

Configuration and Management of Speaker Verification Systems

W3C Workshop on Speaker Biometrics and
VoiceXML 3.0

Chuck Johnson
Architect
iBiometrics, Inc.

Introduction

For peak performance of a Speaker Verification solution, the VoiceXML client (voice application) needs to be able to query and set the necessary initialization and configuration (setup) parameters, control the operation of Speaker Verification resources (engines), and interpret the verification results.

Interpretation of Verification Results

Speaker Verification engines, depending on the vendor, are configured to return raw (numeric) verification scores, normalized verification scores, verification decisions, or some combination of scores and decisions - or error results. In addition, some engines return confidence scores.

Some standardization of return data/info is necessary e.g. a consistent range for normalized scores. The engine should return a basic (minimum) set of error results. The engine should return a pass/fail or a pass/fail/inconclusive decision.

Enrollment - Voice Model Creation

Traditionally (in voice model enrollment scenarios) the client application implements the enrollment dialog - manages the voice dialog, the error handling and the associated call flow.

Within the context of an enterprise security framework, the client application should manage the enrollment process: start/stop/resume/abort enrollment, query enrollment status (in-progress, aborted, etc), and retrieve the enrollment outcome.

Some engines support multiple modes of operation. The client should be able to query and set the mode of operation [for enrollment]

Voice Model Database Management

Set up of the voice model (voiceprint) database entails: creation of the schema (tables), creation of database users, and establishment of rights and access privileges - tasks that are governed by enterprises guidelines and security policies.

Those tasks are usually performed by a system administrator or DBA - not by the client application.

Client applications may be able to manage the voice models: copy the voice model, delete the voice model and/or rename the voice model identifier

Distinct User Populations

Many SIV applications have distinct user populations e.g. Financial Services, Community Corrections, and Social Services. These populations (groups) include children, females, ethnic groups, regional speakers, and application specific groups. Some world [background] models are not optimized for distinct user populations. Custom or group specific background models can improve verification performance (accuracy).

Client applications should be able to utilize custom or group specific background models. And, optionally, update (adapt) group

Different Classes of Users

In Financial Services Applications, access to different features of the service may have a higher (or lower) security setting. In Corrections Applications, different classes of users will have different security settings or levels. The users are often put into classes based on risk – typically high, medium, and low.

Client applications should be able to: query and set the current operating point (security level), query and set unsupervised adaptation thresholds, and manage and control supervised adaptation

Voice Model Adaptation

The idea of voice model adaptation is not intuitive!

In the 'not so distant past' there were articles saying that voice model adaptation not always needed or questioning the efficacy of the adaptation process. Numerous articles/reports from vendors and industry experts have clearly demonstrated the need for, and effectiveness of voice, model adaptation.

The client application should be able to query unsupervised adaptation settings, enable/disable adaptation, query adaptation outcome (result) and, optionally, set adaptation threshold and rollback adaptation [from the last turn].

The client should manage supervised adaptation: control audio buffers, make adaptation requests, and, optionally, rollback adaptation [from the last turn].

Solution Architecture Issues

** Optional Slide **

Time permitting, I may give a brief presentation of interface, security and architectural issues associated with 'loosely coupled' SIV systems. These are systems where most or all of the SIV components/resources (app server, data store, voice interpreter and/or SIV engine) are distributed across multiple systems, across the enterprise, or across multiple enterprises.

Summary and Wrap up