Chapter II-0

Introduction

The Cognitive Pilot Assistant (CPA) is the heart of the system. It accepts speech input, converts the audio to text, then processes the text into commands and/or requests. Examples might be “Say fuel quantity”. The speech output would then be the quantity of fuel indicated by the aircraft system interface.

The software presented in this Part II is primarily written in 'C' for the Linux GCC C compiler. Some routines may call libraries that require the GCC C++ compiler. If so, the accompanying “make files” will call the correct compiler. The software has been tested in Ubuntu 8 and Fedora 14.

Figure 0-1.1 is a diagram of the CPA. For a checklist request, the processor fetches the current checklist in internal memory. If the checklist cannot be found, the processor generates a short phrase stating so and outputs this to the TTS routine. Checklists may be updated before flight using a USB memory drive.

The first chapter discusses TTS systems. The voice in a TTS system provides its personality. This is closely related to the *prosody* of the voice (from the Greek for *song*). A TTS system is available on the CD-ROM (LPC Demo) which has little or no prosody. It sounds quite mechanical. This demonstration system illustrates the importance of prosody Proper prosody is key to a friendly system. Too much and it would sound out of place on the flight deck. Too little, and it would border on the annoying.

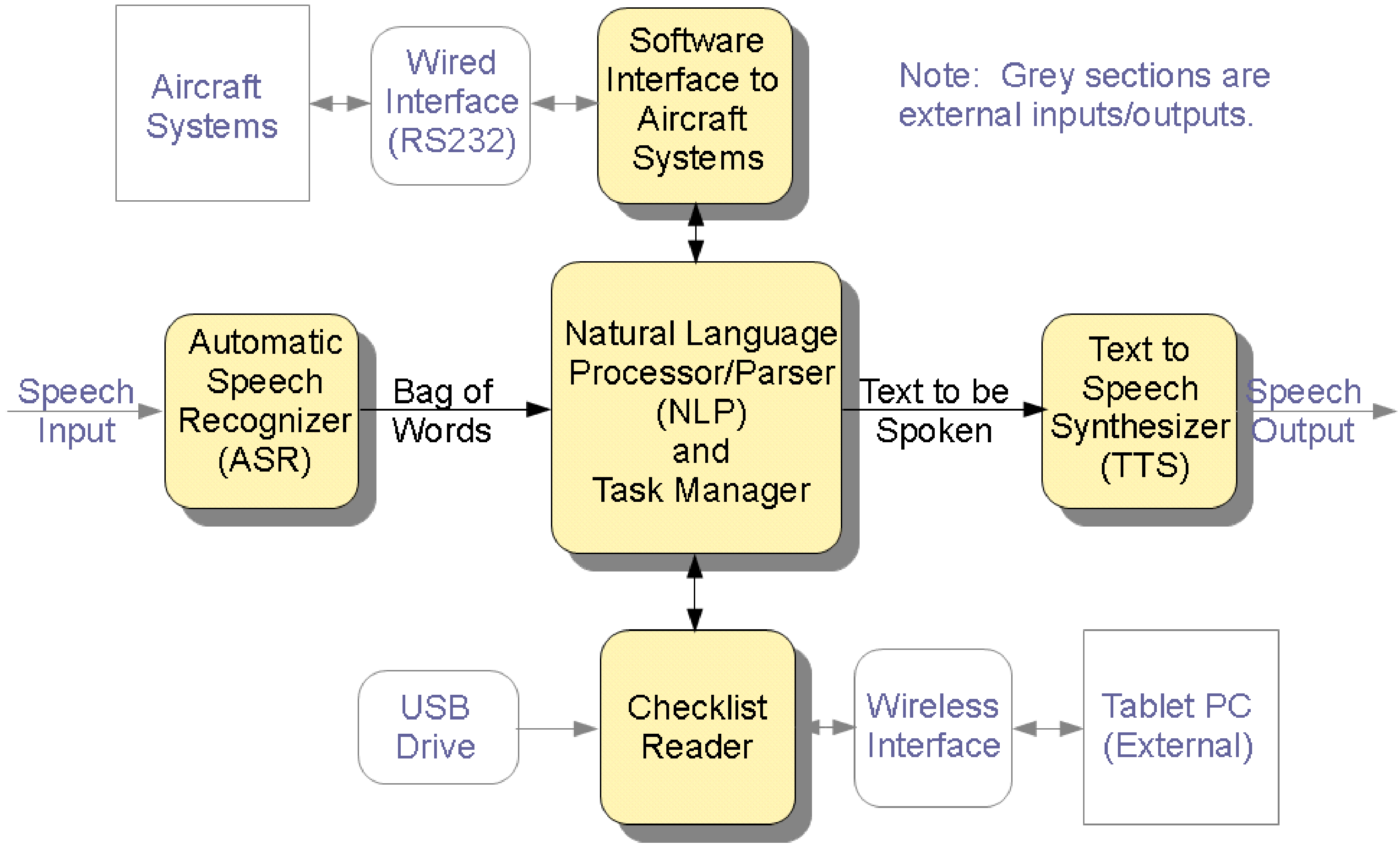
The second chapter in this Part discusses ASR systems. There are many ASR systems, but we have chosen PocketSphinx (written by Dr. David Huggins-Daines). It may not be the easiest to use, but it is available as freeware from Carnegie-Mellon University. And, it works well for this application. The inner workings of PocketSphinx are not explored, but a general overview of the typical modern ASR systems is presented. For a commercially fielded system, one of the commercially available speech recognizers might be preferred.

The third chapter deals with the natural language processor (NLP). The ASR only produces a bag of words. It does not attach any meaning to them. This is the task of the NLP. This is where the “intelligence” is found. There is, of course, no real intelligence in the machine in the sense of consciousness or creativity. The NLP is only a system of rules. But, with the availability of aircraft state and limited ability to communicate using spoken language, rules based decisions can indeed seem intelligent. (See also Polson, et al, Journal of XXX, date) We may be using the phrase “Cognitive Avionics” rather loosely (it does have a nice cachet), but, except for the ability to learn, the elements are all there, albeit in rather primitive forms.

The fourth chapter is a guide to implementing a wireless access point to communicate with the tablet. There are many pitfalls to doing so, and these will be discussed. Hopefully, the various required protocols are sufficiently stable that the method presented works without further modifications.

Several appendices supplement these chapters. Appendix A is a short tutorial on the Discrete Fourier Transform (DFT). Appendix B describes the process of computing a spectrogram.

The CD-ROM has a spreadsheet illustrating the DFT, a demo of a spectrogram, and a TTS system based on the Festival KAL voice using Linear Predictive Coding (LPC) coefficients. The spectrogram may be useful in debugging spoken word recognition errors. The latter illustrates the basics of one method of TTS.



Cognitive Pilot Assistant
Software Diagram

Figure 0-1.1 (Copyright 2012 AeroSpectra Inc)