1 SPEECH-ENABLED APPLICATIONS

1.1 INTRODUCTION

As humans, we see, hear, feel and smell. Human interaction is enriched by the resulting redundancy introduced by multimodal communication. In contrast, computer interfaces have relied primarily on visual interaction —today’s interfaces are like the silent movies of the past! As we approach the turn of the century, computers now have the ability to talk, listen and perhaps, even understand. Technological advances in the areas of speech synthesis and recognition, combined with the availability of computational power on the desktop that a few years ago could be found only in research labs, makes it possible to enrich human-computer interaction in ways that would once have counted as pure science-fiction.

Integrating modalities like speech into human-computer interaction requires rethinking how computing applications are designed in today’s world of visual computing. Applications that talk and listen need to be designed from the start to take advantage of the spoken medium; spoken interaction is different from and in many ways complementary to traditional visual interaction.

This book describes the speech-enabling approach —a technique that integrates speech interaction as a first class citizen into the user interface. We describe this
approach as implemented in two large systems, ASER—a computing system that produces high-quality interactive aural renderings of electronic documents—and Emacspeak—a full fledged speech interface to workstations including fluent spoken access to the World Wide Web (WWW) and many desktop applications.

This book is directed towards practitioners in the field of computer interface design. It is likely to prove of interest to academic and industrial researchers as well as engineers working on the design and implementation of computing systems of the future. Communication devices ranging from hand-held computers to smart telephones will need to integrate spoken interaction to be effective; the work described in this book is relevant to designing such interfaces. Examples include talking WWW browsers, listening to email messages as one is driving, and proofreading documents.

1.2 WHAT IS UI?

Computing applications typically consist of three phases:

1. Obtain user input,
2. Compute on the information,
3. Display the results.

Applications ranging from the simple “Hello World” program to complex windowing systems can be decomposed into these three phases—see Figure 1.1 on the facing page. The first and third steps, namely obtaining user input and displaying the computed results, constitute the user interface of an application. Notice that in this decomposition the computational phase is separable from the user interface.

1.2.1 Evolution of Complex Interfaces

The increasing complexity of the tasks our machines can perform has resulted in the concomitant evolution of equally complex user interfaces—this is necessary to enable effective human interaction with devices capable of performing computations in a fraction of the time it would take us to describe these tasks. Development of user interface peripherals such as high-resolution graphical displays and pointing devices have allowed application designers to construct sophisticated dialogues that open up user-level access to complex computational tasks. Rich graphical displays enable the construction of intricate and highly structured visual layout that can intuitively convey a vast amount of information—to re-use an oft quoted cliché, a picture is worth a thousand words!

This has led to the enormous success of the Graphical User Interface (GUI). Instead of spelling out complex incantations, users can intuitively access all the functionality available at the electronic desktop using a consistent and predictable interface. However, in our blind rush to exploit the Graphical User Interface, we are also guilty of a few oversights.