Design Technique for Voice Browsers

Abstract

Browser technology is changing very fast these days and we are moving from the visual paradigm to the voice paradigm. Voice browser is the technology to enter this paradigm. A voice browser is a “device which interprets a (voice) markup language and is capable of generating voice output and/or interpreting voice input, and possibly other input/output modalities."This paper describes the requirements for two forms of character-set grammar, as a matter of preference or implementation, one is more easily read by (most) humans, while the other is geared toward machine generation.

1. Introduction

A voice browser is a “device which interprets a (voice) markup language and is capable of generating voice output and/or interpreting voice input, and possibly other input/output modalities." The definition of a voice browser, above, is a broad one. The fact that the system deals with speech is obvious given the first word of the name, but what makes a software system that interacts with the user via speech a "browser"? The information that the system uses (for either domain data or dialog flow) is dynamic and comes somewhere from the Internet. From an end-user's perspective, the impetus is to provide a service similar to what graphical browsers of HTML and related technologies do today, but on devices that are not equipped with full-browsers or even the screens to support them. This situation is only exacerbated by the fact that much of today's content depends on the ability to run scripting languages and 3rd-party plug-ins to work correctly.

Much of the efforts concentrate on using the telephone as the first voice browsing device. This is not to say that it is the preferred embodiment for a voice browser, only that the number of access devices is huge, and because it is at the opposite end of the graphical-browser continuum, which high lights the requirements that make a speech interface viable. By the first meeting it was clear that this scope-limiting was also needed in order to make progress, given that there are significant challenges in designing a system that uses or integrates with existing content, or that automatically scales to the features of various access devices.

2. Voice Browser Documents

2.1 Dialog Requirements:

"A prioritized list of requirements for spoken dialog interaction which any proposed markup language (or extension thereof) should address."

The Dialog Requirements document describes properties of a voice browser dialog, including a discussion of modalities (input and output mechanisms combined with various dialog interaction capabilities), functionality (system behavior) and the format of a dialog language. A definition of the latter is not specified, but a list of criteria is given that any proposed language should adhere to. An important requirement of any proposed dialog language is ease-of- creation. Dialogs can be created with a tool as simple as a text-editor, with more specific tools, such as an (XML) structure editor, to tools that are special-purposed to deal with the semantics of the language at hand.

2.2 Grammar Representation Requirements

It defines a speech recognition grammar specification language that will be generally useful across a variety of speech platforms used in the context of a dialog and synthesis markup environment."

When the system or application needs to describe to the speech-recognizer what to listen for, one way it can do so is via a format that is both human and machine-readable.

2.3 Model Architecture for Voice Browser Systems Representations

"To assist in clarifying the scope of charters of each of the several subgroups of the W3C Voice Browser Working Group, a representative or model architecture for a typical voice browser application has been developed. This architecture illustrates one

possible arrangement of the main components of a typical system, and should not be construed as a recommendation."

2.4 Natural Language Processing Requirements

It establishes a prioritized list of requirements for natural language processing in a voice browser environment. The data that a voice browser uses to create a dialog can vary from a rigid set of instructions and state transitions, whether declaratively and/or procedurally stated, to a dialog that is created dynamically from information and constraints about the dialog itself.

The NLP requirements document describes the requirements of a system that takes the latter approach, using an example paradigm of a set of tasks operating on a frame-based model. Slots in the frame that are optionally filled guide the dialog and provide contextual information used for task-selection.

2.5 Speech Synthesis Markup Requirements

It establishes a prioritized list of requirements for speech synthesis markup which any proposed markup language should address. A text-to-speech system, which is usually a stand-alone module that does not actually "understand the meaning" of what is spoken, must rely on hints to produce an utterance that is natural and easy to understand, and moreover, evokes the desired meaning in the listener. In addition to these prosodic elements, the document also describes issues such as multi-lingual capability, pronunciation issues for words not in the lexicon, time-synchronization, and textual items that require special preprocessing before they can be spoken properly.

