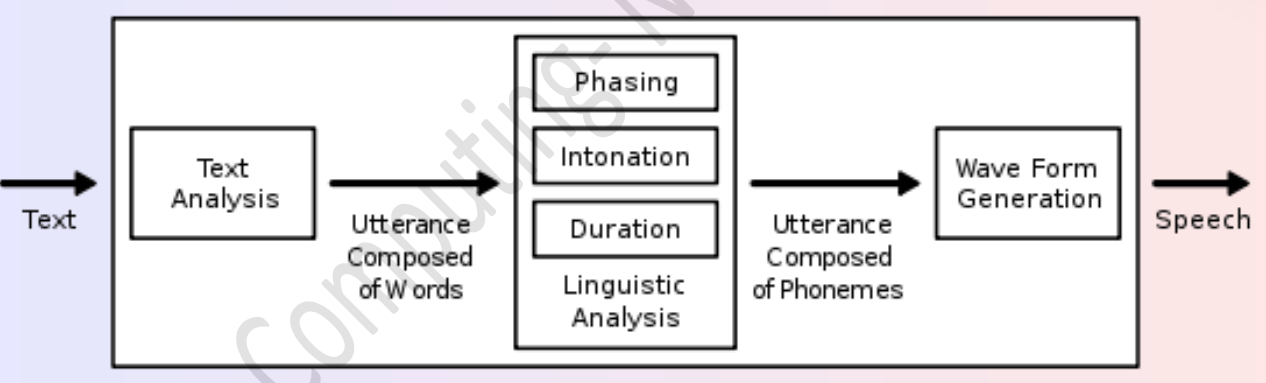
Unit -II

**Text-to-speech (TTS) Techniques:**

Text-to-speech (TTS) is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a Web page. TTS can enable the reading of computer display information for the visually challenged person, or may simply be used to augment the reading of a text message. Current TTS applications include voice-enabled e-mail and spoken prompts in voice response systems.

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products.

A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.



A text-to-speech system (or "engine") is composed of two parts: A front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called *text normalization*, *pre-processing*, or *tokenization*. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called *text-to-phoneme* or *grapheme-to-phoneme* conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the *synthesizer*—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the *target prosody* (pitch contour, phoneme durations), which is then imposed on the output speech.

***“Text-to-speech applications are those applications that change written language into spoken language*.”**

**Designing the Right UI: Multichannel and Multimodal UIs and Services**

**1. Importance of Multichannel UIs**

* **User Expectations**: Modern users expect to interact with apps across multiple devices and platforms, including smartphones, tablets, smartwatches, and web browsers.
* **Consistency Across Channels:**
  + **Unified Branding:** Maintain consistent visual branding, ensuring that the app looks and feels the same across all platforms.
  + **Feature Parity**: While not all features may be available on every platform, critical features should be consistent to prevent user frustration.
* **Case Studies:**
  + **Example 1:** A banking app providing services through mobile apps, web portals, and ATMs, where the user experience is seamless and consistent.
  + **Example 2:** An e-commerce platform available on mobile, web, and voice-assisted devices, offering a consistent shopping experience across all channels.

**2. Evolution of Multimodal UIs**

* **Historical Context:** Initially, UIs were primarily graphical, relying on visual elements like buttons and icons. With the advent of voice assistants, gesture controls, and other input methods, UIs have evolved to support multiple modes of interaction.
* **Modern Multimodal Interfaces:**
  + **Touch + Voice:** Combines touch interaction with voice commands, allowing users to perform complex tasks efficiently (e.g., searching for a product using voice and selecting it by touch).
  + **Gesture-Based UIs:** Widely used in gaming and smart devices, where gestures like swipes or hand movements control the interface.

**3. Designing for Multichannel and Multimodal UIs**

* **Adaptive Design:**
  + **Fluid Layouts:** Use layouts that adapt to different screen sizes and orientations, ensuring that content is always displayed optimally.
  + **Dynamic UI Elements:** Create UI elements that can resize or reformat depending on the device's capabilities (e.g., displaying more detailed information on larger screens).
* **User Context Awareness:**
  + **Contextual UI:** Design UIs that adapt based on the user’s environment, such as switching from a visual to a voice-based interface when the user is driving.
  + **Proactive Assistance:** Leverage AI to anticipate user needs, offering suggestions or shortcuts based on past behavior and current context.

**4. Tools and Technologies for Multichannel and Multimodal UIs**

* **Cross-Platform Development Tools:**
  + **Flutter:** A UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.
  + **React Native:** Allows developers to create mobile apps using React, a popular web framework, ensuring a consistent user experience across platforms.
* **Voice and Gesture Recognition Technologies:**
  + **Google Assistant SDK:** Provides tools for integrating voice interactions into apps, enabling multimodal experiences.
  + **ARCore:** Google's platform for building augmented reality experiences, incorporating gestures and environmental awareness into UIs.

**5. Future Trends in Multichannel and Multimodal UIs**

* **Increased Use of AI:** AI will play a significant role in personalizing and optimizing user experiences across channels and modes of interaction.
* **Expansion of Multimodal Interfaces:** As devices like smart glasses, AR/VR headsets, and IoT devices become more prevalent, the demand for UIs that seamlessly integrate multiple interaction methods will grow.
* **Cross-Device Continuity:** Future UIs will enable users to start an interaction on one device and seamlessly continue it on another, without losing context or data.