2 NUTS AND BOLTS OF AUDITORY INTERFACES

2.1 INTRODUCTION

This chapter covers various tools and techniques relevant to the design of effective auditory interfaces. It is designed to be a brief overview of currently available technology in this field. The topics introduced here will be used to advantage throughout this book in describing various kinds of auditory interaction. An excellent and frequently updated source of information in this field can be found in the monthly Frequently Asked Questions (FAQ) posting to the Usenet newsgroup comp.speech—a hypertext version can be found on the WWW at URL http://www.speech.cs.cmu.edu/comp.speech/.

Spoken interaction forms the backbone of all aural interfaces. We give a brief overview of currently available speech synthesis and recognition tools from the point of view of the application developer in Sec. 2.2 and Sec. 2.3. Spoken output can be effectively augmented by the use of non-speech auditory cues—the synthesis and application of such auditory icons is described in Sec. 2.4.3. All digitized sounds—both synthesized speech as well as non-speech sounds—can be modulated and modified to produce interesting auditory effects such as spatial audio by applying techniques from the field of Digital Signal Processing (DSP). We give an overview of such techniques in Sec. 2.4. All of these techniques together lead to the design of rich
auditory displays. The resulting potential of audio formatting to produce rich aural layout is covered in Sec. 2.5.

The successful design of innovative user interaction is influenced to a large extent by the development environment used to prototype such interfaces. The secret of success is rapid prototyping and user testing without spending large amounts of time in implementing interface strategies that upon further examination end up being dropped. Interactive and interpreted development environments like those provided by modern languages like Java, TCL/Tk, Perl5, and older interpreted languages like Lisp have a vital role to play in this process. We outline the use of development environments like Java in the context of producing both usable and re-usable interface components in Sec. 2.6.

2.2 SPEECH SYNTHESIS

This section gives a quick overview of various speech synthesis techniques and points the reader at the relevant technology in the field for further details. The focus of this book is on applying these techniques in various speech interaction contexts—not the implementation of these technologies. For details on specific speech synthesis technologies, see O'S87, Kla87, JAK87, KP95, Her91, Bos86, Wit82. In building practical applications, one is forced to trade-off speech quality versus the available computational power—we indicate the computational resources needed for the various synthesis techniques to aid in making this decision.

However, when describing various speech interaction scenarios throughout this book, we will assume the best-quality synthesis available. This will ensure that the interaction techniques outlined remain relevant even as increasing computational power brings today's compute intensive technologies within reach of tomorrow's hand-held devices.

2.2.1 Evaluation Of Speech Synthesis

Speech synthesis can be evaluated according to two completely orthogonal dimensions—naturalness and intelligibility. In addition, the effectiveness of the spoken interaction is highly dependent on expressiveness of the synthetic voice.

We define these evaluation criteria first:

Naturalness We perceive human speech as being the most natural. Naturalness is hard to quantify but easy to appreciate—most synthesized speech still sounds artificial—conversely, all human speech (even when not too intelligible) sounds very natural to our prejudiced ears! Lack of naturalness is often perceived as a distinct absence of emotion in synthetic speech.

Intelligibility Intelligibility is quantifiable—it measures the extent to which one can listen to, comprehend and absorb information conveyed using synthetic speech.