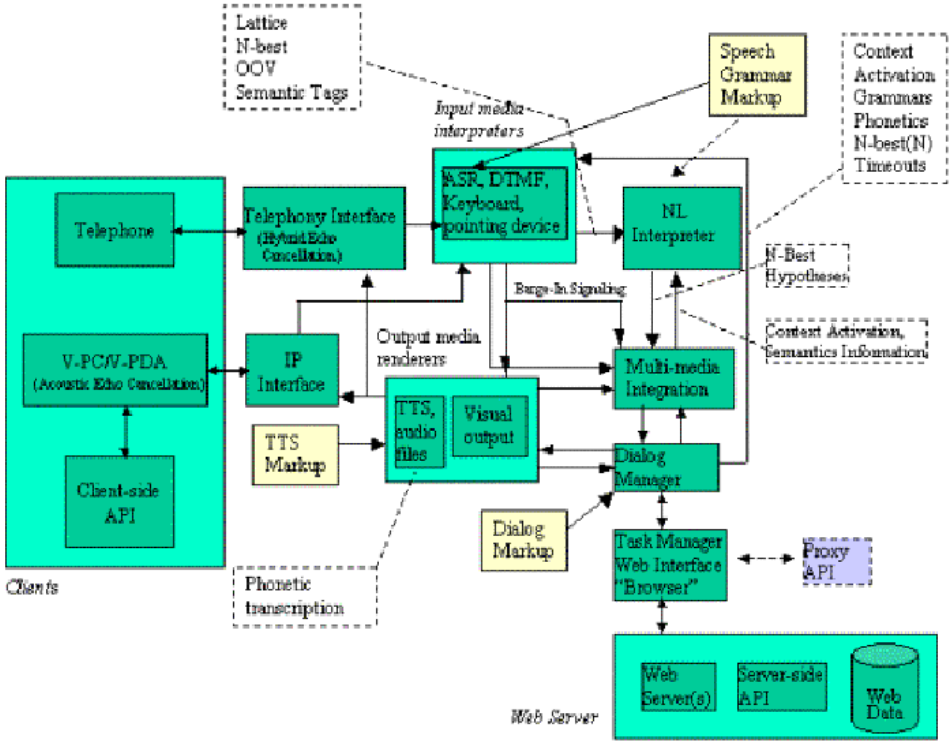
3. System Architecture

"The architecture diagram was created as an aid to how we structure our work into subgroups. The diagram will help us to pinpoint areas currently outside the scope of existing groups."

Although individual instances of voice browser systems are apt to vary considerably, it is reasonable to try and point out architectural commonalties as an aid to discussion, design and implementation. Not all segments of this diagram need be present in any one system, and systems which implement various subsets of this functionality may be organized differently. Systems built entirely third-party components, with architecture imposed, may result in unused or redundant functional blocks.

Two types of clients are illustrated: telephony and data networking. The fundamental telephony client is, of course, the telephone, either wirelined or wireless. The handset telephone requires PSTN (Public Switched Telephone Network) interface, which can be either tip/ring, T1, or higher level, and may include hybrid echo cancellation to remove line echoes for ASR barge-in over audio output. A speakerphone will also require an acoustic echo canceller to remove room echoes. The data network interface will require only acoustic echo cancellation if used with an open microphone since there is no line echo on data networks. The IP interface is shown for illustration only. Other data transport mechanisms can be used as well.

The model architecture is shown below. Solid (green) boxes indicate system components, peripheral solid (yellow) boxes indicate points of usage for markup language, and dotted peripheral boxes indicate information flows.


Lattice
N-best
OOV
Semantic Tags
Input media interpreters
Speech Grammar Markup
Context Activation Grammars Phonetics N-best(N) Timeouts
Telephone
Telephony Interface (Hybrid Echo Cancellation)
ASR, DTMF, Keyboard, pointing device
NL Interpreter
N-Best Hypotheses
Barge-In Signaling
V-PC/V-PDA (Acoustic Echo Cancellation)
IP Interface
Output media renderers
Context Activation, Semantics Information
Multi-media Integration
TTS, audio files
Visual output
TTS Markup
Dialog Manager
Client-side API
Dialog Markup
Task Manager Web Interface "Browser"
Proxy API
Clients
Phonetic transcription
Web Server(s)
Server-side API
Web Data
Web Server

4. Conclusion

If a voice browser is to converse with the user, then a description, either explicit or derived and implicit, must exist for the underlying system to "render" into a dialog. Ultimately, it will be up to solution-providers to take an inventory of the existing content (if any), development tools, data-access requirements, deployment platforms, and application goals such as cost, security, richness and robustness, before they can decide what technology to use. More likely than not, for the time-being, multiple content types will be required to deliver the most natural experience on each type of browsing device -- this is both a technical limitation, and driven by the user's who expect the latest-and-greatest attributes of each modality to be featured in their applications.