_meet4/`).
9) T. Watanabe, T. Kamae, T. Kurihara; "*Multilingual Text-to-Speech system with active Auditory User Interface under Microsoft(R) Windows(R)*",
Interaction'99, (Tokyo Institute of Technology, March 4–5, 1999).
T. Watanabe, T. Kamae, T. Koide, H. Honda, S. Uno, T. Kurihara, S. Tajima:
"*Multilingual, Multimodal, and Multilingual Text-to-Speech System*",
submitted to Trans. of IPSJ (1999).